Design a Vulnerable AD set

Table of Contents

Hi Folks, today I would like to share how did I design and build a vulnerable AD set. Before moving to this topic, let me introduce the motivation and some features of this AD set.

Motivation

I know there are few scripts can automate the process of generating common AD misconfigurations such as DACL abuse, weak credential, kerberoasting, etc. If you are interested in them, here are the github repo: https://github.com/WaterExecution/vulnerable-AD-plus and https://github.com/Orange-Cyberdefense/GOAD. These authors already did a great job, they make the process simple and fast. However, some other common elements in AD exploitation cannot be produced easily only with script, so some manual configuration and setup is also very important. Besides, I do not want my vulnerable AD set to be a purely AD exploitation. I hope it is more complex, difficult, and realistic.

Features

1: It is not CTF style, no side quest. All flags are on Linux home folder or Windows Desktop. Its style is similar to many famous AD labs like CPTX, Cybernetics, CRTP, etc.

2: The AD consists of 6 machines, including 2 Linux domain joint machines. Many people are already familiar with AD exploitation in Windows environment, but how about Linux domain joint machines? You even need to exploit the AD from your Kali VM.

3: Multiple services and apps make the vulnerable AD more fun and complex, such as FTP, SMTP, POP3, IMAP, Samba, ElasticSearch, WordPress, Kibana, etc.

4: Few rabbit holes, but not just for misleading you. They are reasonable. Get RCE from a web app? But it will not help too much. A lot of privilege escalation vectors? But they are not necessary.

5: Basic OSINT and inference according to context.

6: Hardened machines. They implemented latest Windows Defender, AppLocker, etc. But I don’t think they will be the biggest issue, enumeration does matter.

7: Classic elements in AD: SQL Linked Server, Kerberos Delegation, Kerberoasting/ASREPRoasting, Credential Reuse

8: Some barriers during typical exploitation. Copy and paste steps in an AD exploitation cheat sheet? They will not work, you need to understand why your exploitation failed.

Preparation

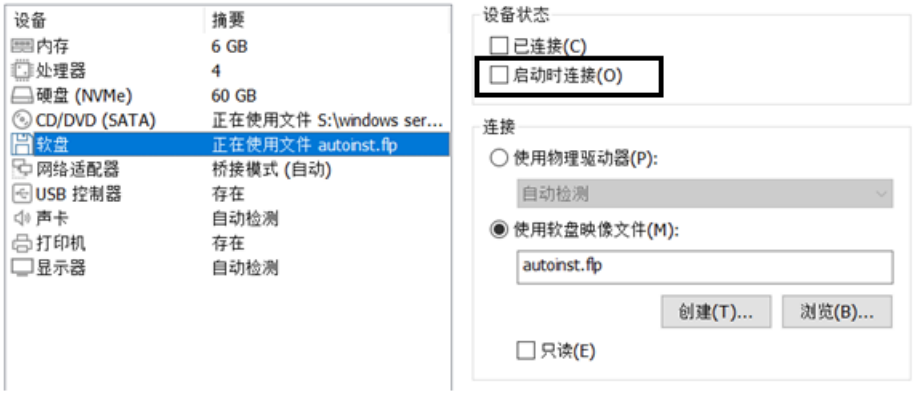
During the design, I downloaded multiple apps/tools, and referred many articles. But before building the AD set, only 2 things are required.

Windows Server 2019: https://www.microsoft.com/en-us/evalcenter/evaluate-windows-server-2019

Windows 10: https://www.microsoft.com/en-us/software-download/windows10%20 (You can also use Windows Server 2019 instead)

Ubuntu 22.04: https://ubuntu.com/download/desktop

I used VMWare workstation to host these VMs, and I used Bridged Network. I tested NAT network, it also works well! After creating a Windows Server 2019 VM, do not forget to uncheck **Connect at Power Up** (in screenshot), in section **floppy disk**, otherwise you cannot install the OS successfully.



How to assign hardware resource to these VM? I list them on the following table. In my opinion, they are all above the required resources, I feel each VM runs smoothly.

OS	RAM	CPU	Hard Drive
Domain controller *1	6G	1*2	80G
Other Windows computers *3	4G	1*2	35G
Ubuntu domain computers *2	4G	1*2	30G

Forget to mention that, I used a Windows 10 pro as the client server in domain. I cannot remember clearly where did I download the image. If it is not convenient for you to download a Windows 10 pro image, you can absolutely use Windows Server 2019 instead, it does not matter. After installing all VM, we can start to configure the OS.

Configurations and Design

Just clarify, this part is not a detailed guidance for building an AD environment. Instead, this part focuses more on the design. Of course, I will absolutely go through some technique difficulties and how did I resolved them.

Let's take a look at all machines and their roles.

Computer	IP	Role
dc.blackops.local	192.168.0.56	Domain Controller
web01.blackops.local	192.168.0.51	Public Web Server
file01.blackops.local	192.168.0.52	File Server in Domain
client01.blackops.local	192.168.0.53	Client Server in Domain
srv01.blackops.local	192.168.0.54	SQL Server 1
srv02.blackops.local	192.168.0.55	SQL Server 2

Web01 simulates a public-facing server in the domain, external user has access to its services. It hosts multiple services, including web apps, SMB, SMTP, POP3, etc.

File01 simulates an internal file server in the domain, because it is running a FTP server. Domain user can exchange file on this host.

Client01 simulates a client computer in the domain. Domain user Helen Park is the owner of it. Helen is a member of Help Desk group, so she has some permissions.

SRV01 simulates a normal server in the domain, it has an SQL instance. It is also linked to an SQL instance on SRV02

SRV02 simulates another server in the domain, it not only has an SQL instance, but also is able to delegate other domain users, except those protected high-privileged ones.

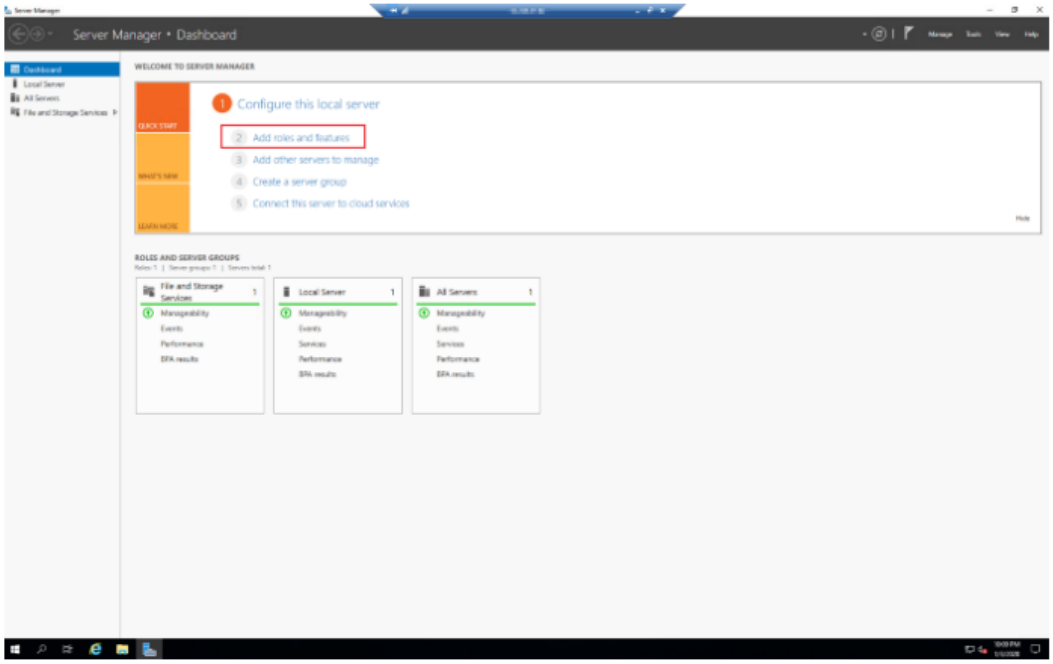
Domain Controller

dc.blackops.local

First, we need to configure the domain controller. There are already many articles about it, so I recommend you to check this article: https://kamran-bilgrami.medium.com/ethical-hacking-lessons-building-free-active-directory-lab-in-azure-6c67a7eddd7f. You can jump to **[Configuring Services]** and continue.

Configuring Services

Now that we are connected to the machine, its time to configure it as a Domain Controller. Let's launch the Server Manager and click on **Add roles and features** option.

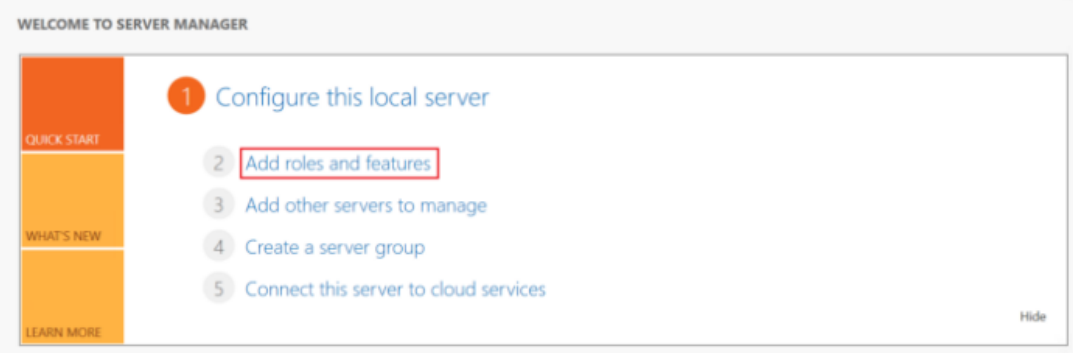


If you just want to replicate my AD set, you can stop at [Configuring Certificate Services] since I did not adopt ADCS this time. If you are interested in this part, you can absolutely continue to read. And I plan to add ADCS feature to my next

AD set.

Configuring Certificate Services

The next step is to setup Certificate Services. Let's launch the Server Manager again and click on the **Add roles and features.**



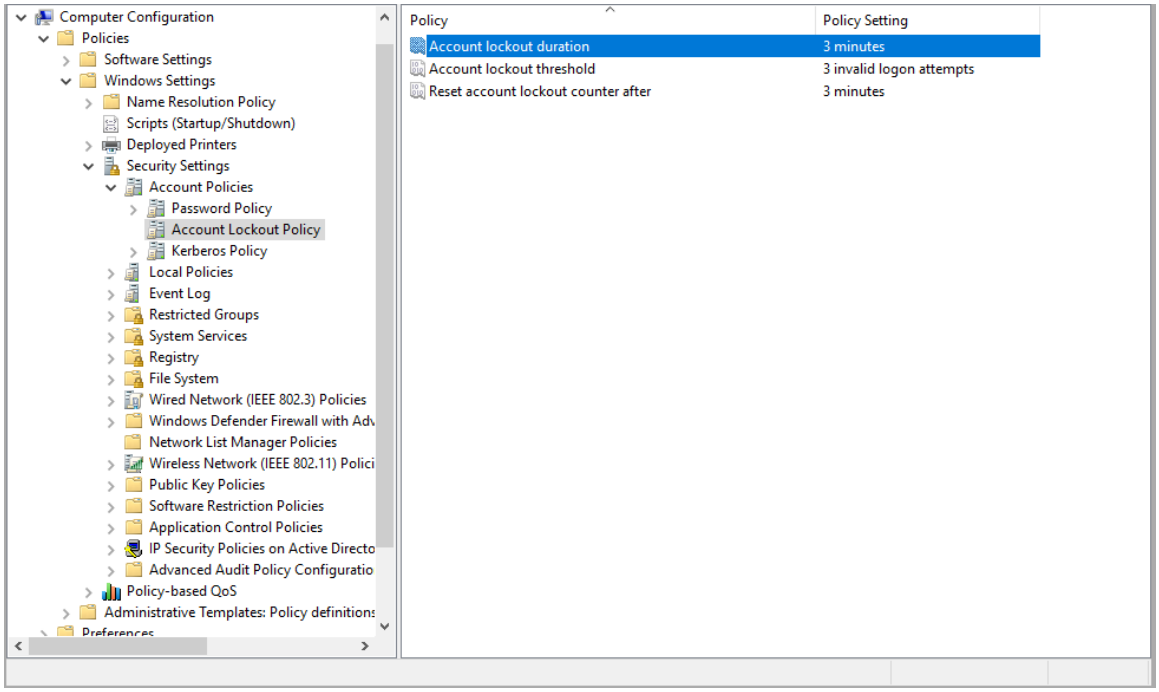
Personaly I set the domain as **blaskops.local**, the NETBIOS name is **BLACKOPS**, and IP for domain controller is **192.168.0.56**. Then open Active Directory Users and Computers application, let's make some changes.

First, let's create some domain users. Of course, you can create more domain users to increase the enumeration difficulty.

Account	Password	Functionality/Role
Administrator	<Any strong password>	Domain Admin
russell.adler	Ajobtodo!	DACL Abuse
alex.mason	CIAAgent1984	Credential Reuse
svc_sql	<Any strong password>	Service account
jason.hudson	jkhnrjk2020!	In a special group
ir_operator	Pass1kirsty **(in rockyou.txt)**	Credential Reuse
df_operator	Pass1kirsty **(in rockyou.txt)**	Credential Reuse/DACL Abuse
helen.park	Summer2022!	In a special group
frank.woods	<Any strong password>	DACL Abuse
svc_sq1	Password1 **(in rockyou.txt)**	Honeypot user

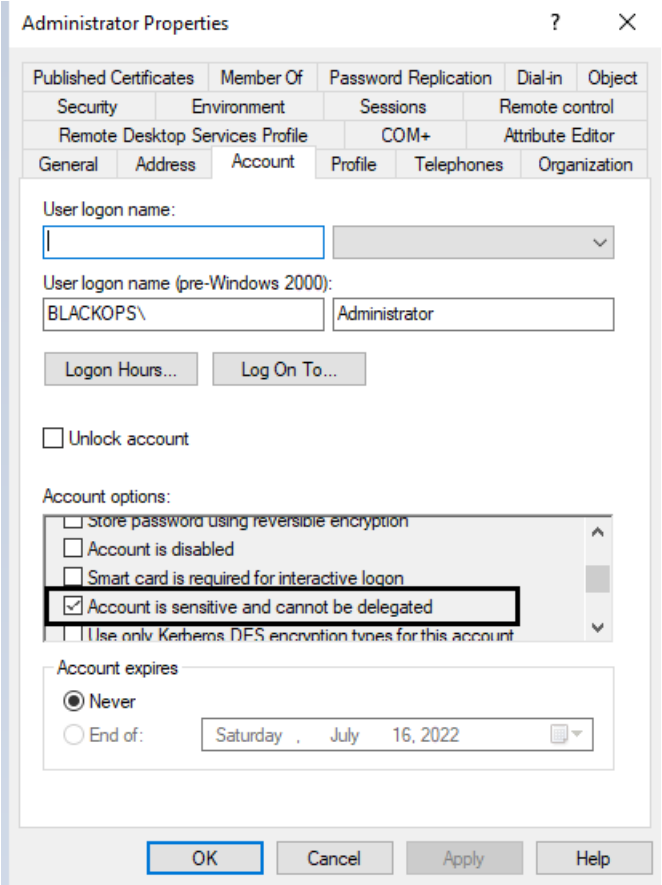
As you notice, there are some accounts with weak password. I set those weak passwords on purpose. **ir_operator** and **df_operator** share the same password to make room for credential use. In exploitation chain, ir_operator can be set a SPN, then ir_operator can be kerberoasted, this is the reason why I set a weak password for both of them. As to svc_sq1 account, you may find that this is a honeypot account, because it is sq1 not sql : D You can easily kerberoasting svc_sq1 and crack the password, but it will not help at all. In reality, your attack

will be logged then blue team will notice it. Anyway, since there are few weak passwords, we must eliminate dictionary attack and brute-force attack, so we need to implement account lockout policy. This article tells you how to achieve this: https://www.windows-active-directory.com/account-lockout-policy-active-directory.html#:~:text=Double%2Dclick%20the%20domain%20to,Policies%20%E2%86%92%20Account%20Lockout%20Policy.

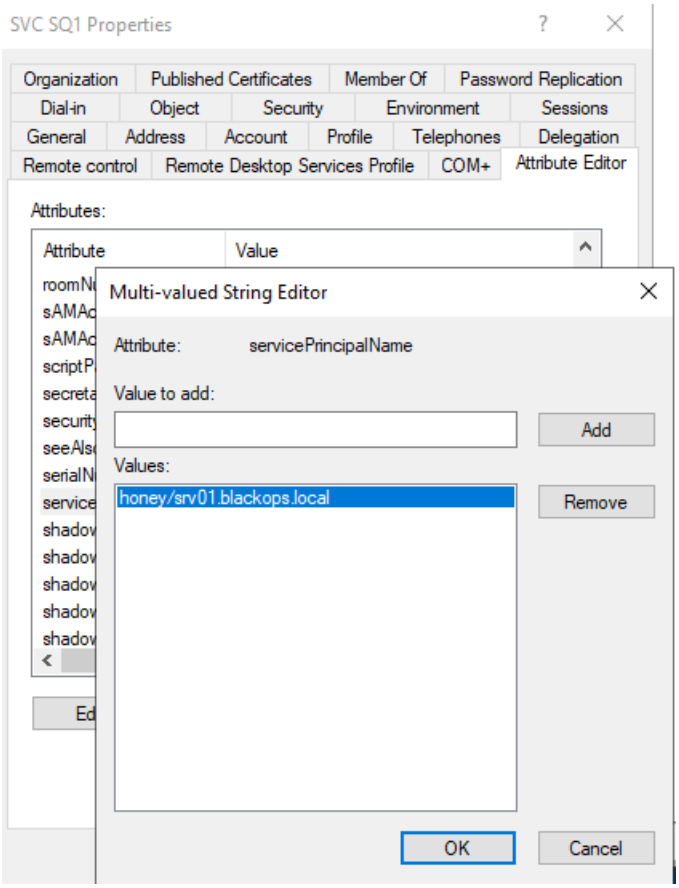


Apart from few passwords, I specified few passwords such as russell.adler's, these passwords cannot be cracked with a normal dictionary, but they will be used later in design steps, you can change them but just change them in following steps as well.

