



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.

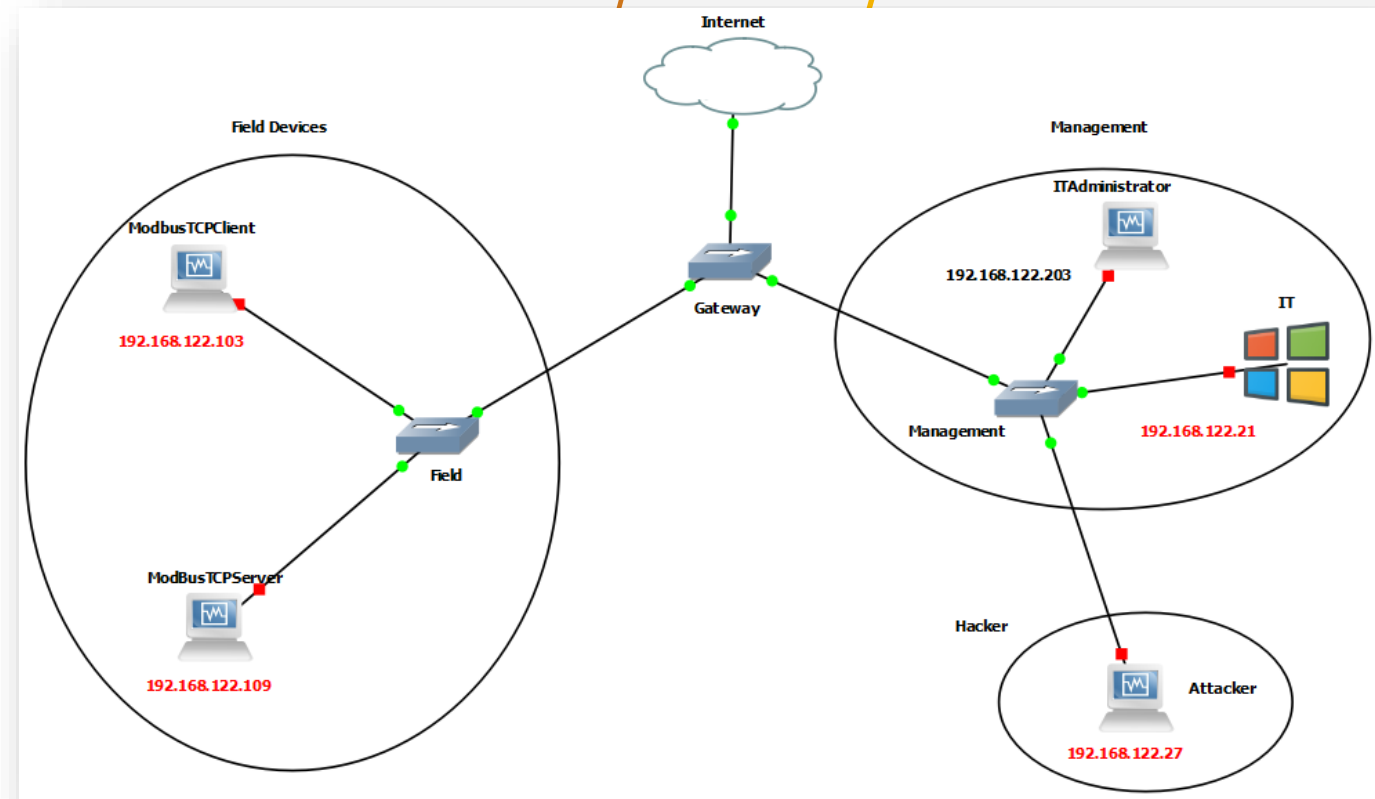
GNS3 Simulator

GNS3 Simulator

- 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?