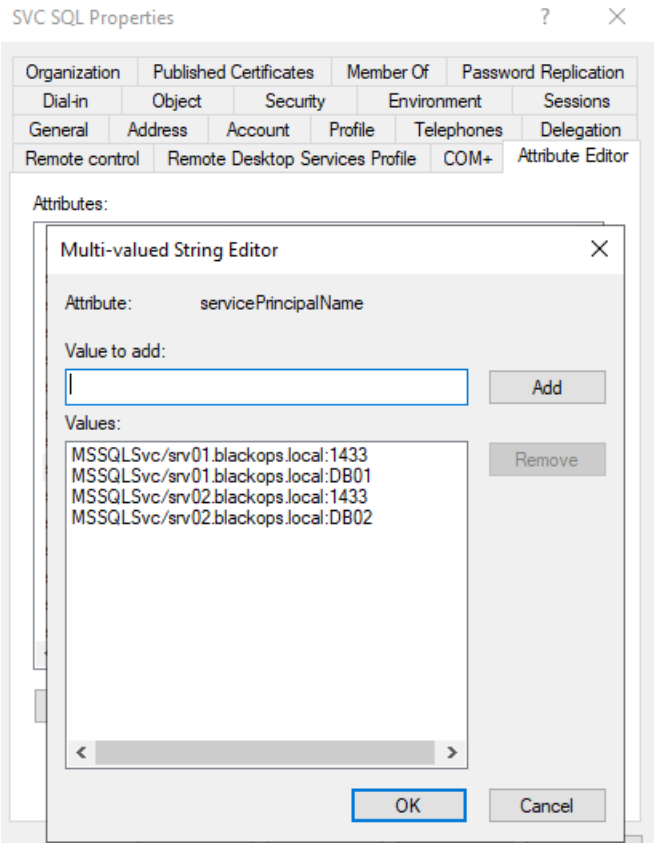
By the way, I set Administrator as a protected user, which cannot be delegated. It not only makes the environment more realistic but also increases the difficulty when abusing delegation.



I added an OU called Service Accounts, and I moved svc_sql and svc_sq1 to this OU. Since they are designed as service accounts, we need to set SPN for them. svc_sq1 is a honeypot account, so it is easy to set, as long as the SPN is in correct format.



After that, we can choose to set SPN for svc_sql, which is designed as service account for SQL Server instances in domain. It is not required to set it now, but it does not hurt. Why? Because we can use a tool to automate this process later without getting any error. If you follow my SPN settings, please make sure your SQL Server instance are named DB01 and DB02 respectively. Later I will show how to configure SQL Server instances.



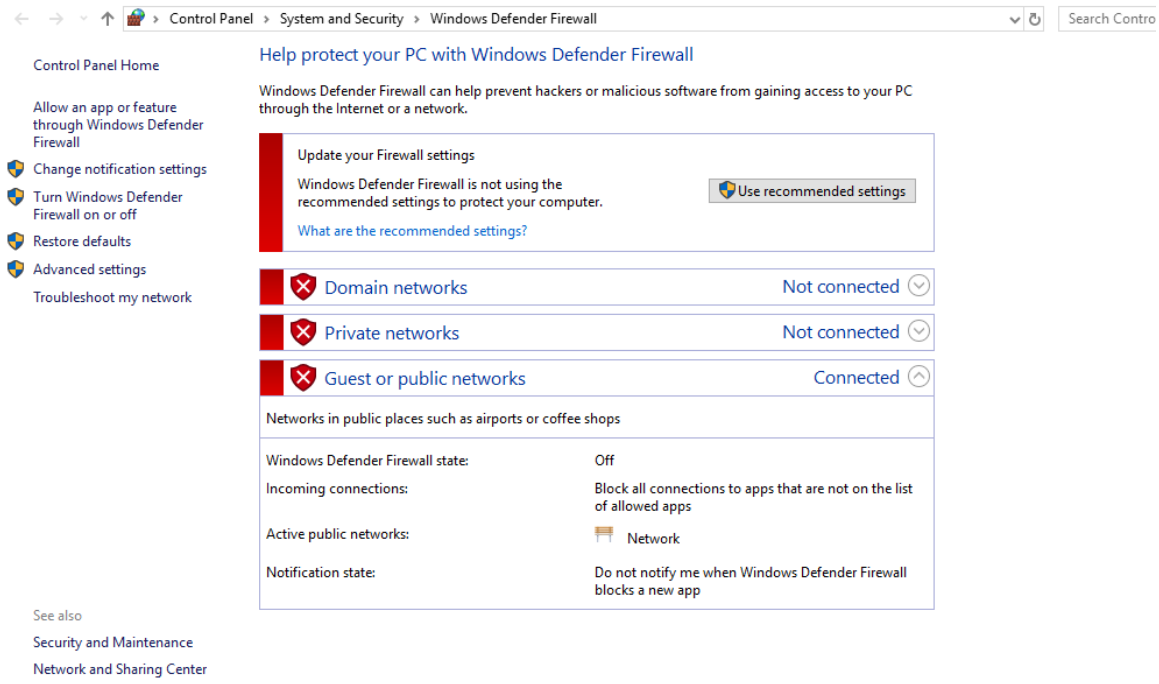
Then, I add helen.park to Helpdesk group. You can also create more groups, or add more users to groups.

Account	Group	Comment	Functionality
helen.park	Helpdesk	Created Group	For context
jason.hudson	Monitor Group	Created Group	For context
svc_sql	SQL Manager	Created Group	For context

helen.park should be able to **RDP** to **client01**, we need to add helen.park to a localgroup in client01, I will mention it later. svc_sql has RDP access to SRV01 and SRV02. jason.hudson has **WinRM** right to **SRV01**, so add him to a localgroup in SRV01. Therefore, we need to impersonate **jason.hudson** instead of Administrator when abusing delegation :). Instead of adding these users to local group, we can also link a GPO to them to enforce. This video shows detailed steps to achieve this: https://www.youtube.com/watch?v=euFiRyjRt1E

And I also turn on **automatic logon** for domain administrator on DC, you can check this article: https://docs.microsoft.com/en-us/troubleshoot/windows-server/user-profiles-and-logon/turn-on-automatic-logon.

Besides, it is up to you whether turn on/off windows defender firewall. It is on by default, but I turn off it. **The setting is the same for every windows domain computers.**



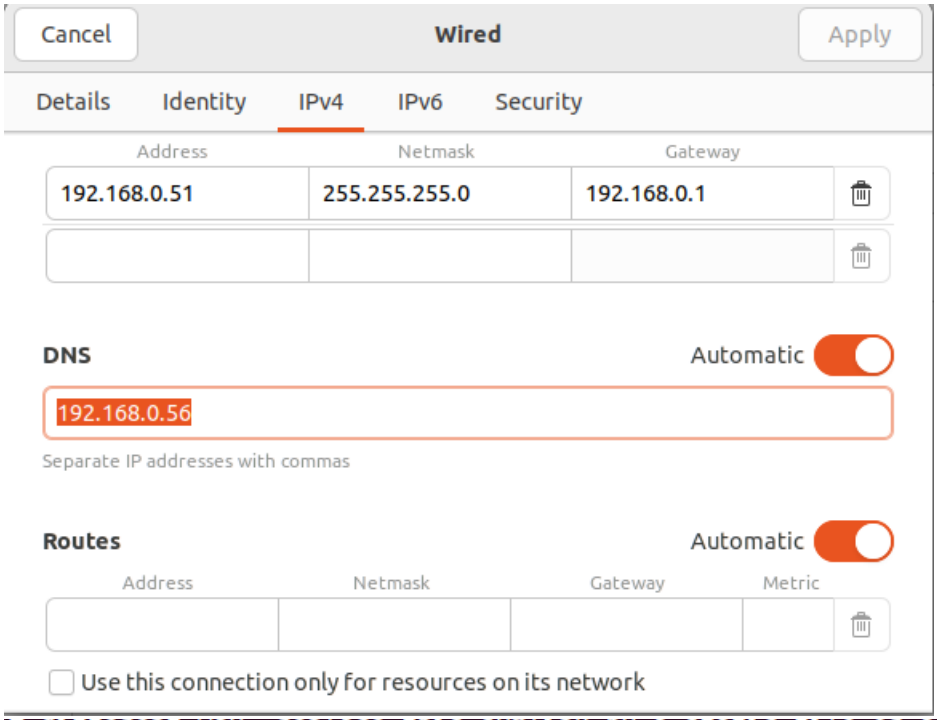
Now, we completed basic settings on DC, but we will revisit DC after adding domain computers and configuration of SQL Server instances.

Linux Domain Computer 1

web01.blackops.local

Since it is difficult to configure SRV01 and SRV02, so let's start from easier ones.

First of all, we need to set DC's ip as DNS. And add a new entity to /etc/resolve.conf. Be aware that after each reboot, we need to re-add the entity.




```
# different way, replace this symlink by a static file or a
#
# See man:systemd-resolved.service(8) for details about the
# operation for /etc/resolv.conf.

nameserver 192.168.0.56
nameserver 127.0.0.53
options edns0 trust-ad
search .
```

By this way, we can look up domain computers and join domain later.

```
mason@web01:~/Desktop$ nslookup dc.blackops.local
Server:         192.168.0.56
Address:        192.168.0.56#53

Name:   dc.blackops.local
Address: 192.168.0.56
Name:   dc.blackops.local
Address: 2601:18f:380:2100:c060:4344:85fa:1f4c
```

Then we need to set up few Linux local accounts.

Account	Password	Functionality	Comment
ubuntu	Strong Password	Privileged account	Not for exploitation
mason	CIAAgent1984	Credential reuse	Reflection of domain password
mailadmin	Password	Log in POP3 server	Get it from wordpress web app
hudson	Strong Password	Send an email to mailadmin	Not for exploitation

In regard to how to join a Linux computer to domain, this article gives detailed instruction: https://www.informaticar.net/join-ubuntu-machine-to-windows-domain/. You will not make any mistake as long as you follow steps. You can use **klist** to check tickets to verify that the Linux machine successfully joined domain.

```
mason@web01:~/Desktop$ su administrator@blackops.local
Password:
administrator@blackops.local@web01:/home/mason/Desktop$ klist
Ticket cache: FILE:/tmp/krb5cc_1854800500_g98N0c
Default principal: Administrator@BLACKOPS.LOCAL

Valid starting       Expires              Service principal
06/16/2022 13:54:24  06/16/2022 23:54:24  krbtgt/BLACKOPS.LOCAL@BLACKOPS.LOCAL
        renew until 06/17/2022 13:54:24
administrator@blackops.local@web01:/home/mason/Desktop$
```

File **/etc/krb5.keytab** is readable for root by default, it contains machine account web01$'s credential. We can use python script keytabextract.py (https://github.com/sosdave/KeyTabExtract) to extract them.

```
└─# python3 keytabextract.py krb5.keytab
[*] RC4-HMAC Encryption detected. Will attempt to extract NTLM hash.
[*] AES256-CTS-HMAC-SHA1 key found. Will attempt hash extraction.
[*] AES128-CTS-HMAC-SHA1 hash discovered. Will attempt hash extraction.
[+] Keytab File successfully imported.
        REALM : BLACKOPS.LOCAL
        SERVICE PRINCIPAL : WEB01$/
        NTLM HASH : 5db7a1891649cef400f8cd6923bb4a69
        AES-256 HASH : 225f9088e80de3f9b69064bf671d89345eca94ee76a87c8f1d0459a4a793af0d
        AES-128 HASH : 99a41017c5243b62d15c9b255be7b40d
```

After gaining root, or even if we can read it as a normal user, we can use credential **web01$: 5db7a1891649cef400f8cd6923bb4a69** to authenticate to domain to have a domain context or enumerate domain information. One example is to use bloodhound-python to collect domain information.

```
└─# bloodhound-python -c ALL -u 'WEB01$@BLACKOPS.LOCAL' --hashes 00000000000000000000000000000000:5db7a1
891649cef400f8cd6923bb4a69 -d blackops.local  -ns 192.168.0.56 --dns-tcp
INFO: Found AD domain: blackops.local
INFO: Connecting to LDAP server: dc.blackops.local
INFO: Found 1 domains
INFO: Found 1 domains in the forest
INFO: Found 7 computers
INFO: Connecting to LDAP server: dc.blackops.local
INFO: Found 13 users
INFO: Found 54 groups
INFO: Found 0 trusts
INFO: Starting computer enumeration with 10 workers
INFO: Querying computer: srv02.blackops.local
INFO: Querying computer: client01.blackops.local
INFO: Querying computer: evilcomputer.blackops.local
INFO: Querying computer: srv01.blackops.local
INFO: Querying computer: web01
INFO: Querying computer: file01.blackops.local
INFO: Querying computer: dc.blackops.local
WARNING: Could not resolve: web01: All nameservers failed to answer the query web01. IN A: Server 192.16
8.0.56 TCP port 53 answered SERVFAIL
INFO: Skipping enumeration for evilcomputer.blackops.local since it could not be resolved.
INFO: Skipping enumeration for file01.blackops.local since it could not be resolved.
INFO: Done in 00M 01S
```

Okay, we have successfully added web01 to domain, we can use the exact same steps to add file01 to domain. Now, we need to deploy vulnerable services/app, and rabbit holes lol. The following table reflect my design.

Port	Service/App	Functionality	Comment
22	SSH	Initial Foothold	Login as mason
25	Postfix SMTP	Send email to user mailadmin	Not useful for exploitation
80	Apache+Wordpress	Get information for foothold	
110	Dovecot POP3	Read email	
143	Dovecot IMAP	Read email	
445	Samba	Rabbit Hole	Backup files of wordpress
5061	Kibana 6.5	Rabbit Hole	RCE vuln
9200	Elasticsearch 6.6	Along with Kibana	

There are multiple apps/services to be installed and configured, you can check following links to follow steps.

Port 22: SSH

Add a line to **/etc/ssh/sshd_configure**:

Denyusers mailadmin

```
root@web01:/etc# cat ssh/sshd_config | grep mailadmin
Denyusers       mailadmin
root@web01:/etc#
```

This step is to deny mailadmin's SSH access, since mailadmin has a weak password. It should be like a service account.

Port 25: Postfix SMTP: https://ubuntu.com/server/docs/mail-postfix

To make it simple, we can **stop at** SMTP Authentication section.

SMTP Authentication

SMTP-AUTH allows a client to identify itself through the SASL authentication mechanism, using Transport Layer Security (TLS) to encrypt the authentication process. Once authenticated the SMTP server will allow the client to relay mail.

To configure Postfix for SMTP-AUTH using SASL (Dovecot SASL), run these commands at a terminal prompt:

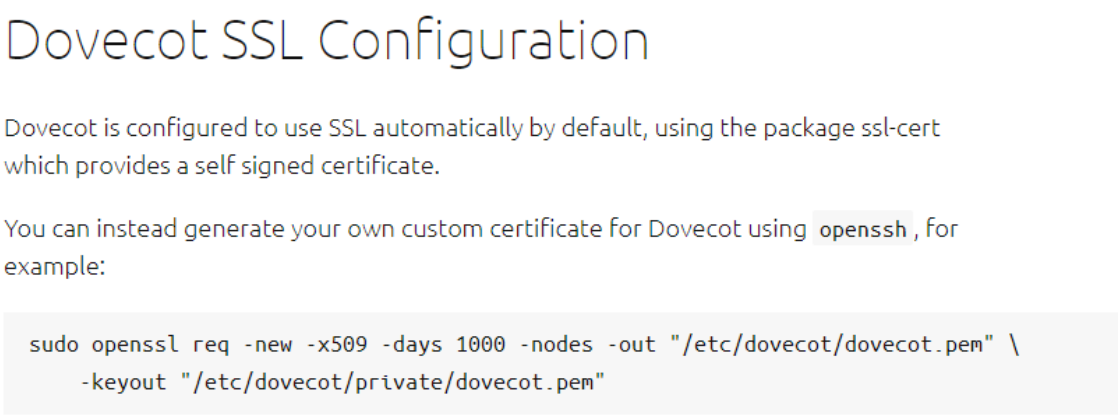
And then, we need to send an email via SMTP, check commands in the screenshot.

```
┌──(root💀os)-[/home/os/Desktop]
└─# nc -nv 192.168.0.51 25
(UNKNOWN) [192.168.0.51] 25 (smtp) open
220 web01 ESMTP Postfix (Ubuntu)
helo hudson@web01
250 web01
mail from: hudson@web01
250 2.1.0 Ok
rcpt to: mailadmin@web01
250 2.1.5 Ok
data
354 End data with <CR><LF>.<CR><LF>
Hey mason, you finally changed your weak SSH and domain password, but please also change mailadmin's pas
sword as well ... By the way, in case you forget your new password, your updated password is CIAAgent1984.
 Please do not forget it ...
.
250 2.0.0 Ok: queued as 812C2E4453
```

So the email will be delivered to mailadmin's inbox.

Port 110 and 143: Dovecot (POP3+IMAP): https://ubuntu.com/server/docs/mail-dovecot

To make it simple, we can **stop at** Dovecot SSL Configuration section.



And we need to allow plaintext authentication to POP3 server, just append two lines to **/etc/dovecot/dovecot.conf**:

disable_plaintext_auth=no

ssl=yes

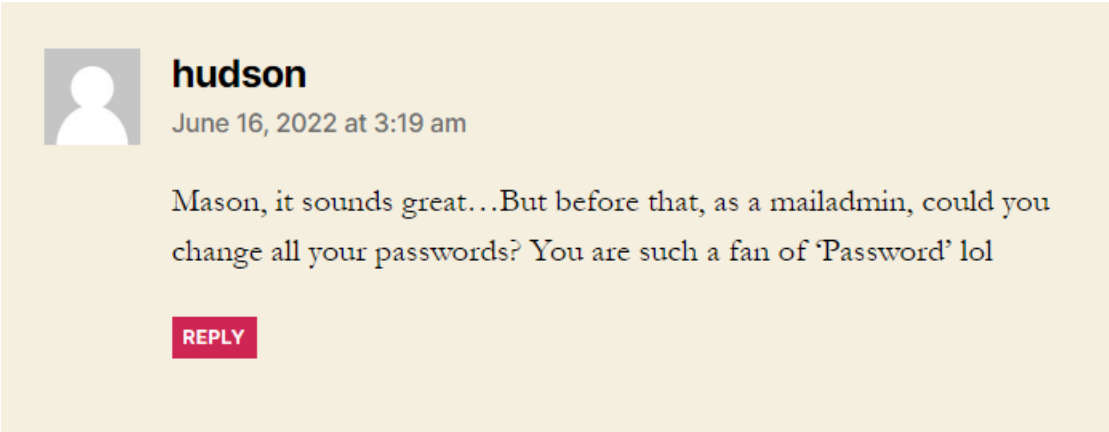
Then we can log in POP3 server, otherwise we cannot authenticate to POP3 server.

Port 80: Wordpress: https://ubuntu.com/tutorials/install-and-configure-wordpress#1-overview

It is simple, just follow steps in this link. After completing the installation, register 2 users: mason, hudson.

Log in as mason, and post an article like this:

Then log in as hudson, leave a comment. This is an indicator that mason manages mailadmin account, and this account has weak password: Password. After that, log in as mason or admin to approve hudson's comment. Otherwise, hudson's comment will not be displayed.



Port 445: Samba

Create a new folder /var/backups/www, and copy /var/www/html/wordpress to the new folder, and create a new share to map to this folder.

```
226    create mask = 0700
227
228 [webapp]
229    comment = Web app files
230    path = /var/backups/www/html
231    writable = yes
232    guest ok = yes
233    browseable = yes
234    create mask = 0777
235    directory mask = 0777
236
237 # Windows clients look for this share name as a
238 # printer drivers
```

Do not forget to assign proper ownership and permission, otherwise the attacker cannot upload or read a file, so he will not upload a shell and fall into the rabbit hole lol

```
root@web01:/etc# ls -al /var/backups/www/html/wordpress/
total 228
drwxr-xr--  5 www-data www-data  4096 Jun 15 23:24 .
drwxrwxrwx  3 www-data www-data  4096 Jun 15 23:27 ..
-rw-r--r--  1 root     root       543 Jun 15 23:24 .htaccess
-rw-r--r--  1 www-data www-data   405 Jun 15 23:24 index.php
-rw-r--r--  1 www-data www-data 19915 Jun 15 23:24 license.txt
-rw-r--r--  1 www-data www-data  7401 Jun 15 23:24 readme.html
-rw-r--r--  1 www-data www-data  7165 Jun 15 23:24 wp-activate.php
drwxr-xr--  9 www-data www-data  4096 Jun 15 23:24 wp-admin
-rw-r--r--  1 www-data www-data   351 Jun 15 23:24 wp-blog-header.php
-rw-r--r--  1 www-data www-data  2338 Jun 15 23:24 wp-comments-post.php
-rw-r--r--  1 www-data www-data  2487 Jun 15 23:24 wp-config.php
-rw-r--r--  1 www-data www-data  3001 Jun 15 23:24 wp-config-sample.php
drwxr-xr--  5 www-data www-data  4096 Jun 15 23:24 wp-content
-rw-r--r--  1 www-data www-data  3943 Jun 15 23:24 wp-cron.php
drwxr-xr-- 26 www-data www-data 12288 Jun 15 23:24 wp-includes
-rw-r--r--  1 www-data www-data  2494 Jun 15 23:24 wp-links-opml.php
-rw-r--r--  1 www-data www-data  3973 Jun 15 23:24 wp-load.php
-rw-r--r--  1 www-data www-data 48498 Jun 15 23:24 wp-login.php
-rw-r--r--  1 www-data www-data  8577 Jun 15 23:24 wp-mail.php
-rw-r--r--  1 www-data www-data 23706 Jun 15 23:24 wp-settings.php
-rw-r--r--  1 www-data www-data 32051 Jun 15 23:24 wp-signup.php
-rw-r--r--  1 www-data www-data  4748 Jun 15 23:24 wp-trackback.php
-rw-r--r--  1 www-data www-data  3236 Jun 15 23:24 xmlrpc.php
root@web01:/etc#
```

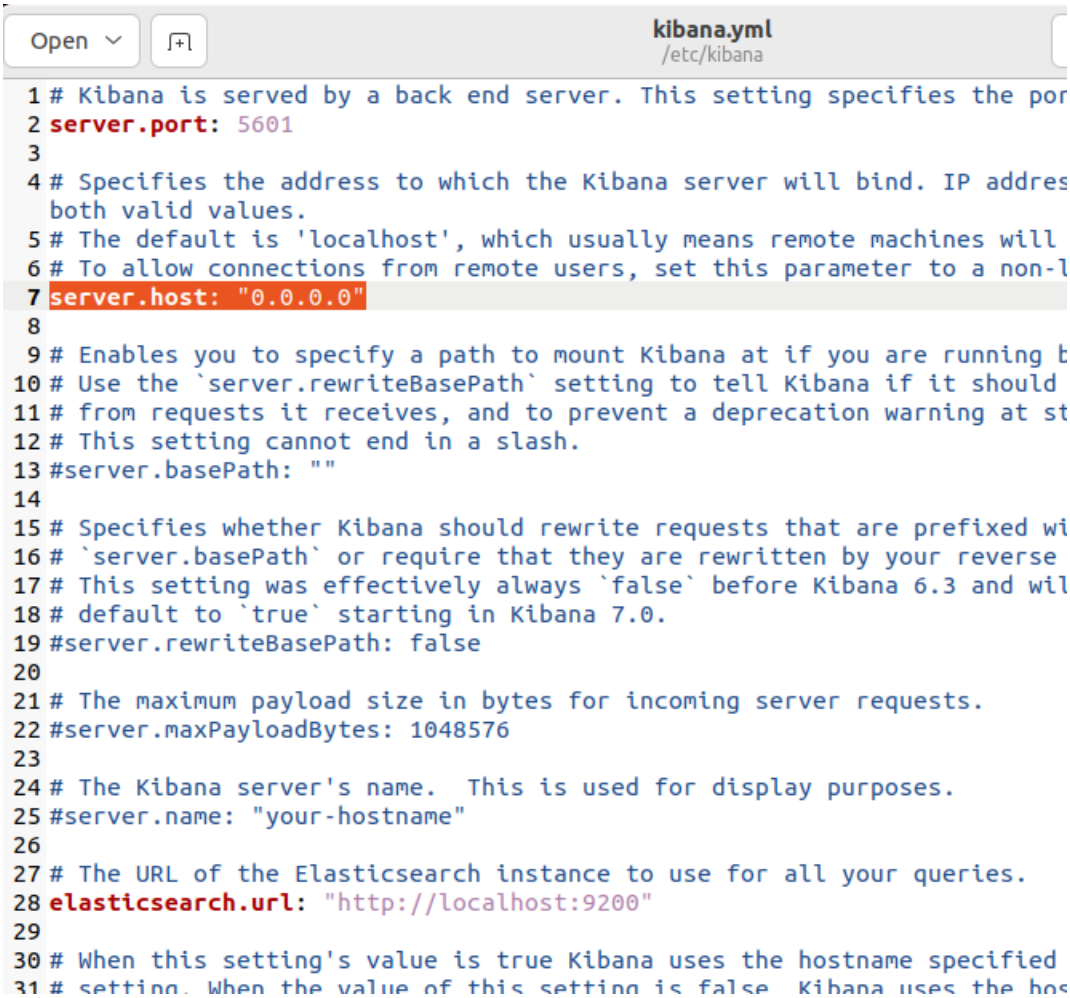
```
smb: \wordpress\> put hash.txt
putting file hash.txt as \wordpress\hash.txt (10.7 kb/s) (average 10.7 kb/s)
smb: \wordpress\> del hash.txt
smb: \wordpress\>
```

Port 5601: Kibana 6.5

Download link: https://www.elastic.co/cn/downloads/past-releases/kibana-6-5-0

Download **deb 64-bit**, and then use dpkg to install it, it is very simple.

But do not forget to edit /etc/kibana/kibana.yml to uncomment few lines and change server.hot to 0.0.0.0.



This version of kibana is vulnerable to a RCE vulnerability, you can find the PoC here: https://github.com/mpgn/CVE-2019-7609

Follow the steps, and you can get a shell as kibana. But unfortunately, there is no intended privilege escalation vector for user kibana, though I am not sure if all Nday vulnerabilities have been fixed. Therefore, it is a rabbit hole.

Port 9200: Elasticsearch 6.6

Download and install Elasticsearch 6.6 from https://www.elastic.co/cn/downloads/past-releases/elasticsearch-6-6-0 like how we installed Kibana, but we do not need to customize it.

Now, we almost finished. The last step is to grant mason a privilege to execute find with sudo permission without password. So only user mason can escalate

our self to root and read /etc/krb5.keytab.

```
root@web01:/home/mason/Desktop# cat /etc/sudoers | grep mason
mason ALL = NOPASSWD: /usr/bin/find
root@web01:/home/mason/Desktop#
```

After knowing that alex.mason is a domain user, we should be aware that linux local user mason could share the same password with domain user alex.mason, so we can use SSH to move to file01 as alex.mason@blackops.local.

Linux Domain Computer 2

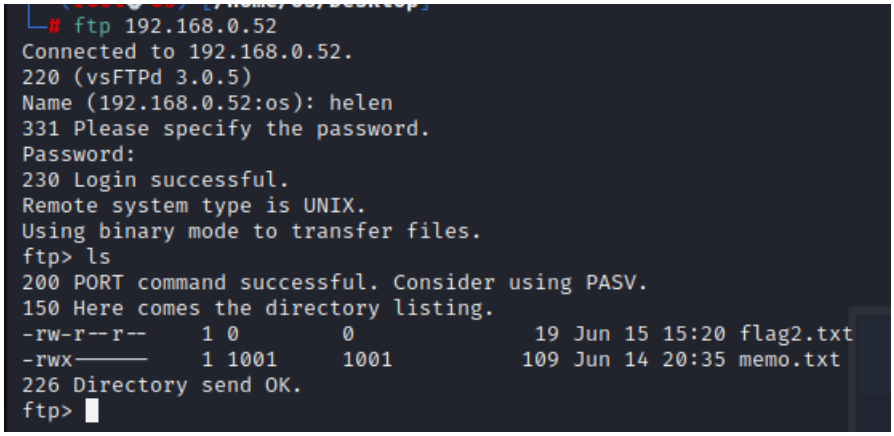
file01.blackops.local

This linux machine is easier to configure. First, we need to set DC's IP as DNS, and join file01 to domain, just as we previously did. We only need to configure FTP and add one user.

Port	Service/App	Functionality	Comment
21	Vsftpd FTP	Let helen to connect to	Get helen's plaintext password
22	SSH	Initial Foothold	Use alex.mason's domain account to access

There is nothing too much to configure FTP. Use apt to install vsftpd. Then add helen as a linux local user. When we can use helen's credential to authenticate to FTP server.

Account	Password	Functionality	Comment
ubuntu	Strong Password	Privileged account	Not for exploitation
helen	Summer2022!	Credential reuse	Reflection of domain password



Many people only care about how to become root, and this is the reason why I make privilege escalation simple, I set multiple common binaries (cat, nc, find, etc.) SUID permission, and I also set tcpdump SUID. If check memo.txt, we can know that Helen keeps authenticating to FTP server. Since FTP does not have encryption, so we can use tcpdump to capture plaintext credential.

```
ubuntu@file01:/home/helen$ cat memo.txt
Just a memo...Don't forget to create a powershell script on Client host to check
 FTP folder automatically...
ubuntu@file01:/home/helen$
```

```
ubuntu@file01:/home/helen$ tcpdump -i ens33 dst port 21
tcpdump: verbose output suppressed, use -v[v]... for full protocol decode
listening on ens33, link-type EN10MB (Ethernet), snapshot length 262144 bytes
16:09:00.086622 IP 192.168.0.53.49379 > file01.blackops.local.ftp: Flags [S], se
q 3315858485, win 8192, options [mss 1460,nop,wscale 0,nop,nop,sackOK], length 0
16:09:00.089840 IP 192.168.0.53.49379 > file01.blackops.local.ftp: Flags [.], ac
k 73629535, win 8192, length 0
16:09:00.100436 IP 192.168.0.53.49379 > file01.blackops.local.ftp: Flags [P.], s
eq 0:14, ack 21, win 8172, length 14: FTP: OPTS UTF8 ON
16:09:00.104010 IP 192.168.0.53.49379 > file01.blackops.local.ftp: Flags [P.], s
eq 14:26, ack 47, win 8146, length 12: FTP: USER helen
16:09:00.107428 IP 192.168.0.53.49379 > file01.blackops.local.ftp: Flags [P.], s
eq 26:44, ack 81, win 8112, length 18: FTP: PASS Summer2022!
16:09:00.183706 IP 192.168.0.53.49379 > file01.blackops.local.ftp: Flags [.], ac
k 128, win 8065, length 0
16:09:00.183707 IP 192.168.0.53.49379 > file01.blackops.local.ftp: Flags [.], ac
k 129, win 8065, length 0
16:09:00.194885 IP 192.168.0.53.49379 > file01.blackops.local.ftp: Flags [F.], s
eq 44, ack 129, win 8065, length 0
```

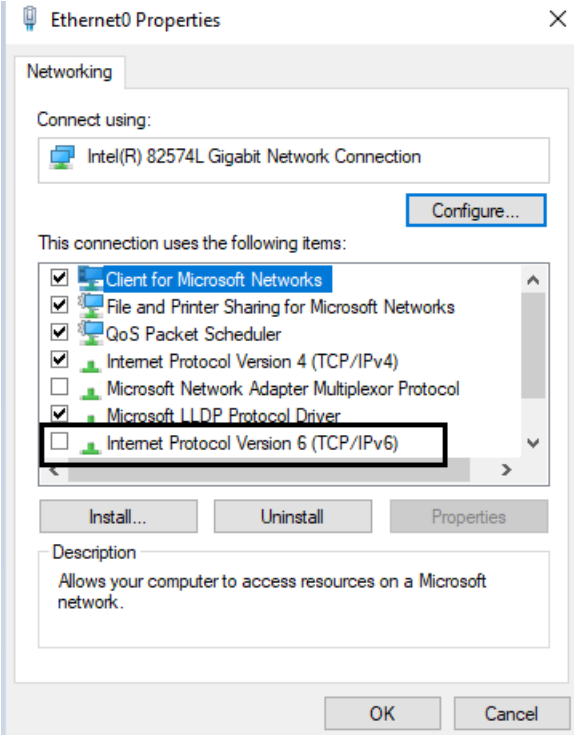
We can clearly see that the credential is helen:Summer2022!. Since helen.park is a domain user in BLACKOPS.LOCAL, so credential reuse is possible, we should be aware of that. So we completed configurations of file01.

Client Server

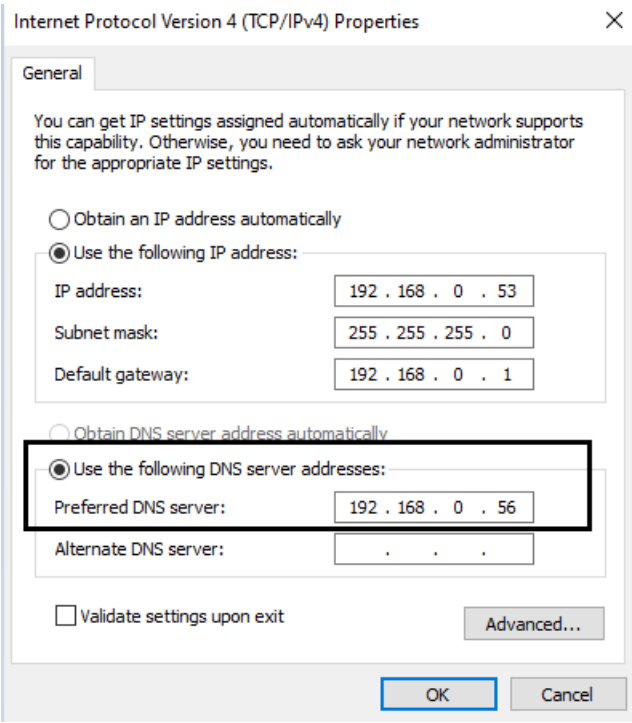
client01.blackops.local

Now we successfully configured all Linux domain computers. Let's configure the client server client01.

The first step is still configuring DNS. But we also need to disable IPv6.



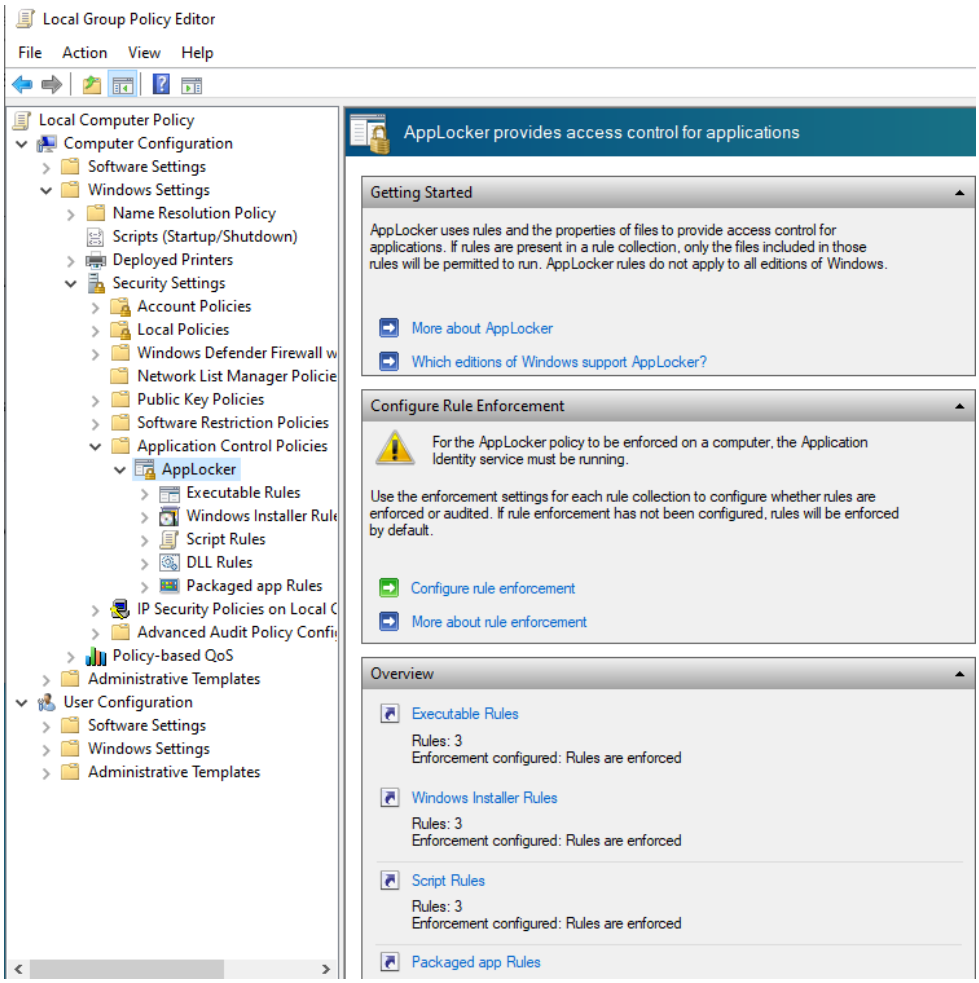
And set DC as DNS server.