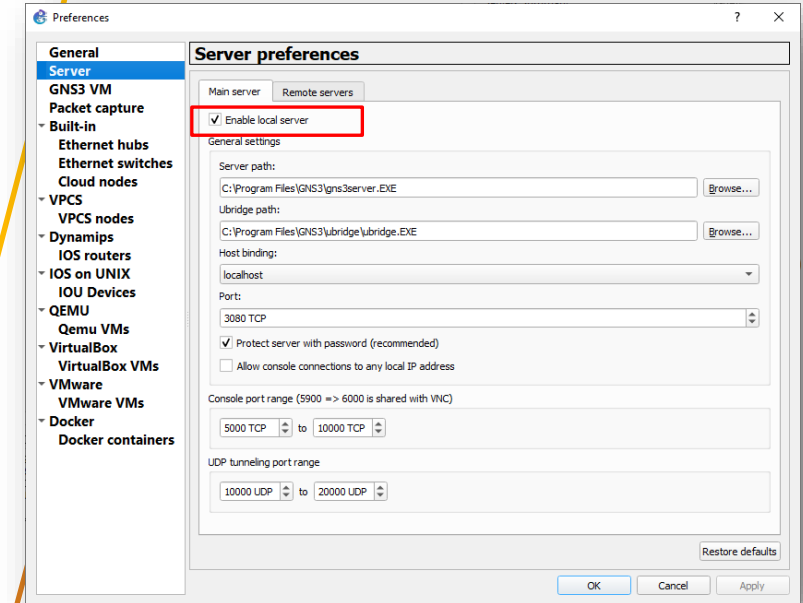
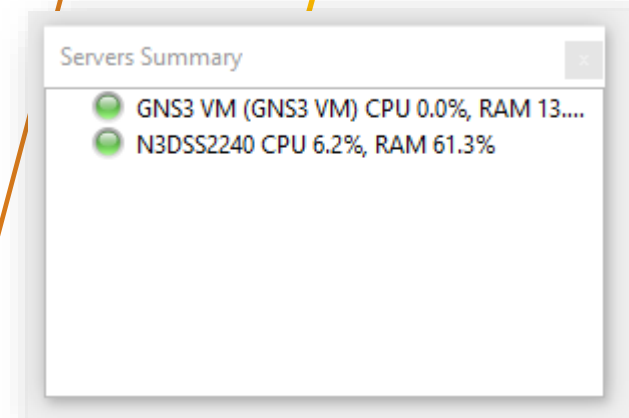
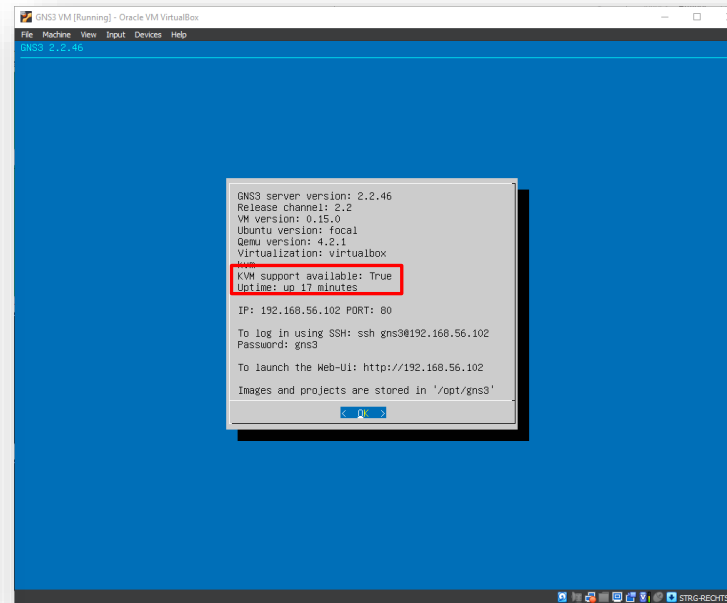
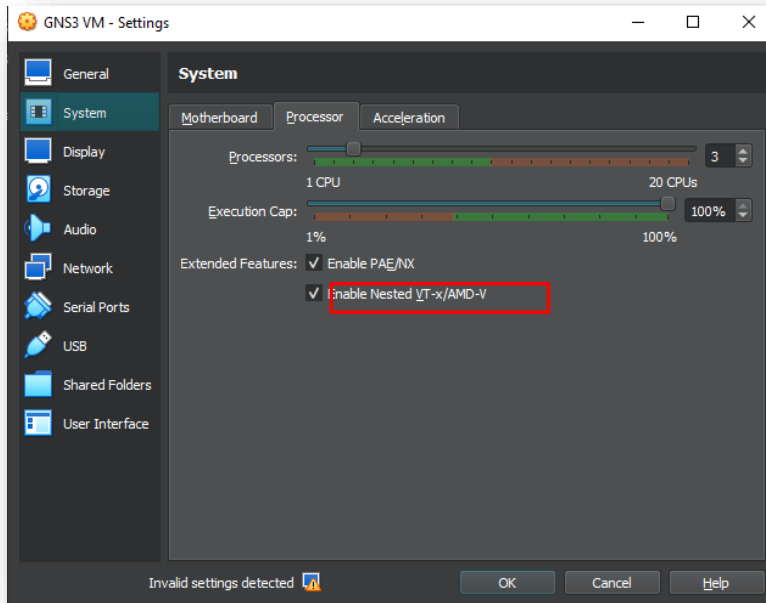
- GNS3 facilitates the building of a complete lab environment by integrating multiple VMs (installed individually).
- This will help create a fully isolated environment consisting of multiple VMs with victim machines and an attacker, allowing you to perform your practical tasks within this course in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