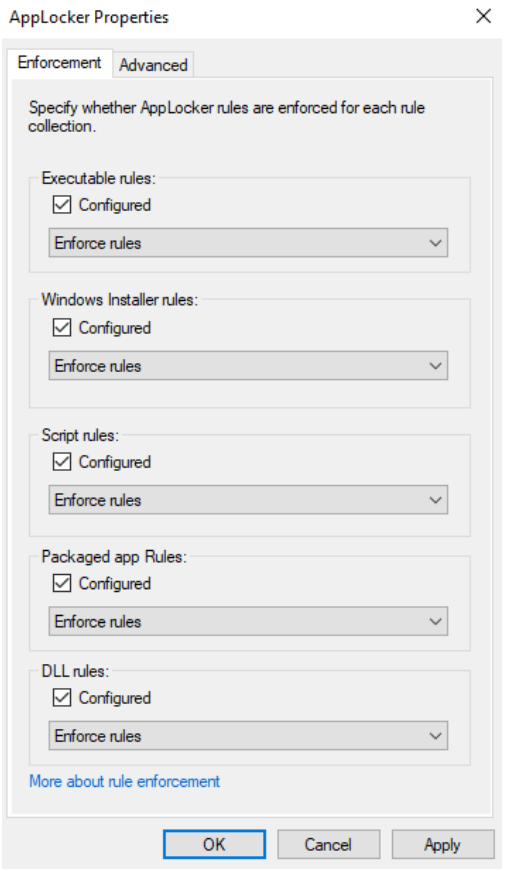


We do not need to configure any app or services on client01, but some common settings on Windows hosts.

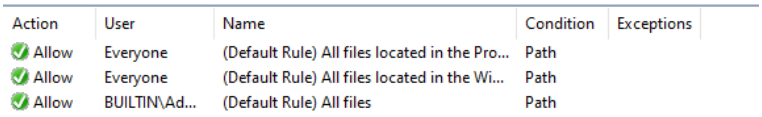
AppLocker

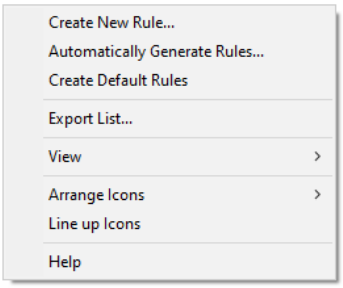
Run Local Group Policy Editor, enable DLL rules, and enforce all types of rules.





Enforcing default rules is okay, even though some paths can be abused to execute binary such as C:\windows\tasks. So it is not necessary to create a custom tradecraft or download bypass-clm (https://github.com/calebstewart/bypass-clm) from github.





Windows Defender

Just use the default settings.

Firewall

I turn off firewall on all windows machines. You could turn on it if you would like to increase a little more difficulty: D

Autologin

Set autologin for domain user helen.park.

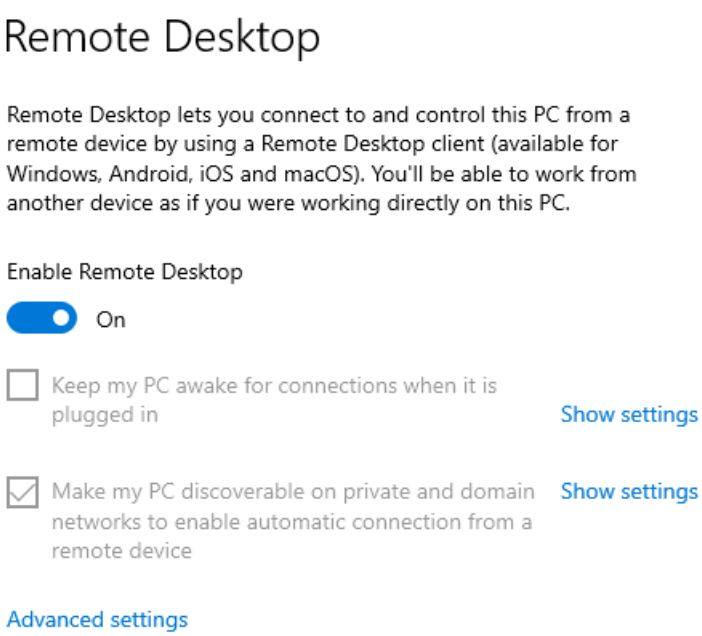
UAC

I don't think UAC bypass is needed in the whole exploitation process, so just leave it default.

Remote Desktop

Enable Remote Desktop setting, and add helen.park to localgroup Remote Desktop Users: **net localgroup "Remote Desktop Users" helen.park /add**

But just as I previously said, we can also achieve this by linking and enforcing a GPO.



```
C:\Users\admin>net localgroup "Remote Desktop Users"
Alias name     Remote Desktop Users
Comment        Members in this group are granted the right to logon remotely

Members

-------------------------------------------------------------------------------
BLACKOPS\helen.park
The command completed successfully.
```

Then we need to create a script to connect to file01's FTP server as helen, and two txt file as well.

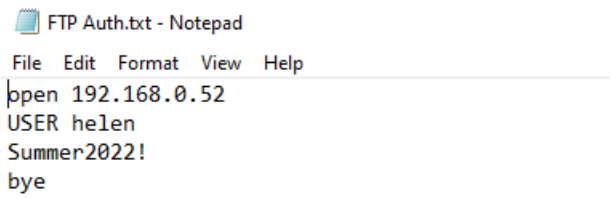
txt file 1: **FTP Auth.txt**

open 192.168.0.52

USER helen

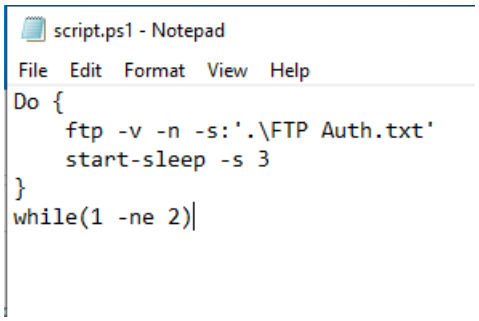
Summer2022!

bye



script: **script.ps1**

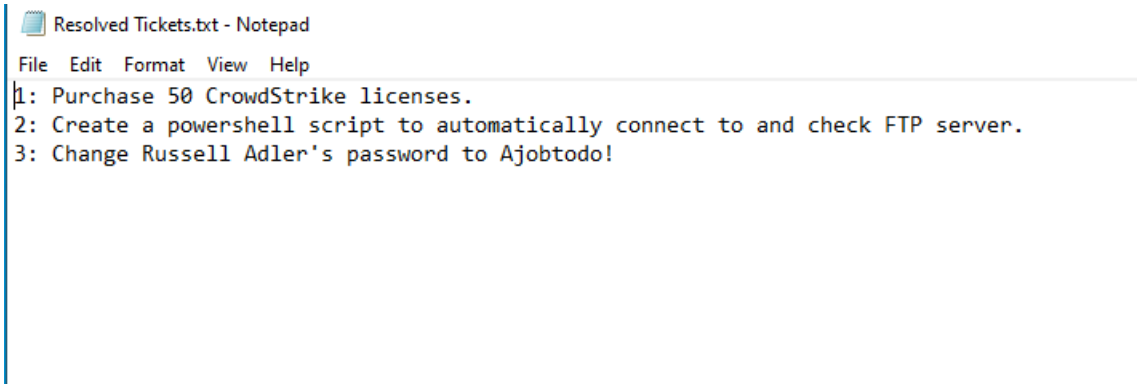
```
Do {
   ftp -v -n -s:'.\FTP Auth.txt'
   start-sleep -s 3
   }
while (1 -ne 2)
```

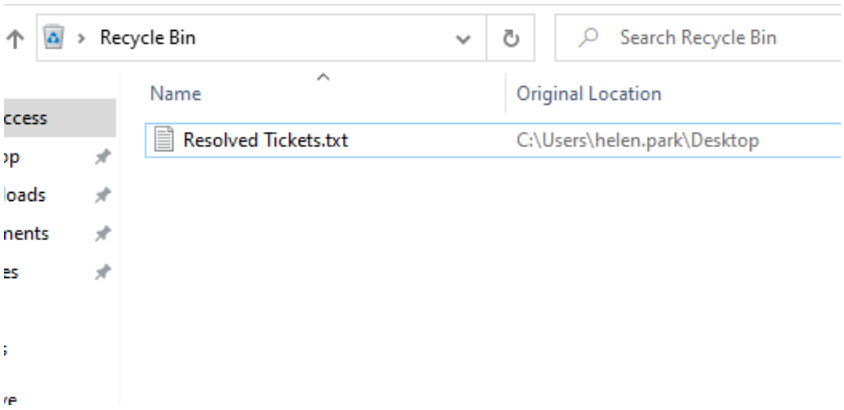


Put these 2 files on helen's document folder.

Then create another txt file on helen's desktop: Resolved Ticket.txt

After finishing editing, just delete it. I just want people not to forget to check Recycle Bin during enumeration.





So we complete the configurations here. Let's move to SRV01.

Server 1

srv01.blackops.local

So we move to the most difficult part of design and configurations. Fortunately, most steps are the same for both SRV01 and SRV02.

First, disable IPv6, then configure IP and DNS.

Do not configure autologin.

Configure AppLocker just as we did on client01.

Add jason.hudson to local group “Remote Management Users”: **net localgroup “Remote Management Users” jason.hudson /add**

Add svc_sql to local group “Remote Desktop Users”: **net localgroup “Remote Desktop Users” svc_sql /add**

By this way, jason.hudson has WinRM access to SRV01.

```
┌──(root💀os)-[/home/os/Desktop]
└─# evil-winrm -i 192.168.0.56 -u blackops.local\\jason.hudson -p jkhnrjk2020!

Evil-WinRM shell v2.4

Info: Establishing connection to remote endpoint

Error: An error of type WinRM::WinRMAuthorizationError happened, message is WinRM::WinRMAuthoriza
or

Error: Exiting with code 1


┌──(root💀os)-[/home/os/Desktop]
└─# evil-winrm -i 192.168.0.55 -u blackops.local\\jason.hudson -p jkhnrjk2020!

Evil-WinRM shell v2.4

Info: Establishing connection to remote endpoint

Error: An error of type WinRM::WinRMAuthorizationError happened, message is WinRM::WinRMAuthoriza
or

Error: Exiting with code 1


┌──(root💀os)-[/home/os/Desktop]
└─# evil-winrm -i 192.168.0.54 -u blackops.local\\jason.hudson -p jkhnrjk2020!

Evil-WinRM shell v2.4

Info: Establishing connection to remote endpoint

*Evil-WinRM* PS C:\Users\jason.hudson\Documents>
```

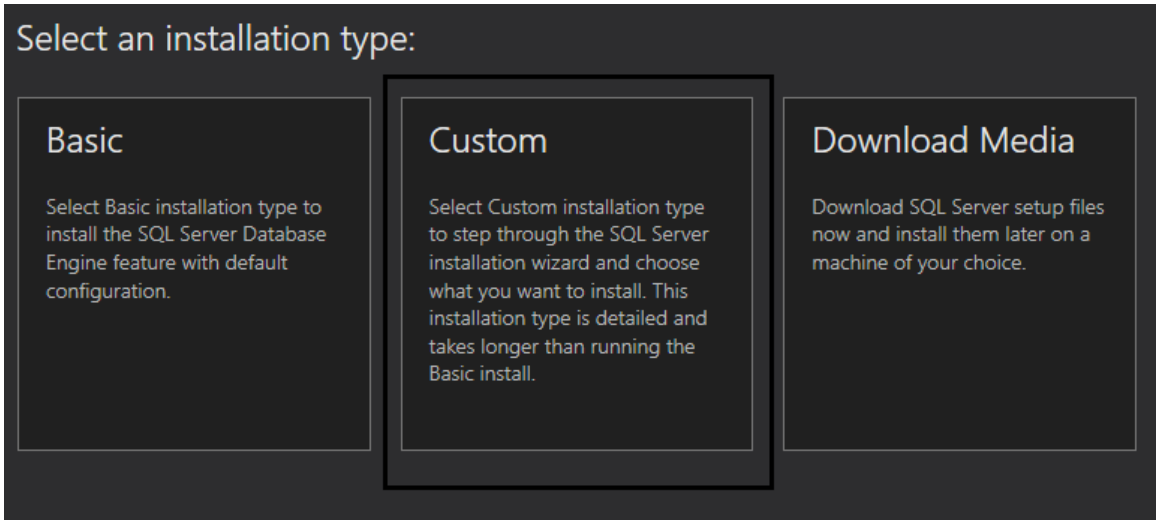
Beside, let’s configure a privilege escalation vector: **AlwaysInstallElevated**. It is very simple, we can check this and just add 2 register key: https://www.ibm.com/docs/en/db2/9.7?topic=prerequisites-setting-up-elevated-privileges-windows.

Now let’s install and configure SQL Server 2019, it is the most difficult and complex part.

Download link: https://www.microsoft.com/en-us/sql-server/sql-server-downloads (Developer)

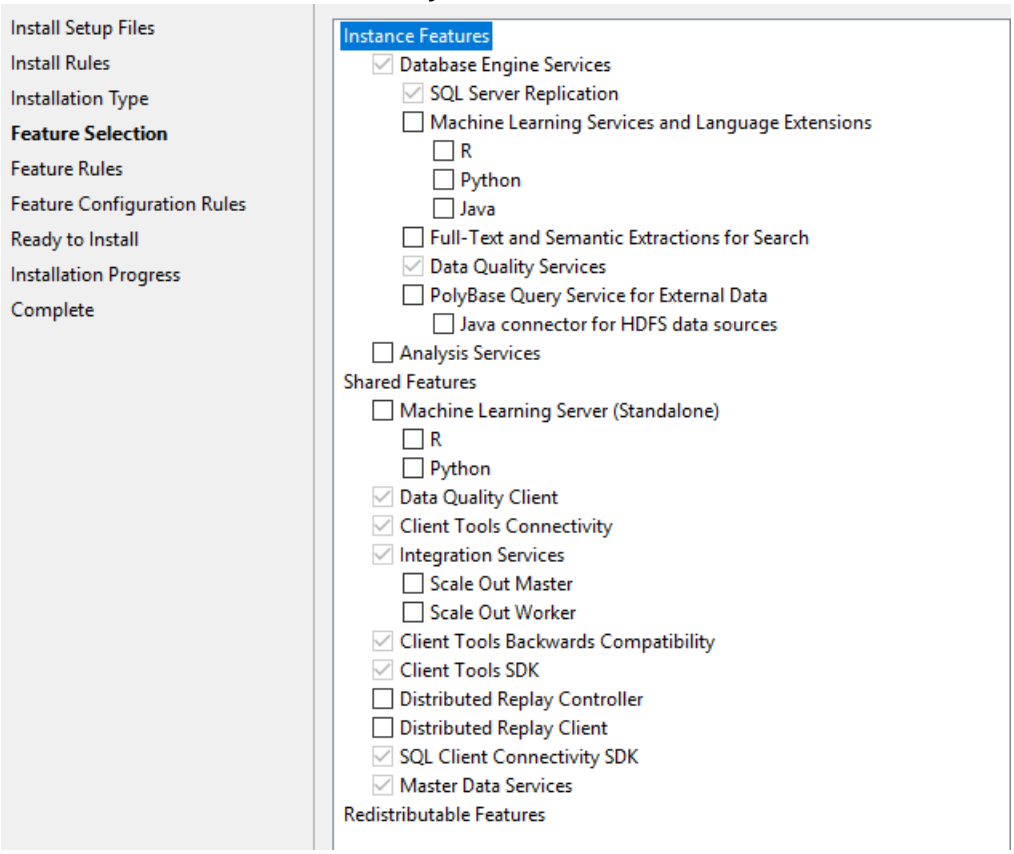
SSMS: https://docs.microsoft.com/en-us/sql/ssms/download-sql-server-management-studio-ssms?view=sql-server-ver16

Install Windows SQL Server 2019 first, something important is that choose Customize Installation, because Basic Installation cannot meet our requirements.

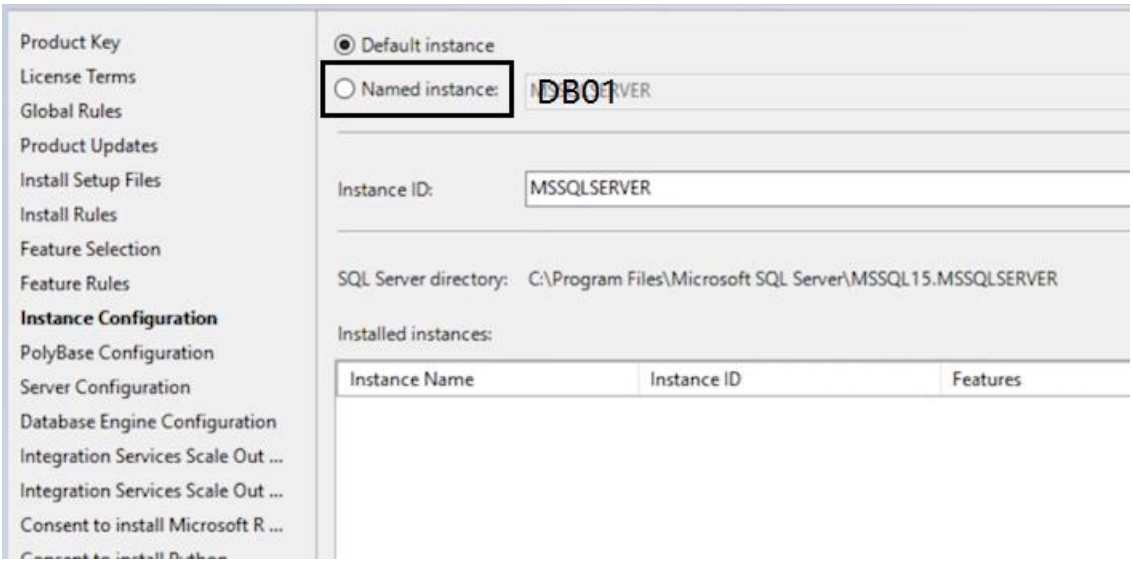


During installation, we can leave most pages default, but something needs customization. When selecting Feature, I cannot tell the minimum selections to

make the AD set works, but my selections work well.