The image shows the GNS3 application interface and its preferences dialog. The main window displays a 'Please create a project' message and a console window. The preferences dialog is open, showing the 'VirtualBox preferences' section. The 'VirtualBox' option in the left sidebar is highlighted with a red box. The 'Path to VBoxManage:' field is set to 'C:\Program Files\Oracle\VirtualBox\VBoxManage.exe'.

GNS3 Main Window:

- File Edit View Control Node Annotate Tools Help
- Select all Ctrl+A
- Select gone Ctrl-Shift+A
- Manage snapshots
- Preferences... Ctrl-Shift+P
- Servers Summary
 - GNS3 VM (GNS3 VM) CPU 0.3%, RAM 13...
 - N3D5S2240 CPU 8.3%, RAM 64.5%
- Topology Summary
 - Node Console
- Please create a project
- Console
 - GNS3 management console.
 - Running GNS3 version 2.2.46 on Windows (64-bit) with Python 3.10.11 Qt 5.15.2 and PyQt 5.15.10.
 - Copyright (c) 2006-2024 GNS3 Technologies.
 - Use Help -> GNS3 Doctor to detect common issues.
 - =>

Preferences - VirtualBox preferences:

- General
- Server
- GNS3 VM
- Packet capture
- Built-in
 - Ethernet hubs
 - Ethernet switches
 - Cloud nodes
- VPCS
 - VPCS nodes
- Dynamips
 - IOS routers
- IOS on UNIX
 - IOU Devices
- QEMU
 - Qemu VMs
- VirtualBox**
 - VirtualBox VMs**
- VMware
 - VMware VMs
- Docker
 - Docker containers