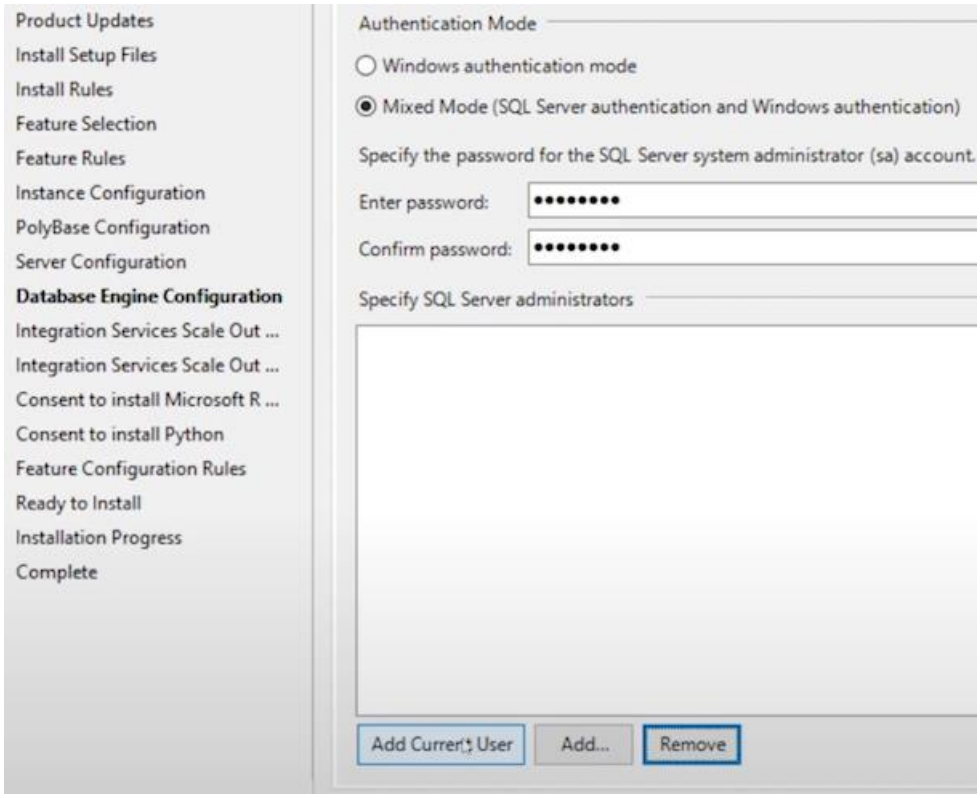


Then you need to specify instance name, to make it in line with my settings, you can change instance name to DB01.



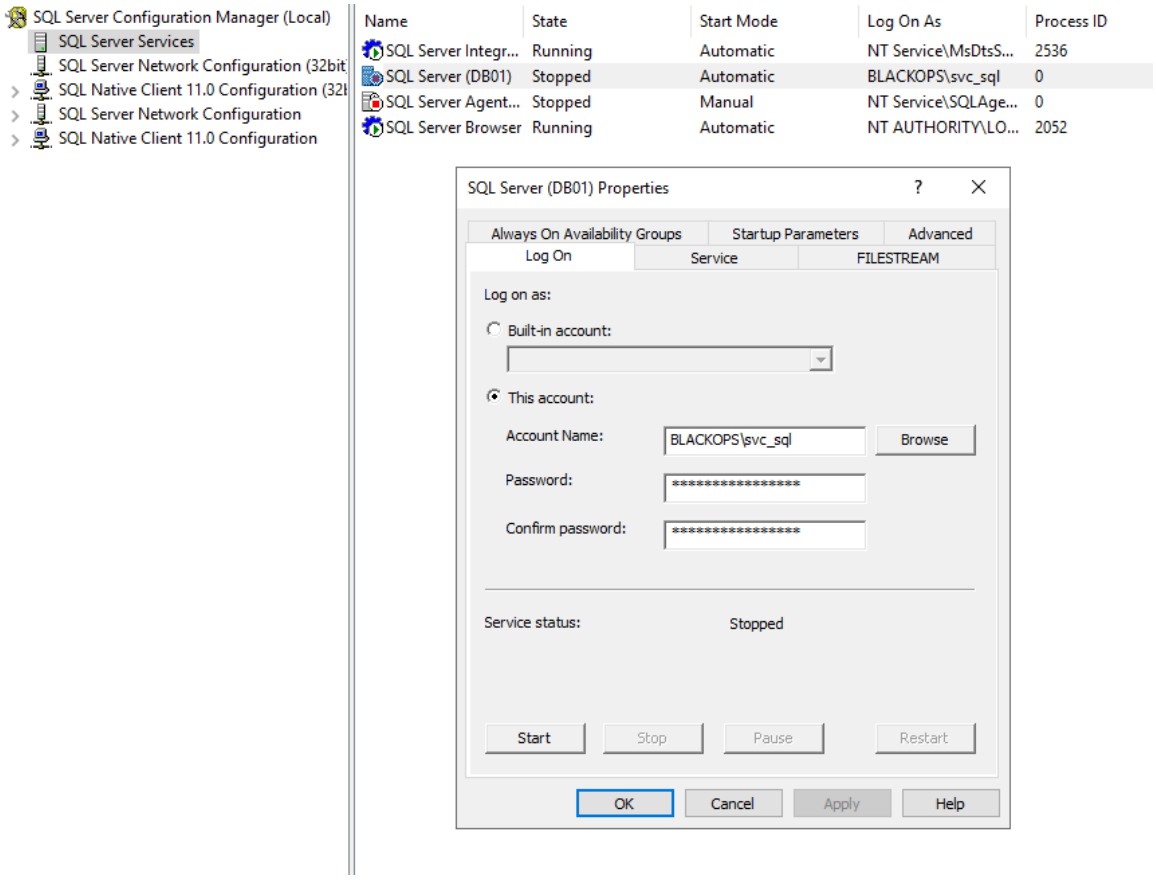
Next, leave service accounts default. When configuring Authentication Mode, choose Mixed Mode. After that, it is recommended to click Add Current User

button to add current local admin to sysadmin. By this way, both sa and local admin account have sysadmin privilege.

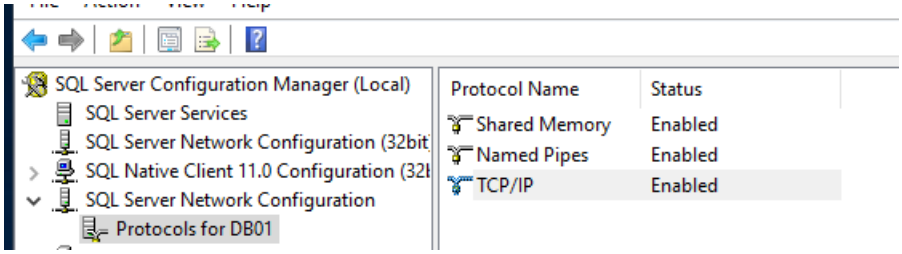


After setting this, we can leave left default and complete the installation. Installing SSMS is simple, we do not need to customize something.

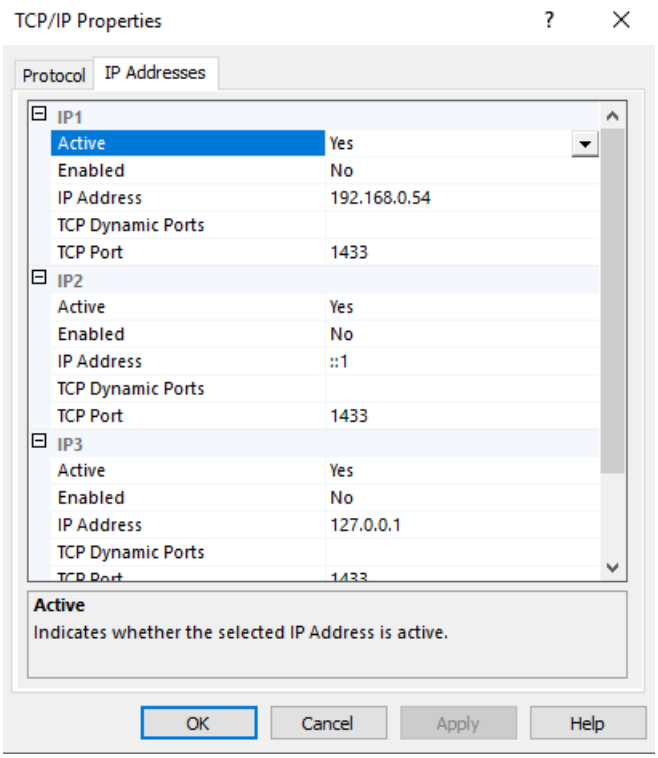
Run **Sql Server Configuration Manager**, we need to modify few settings. First, click **SQL Server Services -> SQL Server (DB01)**, then select **Log On** tab, change logon account to BLACKOPS\svc_sql, type the correct password. Then, we could need a restart of SQL service.



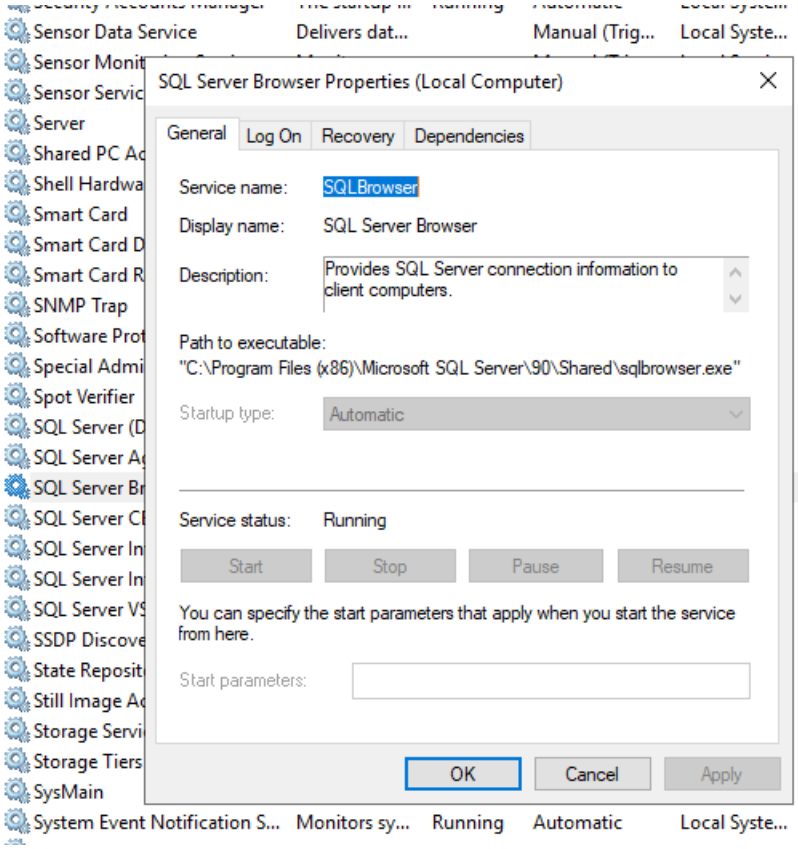
Second, click **SQL Server Network Configuration -> Protocols for DB01 -> TCP/TP**, enable it.



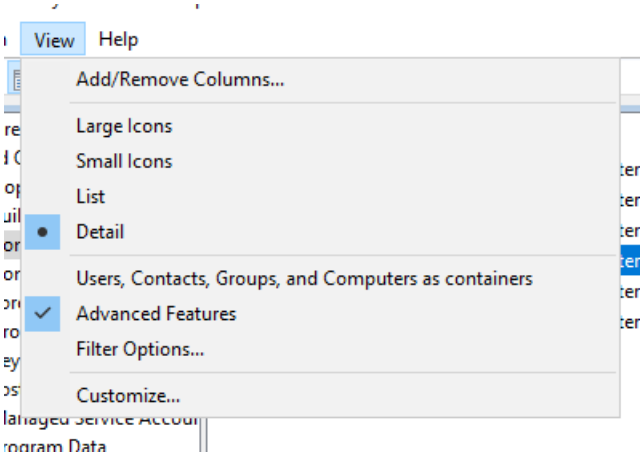
Then double click it, select **IP Address** tab, leave all **TCP Dynamic Ports** blank, and set all **TCP Port** to 1433.



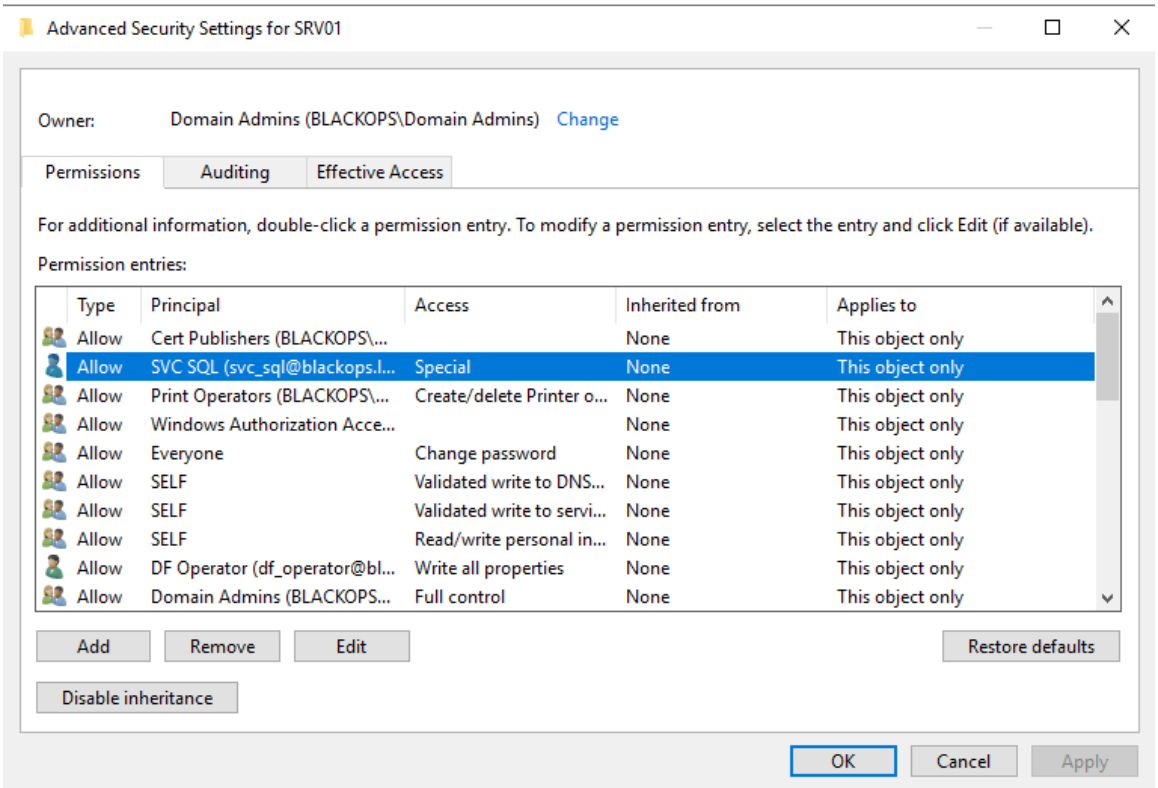
Why we need to disable dynamic ports? Because we will set SPN for svc_sql to make SQL Server supports Kerberos authentication. We also need to set **Start Type** of service **SQL Server Browser** to **Automatic**.



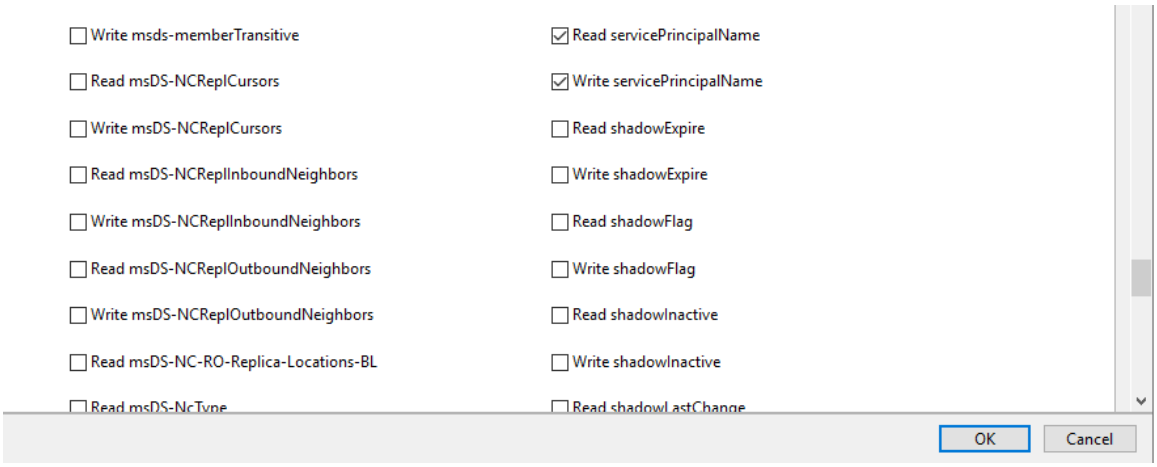
Then, let's revisit to DC to do some configurations. Run Active Directory Users and Computers, check **Advanced Features**.