Local settings

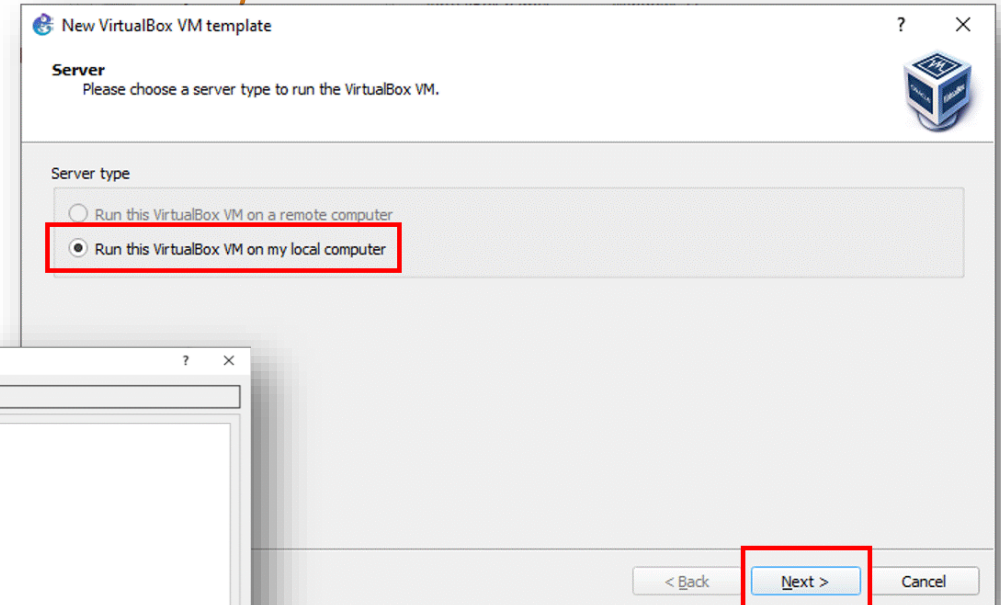
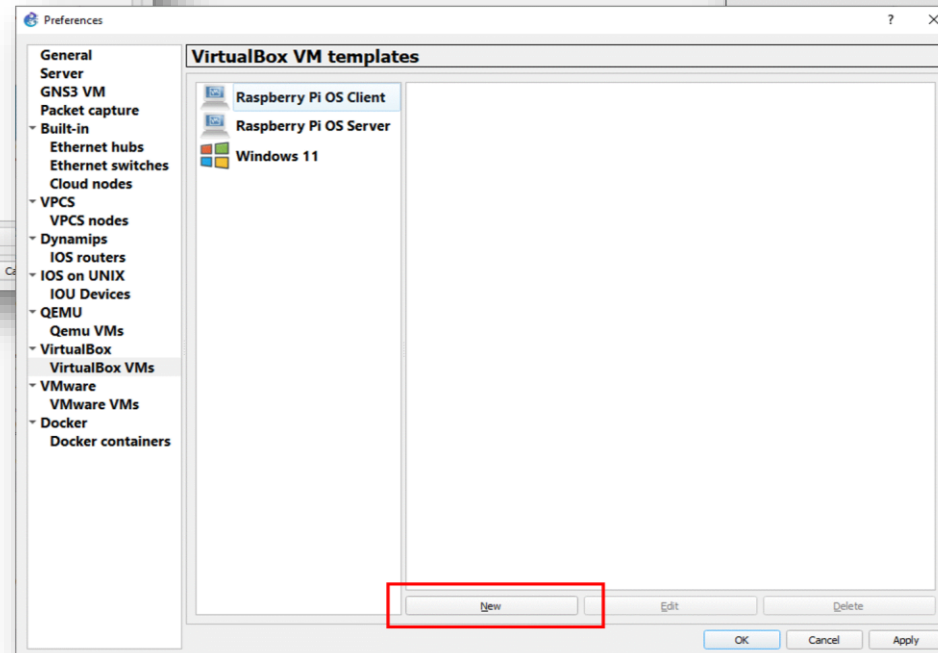
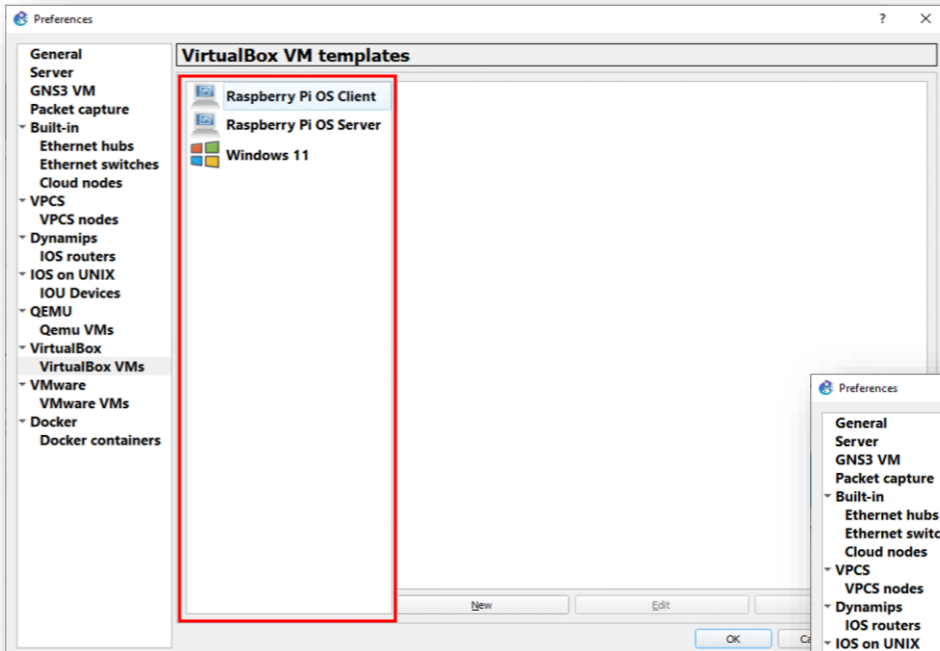
Path to VBoxManage: C:\Program Files\Oracle\VirtualBox\VBoxManage.exe