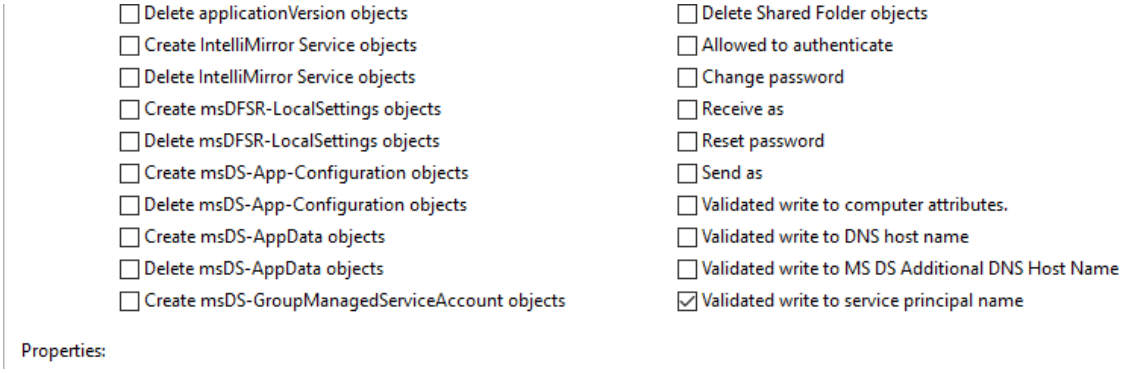


Double click SRV1 (Same steps for SRV02), click **Security** tab and **Advanced** button, add a new permission for svc_sql on SRV1.



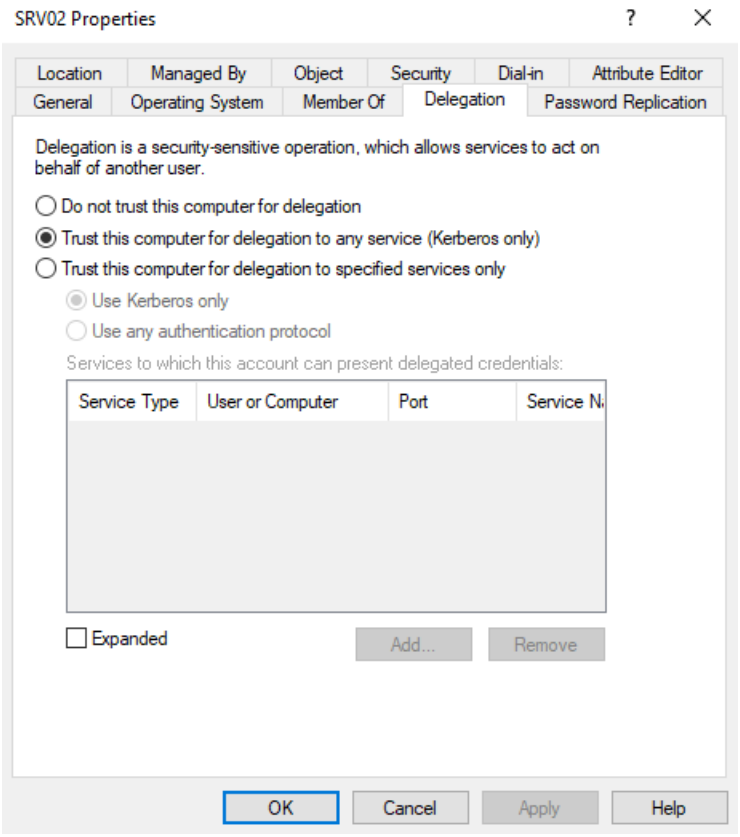
Select principal as svc_sql, apply this permission on this object only. Clear all default check, but check **Read servicePrincipalName**, **Write servicePrincipalName** properties, and **Validated write to service principal name** permission. This official document explains well: https://docs.microsoft.com/en-us/sql/database-engine/configure-windows/register-a-service-principal-name-for-kerberos-connections?view=sql-server-ver16.



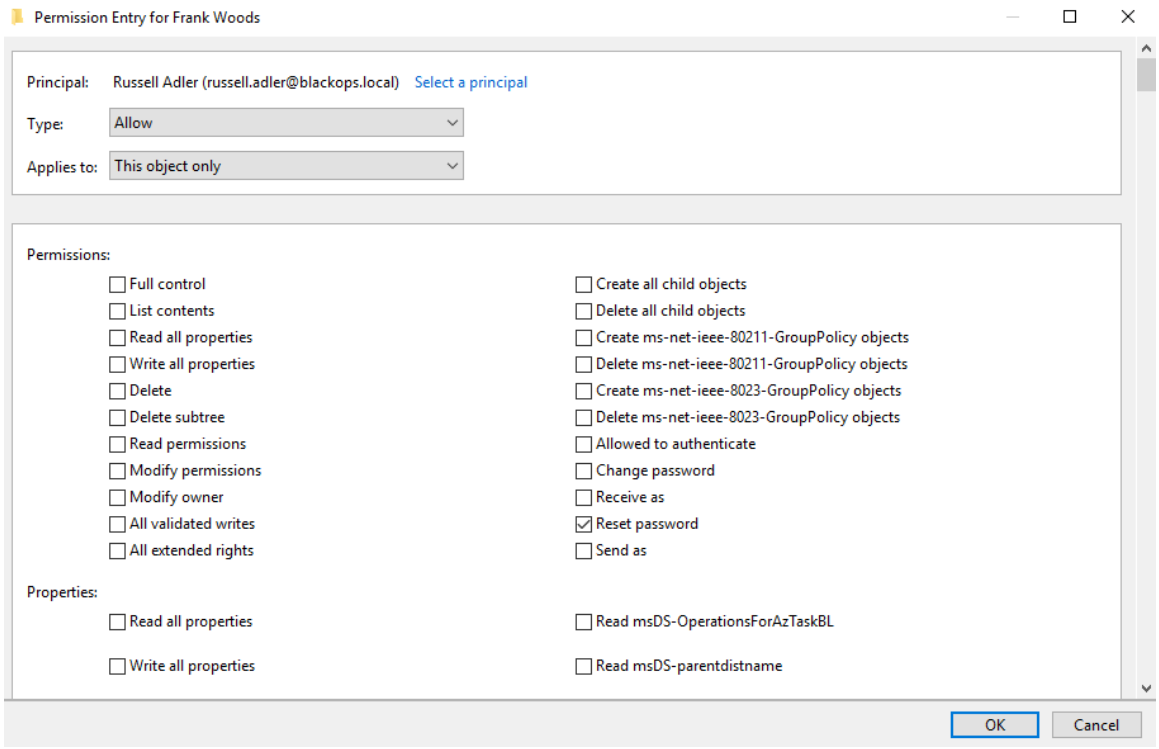


By the way, let's configure DACL and delegation.

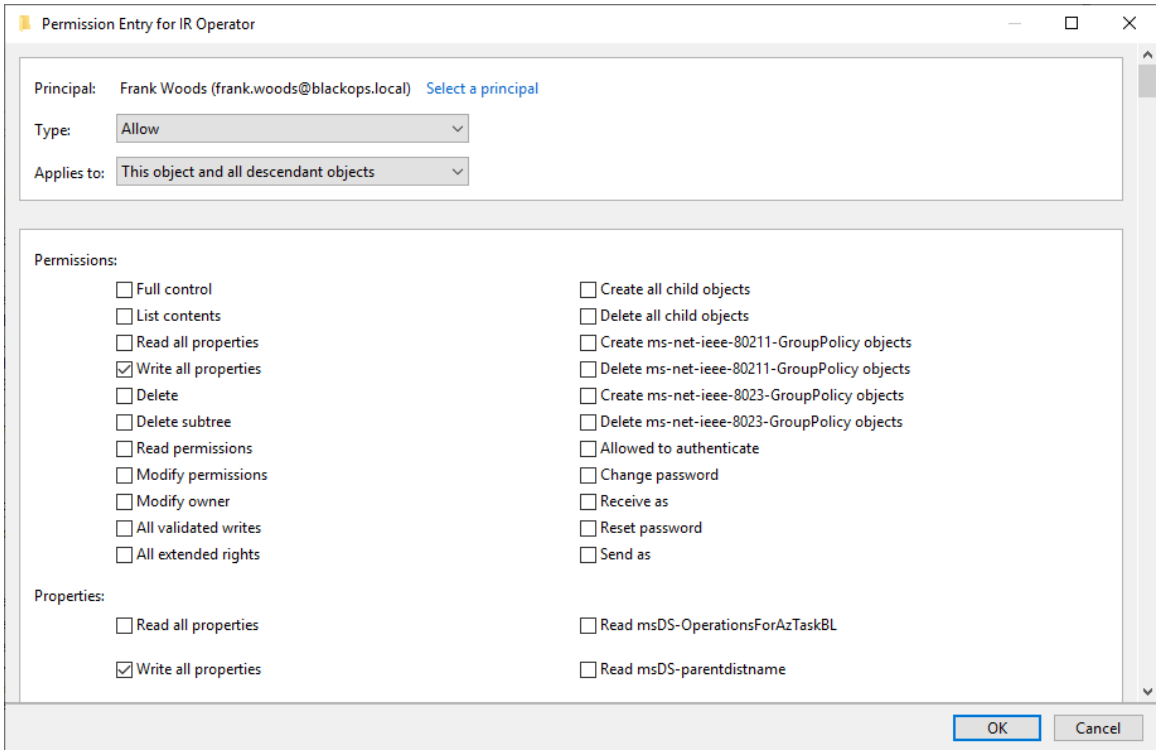
SRV02 is set **unconstrained delegation**.



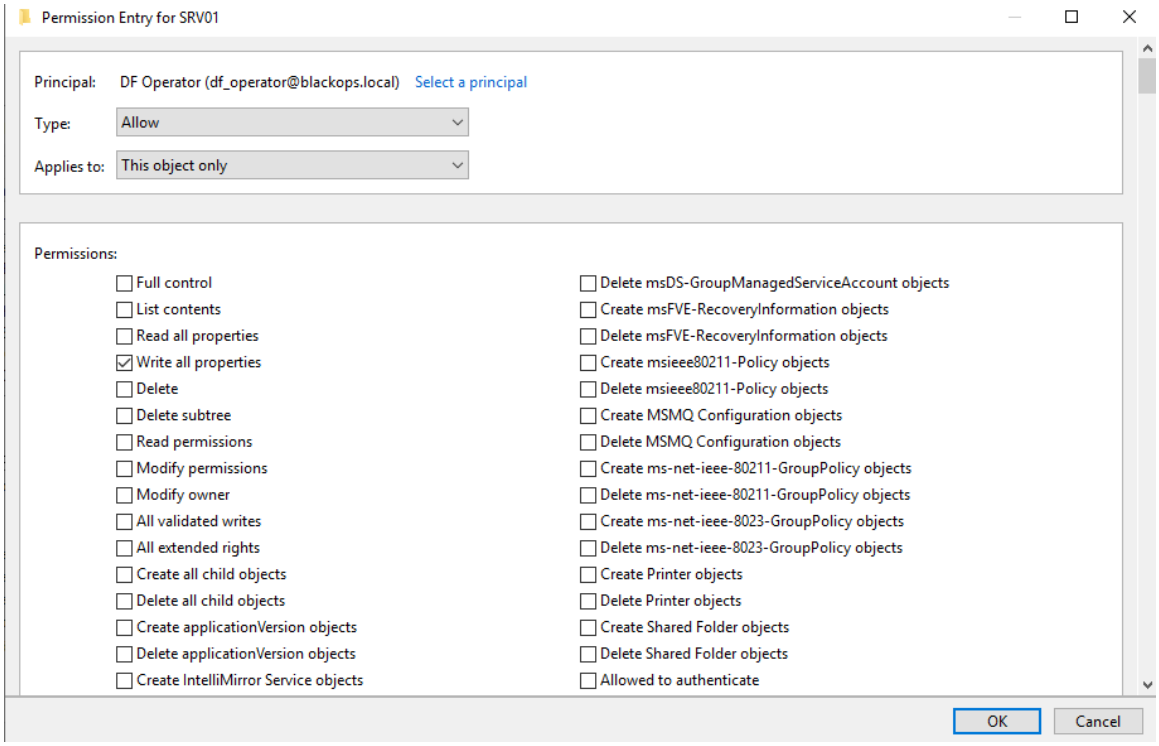
russell.adler has **ForceChangePassword** permission on frank.woods



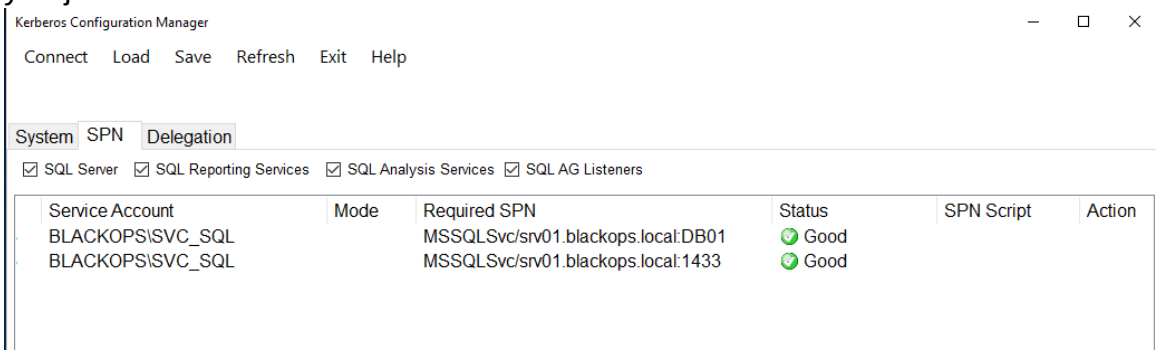
frank.woods has **GeneticWrite** permission on ir_operator



df_operator has **GenericWrite** permission on SRV01

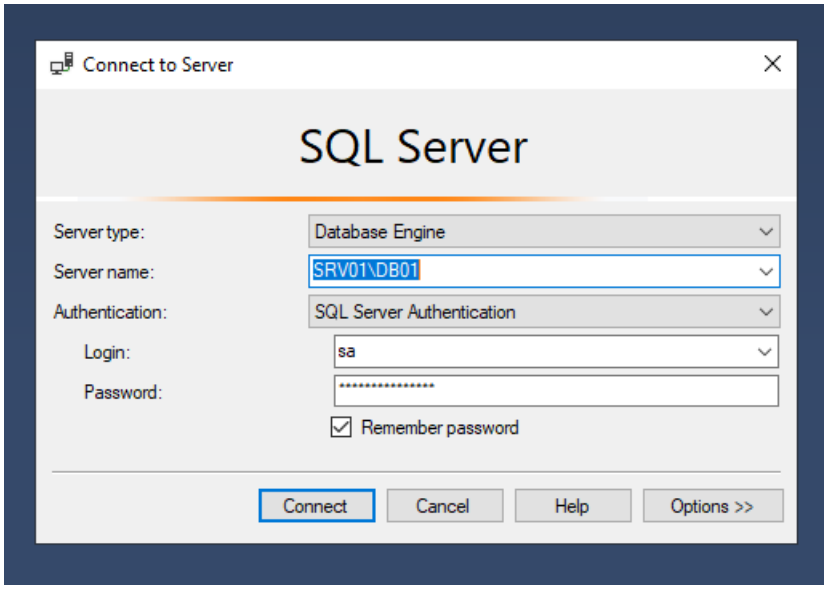


Cool, all set! Back to SRV01, download a tool from https://www.microsoft.com/en-us/download/details.aspx?id=39046 to help us set SPN automatically. If we did not set proper SPN, it helps us correct it as well. After installing it, run it and connect to the instance, no need to provide any credential. Since we configured proper SPNs, so we do not have to make any change. But if you did not configure SPNs properly, the tool will warn you and you just need to click Fix button.

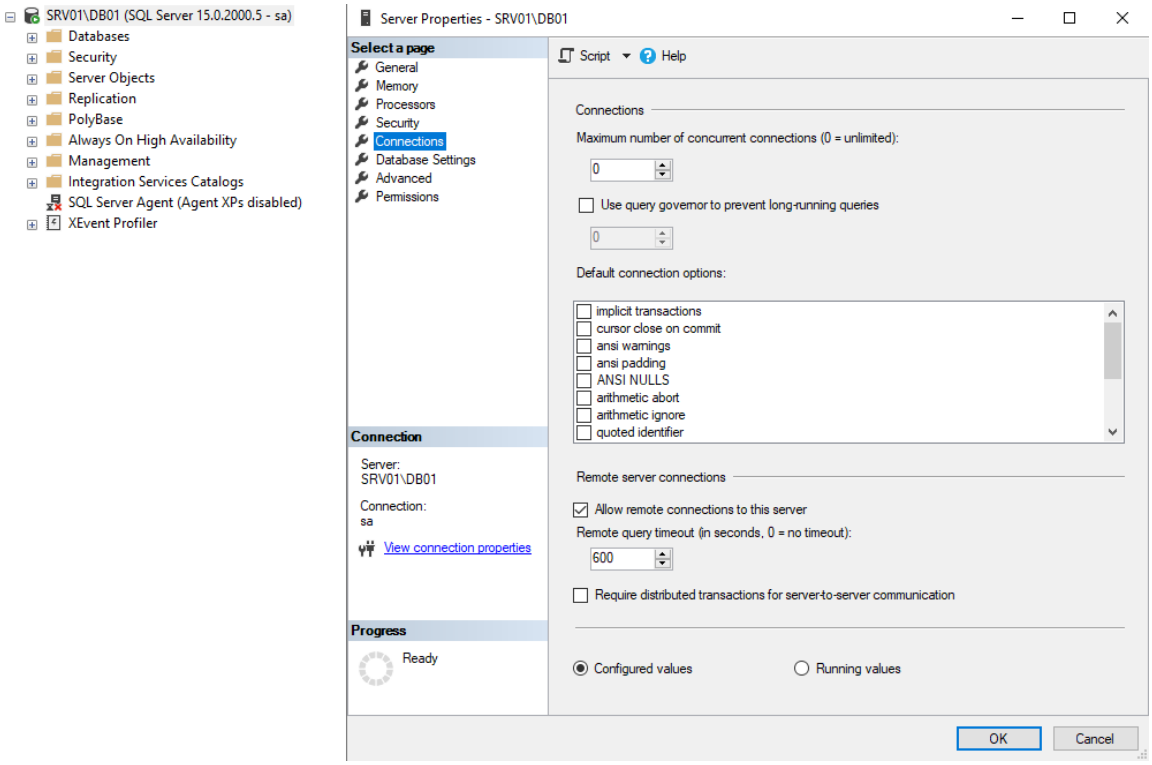


Now, I believe we successfully set SPN and configure Kerberos authentication for SQL instance. But since the process is complex, I cannot make sure if I miss something. If you follow my steps and cannot reproduce it successfully, please let me know.

Then, run SSMS 2018, which we installed previously. Change Server name to SRV01\DB01 and select SQL Server Authentication, connect.



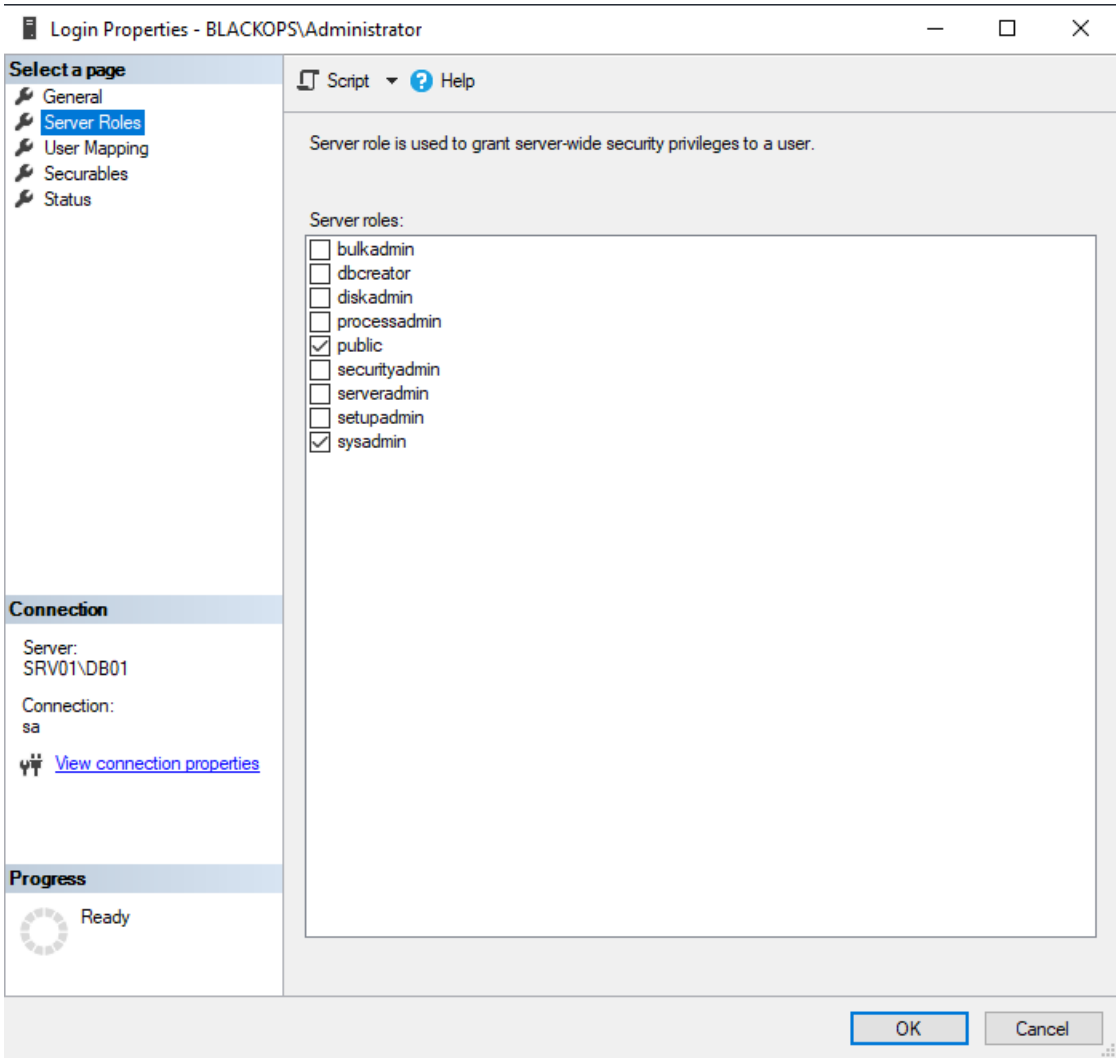
Check SRV01\DB01's property, make sure **Allow remote connections to this server** is checked.



Then, we need to add few logins.

BLACKOPS\Administrator: Sysadmin

Not required, just to make it more realistic.

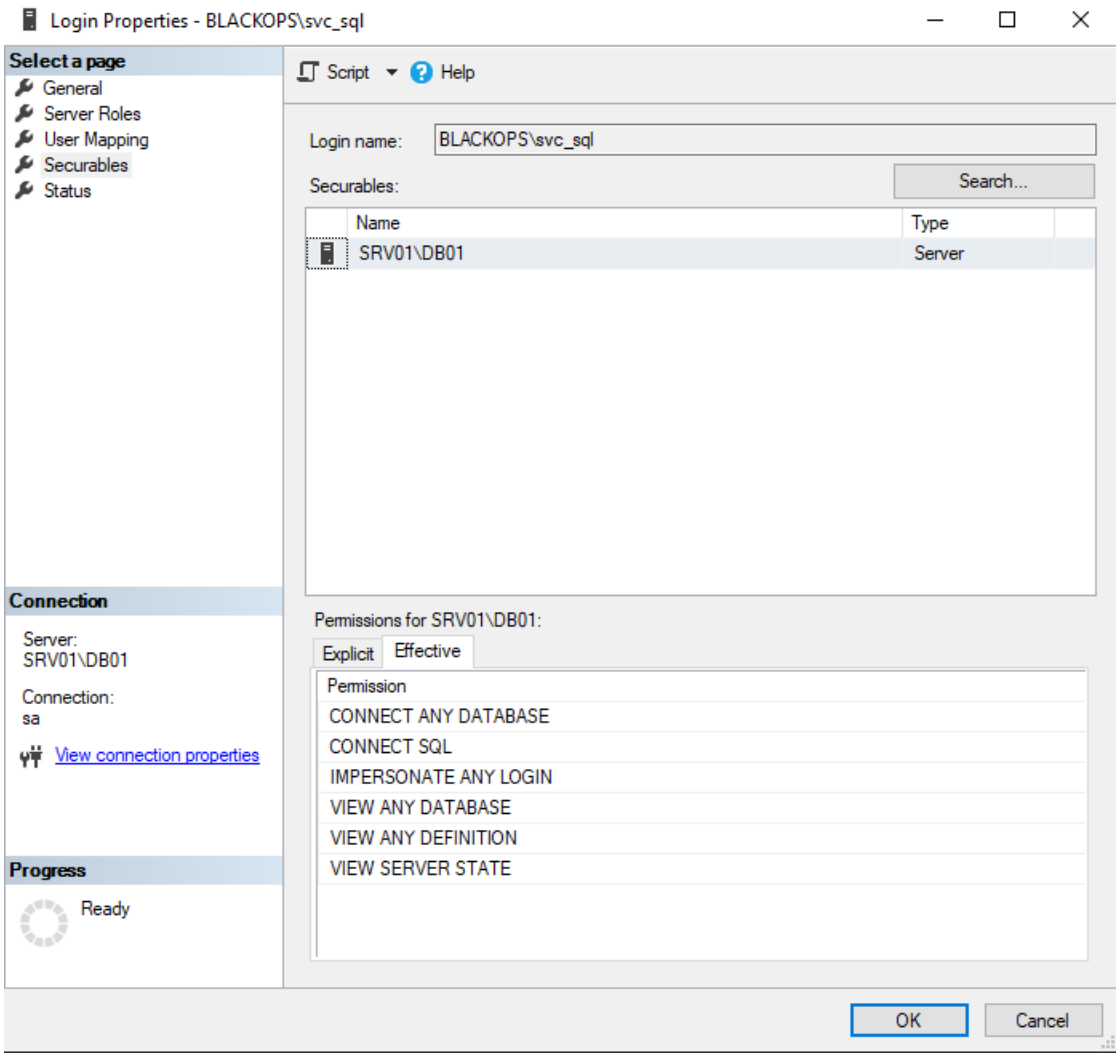


BLACKOPS\Domain Users: Least privilege

Leave everything default

BLACKOPS\svc_sql: Itself is not sysadmin but can impersonate sysadmin.

Select few permissions for svc_sql, **IMPERSONATE ANY LOGIN** is required.



By this way, we can abuse impersonate right to get sysadmin privilege. Let's check if we configured correctly. First, if we successfully integrate Kerberos authentication. Import powerupsql.ps1 script, and enumerate domain instance.

```
PS C:\Users\svc_sql> get-sqlinstancedomain


ComputerName     : srv02.blackops.local
Instance         : srv02.blackops.local,1433
DomainAccountSid : 1500000521000208180242133106192136197201322191838640
0
DomainAccount    : svc_sql
DomainAccountCn  : SVC SQL
Service          : MSSQLSvc
Spn              : MSSQLSvc/srv02.blackops.local:1433
LastLogon        : 6/16/2022 10:25 PM
Description      : Service account for MSSQL.

ComputerName     : srv01.blackops.local
Instance         : srv01.blackops.local,1433
DomainAccountSid : 1500000521000208180242133106192136197201322191838640
0
DomainAccount    : svc_sql
DomainAccountCn  : SVC SQL
Service          : MSSQLSvc
Spn              : MSSQLSvc/srv01.blackops.local:1433
LastLogon        : 6/16/2022 10:25 PM
Description      : Service account for MSSQL.

ComputerName     : srv02.blackops.local
Instance         : srv02.blackops.local\DB02
DomainAccountSid : 1500000521000208180242133106192136197201322191838640
0
DomainAccount    : svc_sql
DomainAccountCn  : SVC SQL
Service          : MSSQLSvc
Spn              : MSSQLSvc/srv02.blackops.local:DB02
LastLogon        : 6/16/2022 10:25 PM
Description      : Service account for MSSQL.

ComputerName     : srv01.blackops.local
Instance         : srv01.blackops.local\DB01
DomainAccountSid : 1500000521000208180242133106192136197201322191838640
0
DomainAccount    : svc_sql
DomainAccountCn  : SVC SQL
Service          : MSSQLSvc
```

It looks great! Then, access any instance to check if we get a TGS for SQL service. Here, I tested srv01.

```
PS C:\Users\svc_sql> get-sqlserverinfo -instance 'srv01.blackops.local,1433'


ComputerName           : srv01.blackops.local
Instance               : SRV01\DB01
DomainName             : BLACKOPS
ServiceProcessID       : 3752
ServiceName            : MSSQL$DB01
ServiceAccount         : BLACKOPS\svc_sql
AuthenticationMode     : Windows and SQL Server Authentication
ForcedEncryption       : 0
Clustered              : No
SQLServerVersionNumber : 15.0.2000.5
SQLServerMajorVersion  : 2019
SQLServerEdition       : Developer Edition (64-bit)
SQLServerServicePack   : RTM
OSArchitecture         : X64
OsVersionNumber        : SQL
Currentlogin           : BLACKOPS\svc_sql
IsSysadmin             : No
ActiveSessions         : 1
```

Check cached tickets, and I find the TGS, it means Kerberos is integrated successfully.

```
PS C:\Users\svc_sql> klist

Current LogonId is 0:0x3d8298

Cached Tickets: (4)

#0>     Client: svc_sql @ BLACKOPS.LOCAL
        Server: krbtgt/BLACKOPS.LOCAL @ BLACKOPS.LOCAL
        KerbTicket Encryption Type: AES-256-CTS-HMAC-SHA1-96
        Ticket Flags 0x40e10000 -> forwardable renewable initial pre_authent name_canonicalize
        Start Time: 6/16/2022 21:27:18 (local)
        End Time:   6/17/2022 7:27:18 (local)
        Renew Time: 6/23/2022 21:27:18 (local)
        Session Key Type: AES-256-CTS-HMAC-SHA1-96
        Cache Flags: 0x1 -> PRIMARY
        Kdc Called: DC

#1>     Client: svc_sql @ BLACKOPS.LOCAL
        Server: MSSQLSvc/srv01.blackops.local:1433 @ BLACKOPS.LOCAL
        KerbTicket Encryption Type: RSADSI RC4-HMAC(NT)
        Ticket Flags 0x40a10000 -> forwardable renewable pre_authent name_canonicalize
        Start Time: 6/16/2022 22:31:07 (local)
        End Time:   6/17/2022 7:27:18 (local)
        Renew Time: 6/23/2022 21:27:18 (local)
        Session Key Type: RSADSI RC4-HMAC(NT)
        Cache Flags: 0
        Kdc Called: dc.blackops.local

#2>     Client: svc_sql @ BLACKOPS.LOCAL
        Server: ldap/dc.blackops.local @ BLACKOPS.LOCAL
        KerbTicket Encryption Type: AES-256-CTS-HMAC-SHA1-96
        Ticket Flags 0x40a50000 -> forwardable renewable pre_authent ok_as_delegate name_canonicalize
        Start Time: 6/16/2022 22:30:34 (local)
        End Time:   6/17/2022 7:27:18 (local)
        Renew Time: 6/23/2022 21:27:18 (local)
        Session Key Type: AES-256-CTS-HMAC-SHA1-96
```

Then, we need to verify permission assignment. I choose three types of users to check

helen.park: Least privilege

```
┌──(root💀os)-[/home/os/Desktop]
└─# python3 impacket/examples/mssqlclient.py -windows-auth blackops/helen.park@192.168.0.54
Impacket v0.9.24 - Copyright 2021 SecureAuth Corporation

Password:
[*] Encryption required, switching to TLS
[*] ENVCHANGE(DATABASE): Old Value: master, New Value: master
[*] ENVCHANGE(LANGUAGE): Old Value: , New Value: us_english
[*] ENVCHANGE(PACKETSIZE): Old Value: 4096, New Value: 16192
[*] INFO(SRV01\DB01): Line 1: Changed database context to 'master'.
[*] INFO(SRV01\DB01): Line 1: Changed language setting to us_english.
[*] ACK: Result: 1 - Microsoft SQL Server (150 7208)
[!] Press help for extra shell commands
SQL> select is_srvrolemember('sysadmin');


-----------

          0

SQL> execute as login='sa'
[-] ERROR(SRV01\DB01): Line 1: Cannot execute as the server principal because the principal "sa" does no
t exist, this type of principal cannot be impersonated, or you do not have permission.
SQL>
```

We can see, helen.park can only access SQL instance and has very limited privilege. She cannot impersonate other logins.

Administrator: Sysadmin

```
┌──(root💀 os)-[/home/os/Desktop]
└─# python3 impacket/examples/mssqlclient.py -windows-auth blackops/administrator@192.168.0.54
Impacket v0.9.24 - Copyright 2021 SecureAuth Corporation

Password:
[*] Encryption required, switching to TLS
[*] ENVCHANGE(DATABASE): Old Value: master, New Value: master
[*] ENVCHANGE(LANGUAGE): Old Value: , New Value: us_english
[*] ENVCHANGE(PACKETSIZE): Old Value: 4096, New Value: 16192
[*] INFO(SRV01\DB01): Line 1: Changed database context to 'master'.
[*] INFO(SRV01\DB01): Line 1: Changed language setting to us_english.
[*] ACK: Result: 1 - Microsoft SQL Server (150 7208)
[!] Press help for extra shell commands
SQL> select is_srvrolemember('sysadmin');


-----------

          1
```

Domain admin has highest privilege.

svc_sql: Can impersonate sa to get sysadmin privilege.

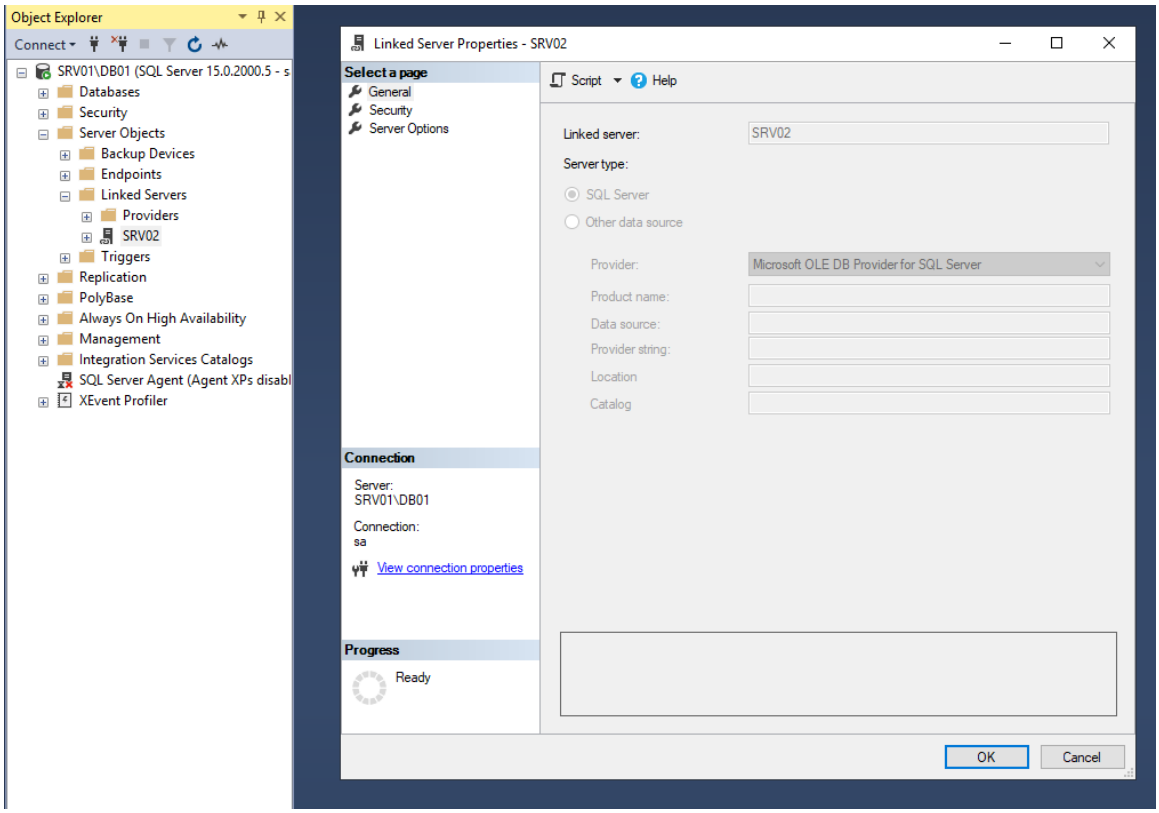
```
┌──(root💀 os)-[/home/os/Desktop]
└─# python3 impacket/examples/mssqlclient.py -windows-auth blackops/svc_sql@192.168.0.54
Impacket v0.9.24 - Copyright 2021 SecureAuth Corporation

Password:
[*] Encryption required, switching to TLS
[*] ENVCHANGE(DATABASE): Old Value: master, New Value: master
[*] ENVCHANGE(LANGUAGE): Old Value: , New Value: us_english
[*] ENVCHANGE(PACKETSIZE): Old Value: 4096, New Value: 16192
[*] INFO(SRV01\DB01): Line 1: Changed database context to 'master'.
[*] INFO(SRV01\DB01): Line 1: Changed language setting to us_english.
[*] ACK: Result: 1 - Microsoft SQL Server (150 7208)
[!] Press help for extra shell commands
SQL> select is_srvrolemember('sysadmin');


-----------

          0

SQL> execute as login='sa';select is_srvrolemember('sysadmin');


-----------

          1

SQL>
```

After impersonation, svc_sql does not have sysadmin privilege. But after impersonate, it has sysadmin privilege.

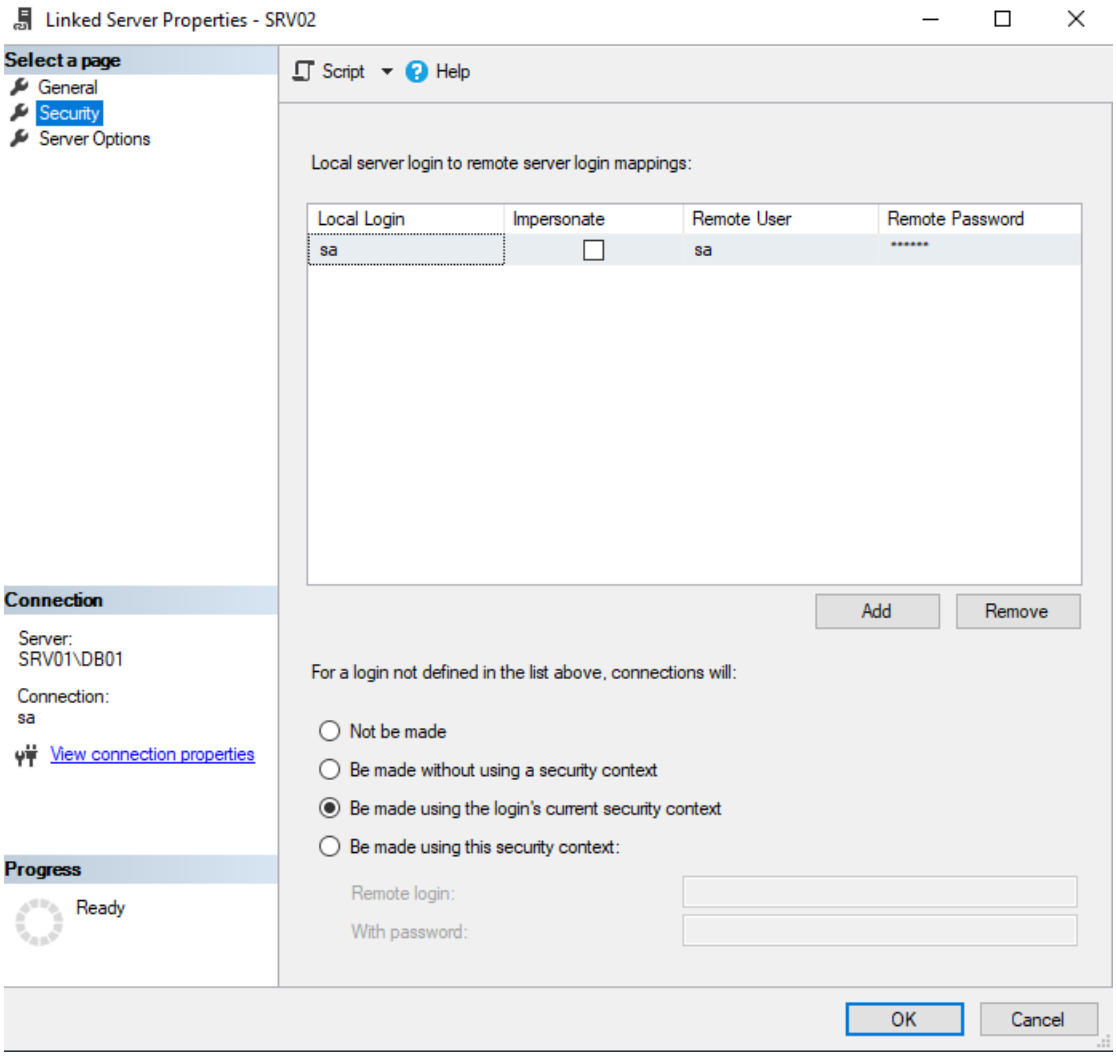
So the permission assignment is successful as well.

Up to now, we can repeat previous steps related in SQL part on SRV02, but just remember to change server/instance value. But then we will configure SQL link on SRV01, I did not configure SQL link on SRV02, but of course you can add one.

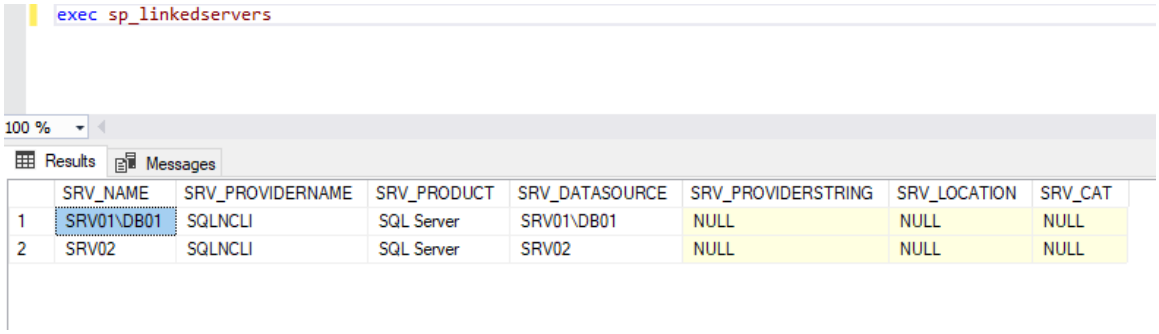
Right click **Server Objects -> Linked Servers,** add a new link. The **General** tab should be like this:



On **Security** tab, we add a new entry to login mappings, map local login sa to remote login sa. If it is confusing, you can change map to SRV02\Administrator. What does it mean? If our current login is sa, we know we have sysadmin privilege on SRV01. But if we follow the link to reach SRV02, we may not have sysadmin privilege. Since we are designing a misconfiguration, so I just map it to an sysadmin login on SRV02. By this way, we still have sysadmin login when reaching SRV02. And select “Be made using the login’s current security context” option, it is easy to understand.



Okay, so we configured SQL link: SRV01 -> SRV02, let's check it.



```
exec sp_linkedservers
```

100 %

Results | Messages

	SRV_NAME	SRV_PROVIDERNAME	SRV_PRODUCT	SRV_DATASOURCE	SRV_PROVIDERSTRING	SRV_LOCATION	SRV_CAT
1	SRV01\DB01	SQLNCLI	SQL Server	SRV01\DB01	NULL	NULL	NULL
2	SRV02	SQLNCLI	SQL Server	SRV02	NULL	NULL	NULL

The link is correct, then let's check if we can still have sysadmin privilege on SRV02 via SQL link.

Before impersonation

```
┌──(root💀os)-[/home/os/Desktop]
└─# python3 impacket/examples/mssqlclient.py -windows-auth blackops/svc_sql@192.168.0.54
Impacket v0.9.24 - Copyright 2021 SecureAuth Corporation

Password:
[*] Encryption required, switching to TLS
[*] ENVCHANGE(DATABASE): Old Value: master, New Value: master
[*] ENVCHANGE(LANGUAGE): Old Value: , New Value: us_english
[*] ENVCHANGE(PACKETSIZE): Old Value: 4096, New Value: 16192
[*] INFO(SRV01\DB01): Line 1: Changed database context to 'master'.
[*] INFO(SRV01\DB01): Line 1: Changed language setting to us_english.
[*] ACK: Result: 1 - Microsoft SQL Server (150 7208)
[!] Press help for extra shell commands
SQL> select * from openquery("SRV02",'select is_srvrolemember(''sysadmin'')');
[-] ERROR(SRV02\DB02): Line 1: Login failed for user 'NT AUTHORITY\ANONYMOUS LOGON'.
SQL>
```

We can see if we do not impersonate sa and follow the link, we will get an error, because we did not map svc_sql to SRV02 previously. However, if we impersonate sa, then the result it totally different.

```
┌──(root💀os)-[/home/os/Desktop]
└─# python3 impacket/examples/mssqlclient.py -windows-auth blackops/svc_sql@192.168.0.54
Impacket v0.9.24 - Copyright 2021 SecureAuth Corporation

Password:
[*] Encryption required, switching to TLS
[*] ENVCHANGE(DATABASE): Old Value: master, New Value: master
[*] ENVCHANGE(LANGUAGE): Old Value: , New Value: us_english
[*] ENVCHANGE(PACKETSIZE): Old Value: 4096, New Value: 16192
[*] INFO(SRV01\DB01): Line 1: Changed database context to 'master'.
[*] INFO(SRV01\DB01): Line 1: Changed language setting to us_english.
[*] ACK: Result: 1 - Microsoft SQL Server (150 7208)
[!] Press help for extra shell commands
SQL> execute as login='sa';
SQL> select * from openquery("SRV02",'select is_srvrolemember(''sysadmin'')');


-----------

          1
```

We can access SRV02 with sysadmin privilege! So the permission is configured well.

After a long journey, we successfully configured SRV01.

Server 2

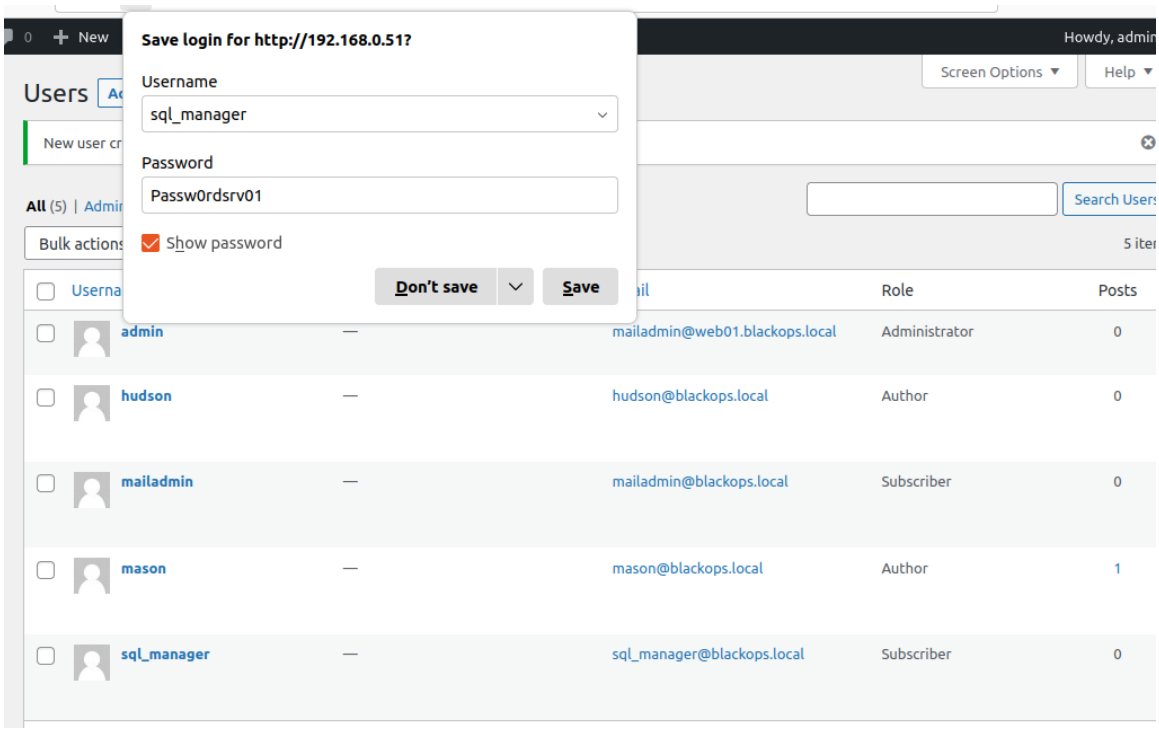
srv02.blackops.local

Autologin: None

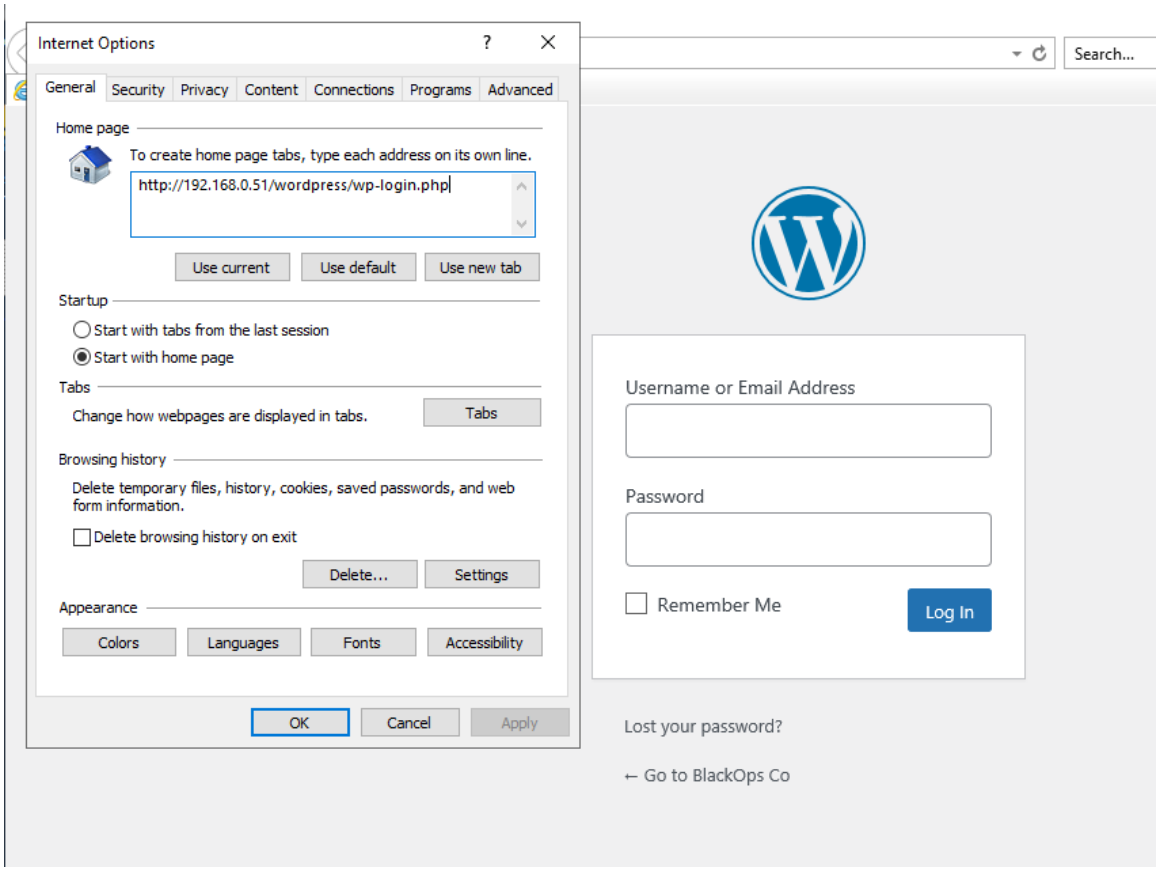
Remote Desktop Login: Enable RDP, and add svc_sql to Remote Desktop Users local group.

SQL Instance: Almost the same as we did on SRV01, but with a different IP/Instance. And no need to add a link, but if you want, that's totally cool as well.

We previously set unconstrained delegation for SRV02, so at this moment we just need to add a privilege escalation vector. Back to web01, add a new user **sql_manager:Passw0rdsrv01** on WordPress App.



It is very simple, open IE browser, set login page of WordPress CMS as home page, and check remember me. From the perspective of an attacker, he needs to check credentials stored in IE. The password itself does not work for SRV02, but fortunately it is a lazy admin, the attacker just need to change Passw0rdsrv01 to Passw0rdsrv02, then we get SRV02 local admin's password.



We finally successfully built the whole vulnerable AD set!

In the End

Thanks for spending time on reading such a long article, I really appreciate! This is the first time for me to design a vulnerable AD set, so there is a lot of room for improvement. And though the guide is very detailed, I cannot make sure I did not miss anything. If you follow my steps and still have difficulty making it work, just let me know!

In the future, if I plan to design more vulnerable AD sets, I would like to cover and add more features and vectors such as ADCS abuse, Relay Attack, Phishing and User Simulation, etc.

Since there is copyright concern, I will make sure if it is legal to share my VM/images. If it is okay, I will share my own VM soon. But building by your own is a good way to learn! I will release the walkthrough of the vulnerable AD soon. I invited my friend (Passed OSCP, CRTO) to test my vulnerable AD set, he reached the 3rd machine after about 24 hours with some hints. And after about 72 hours, he reached DC. Welcome to play with my AD set : D