Restore defaults

OK Cancel Apply

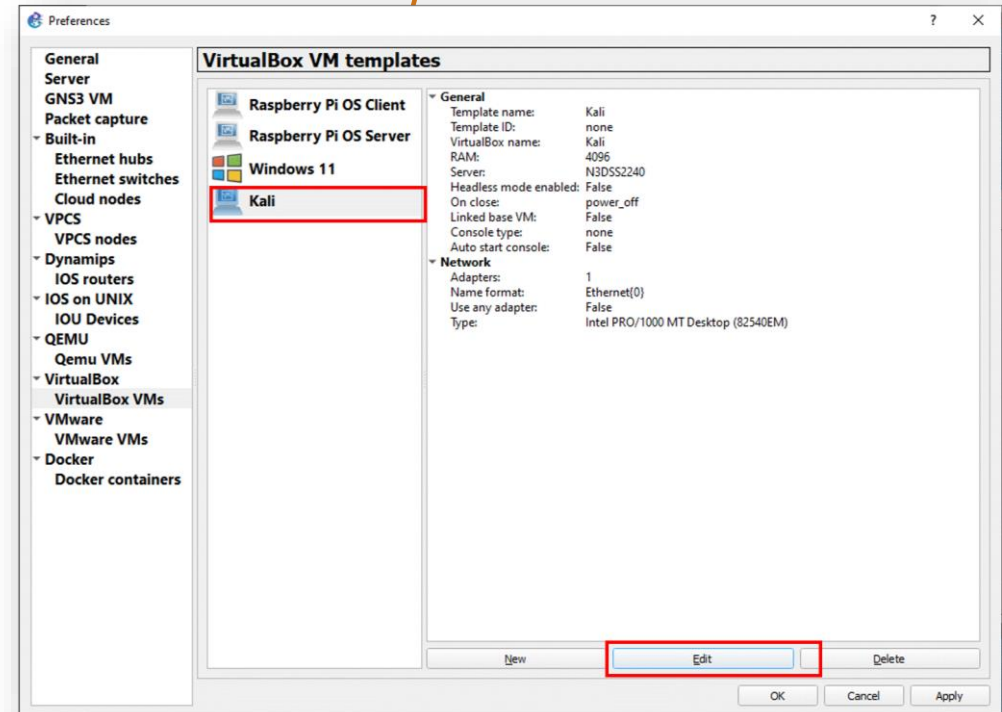
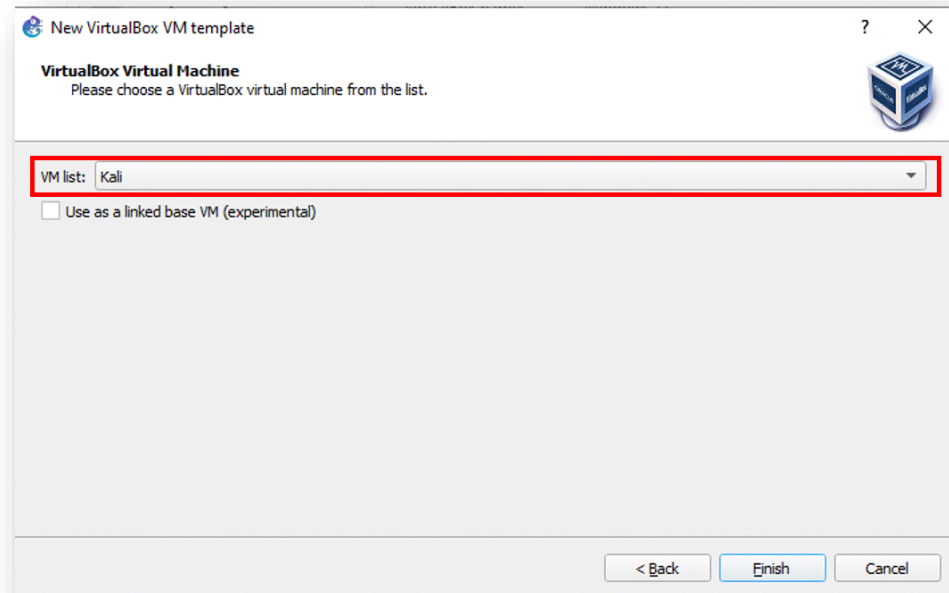
- Do the same for any VM you want to integrate with GNS3

Integrating VMs to GNS3



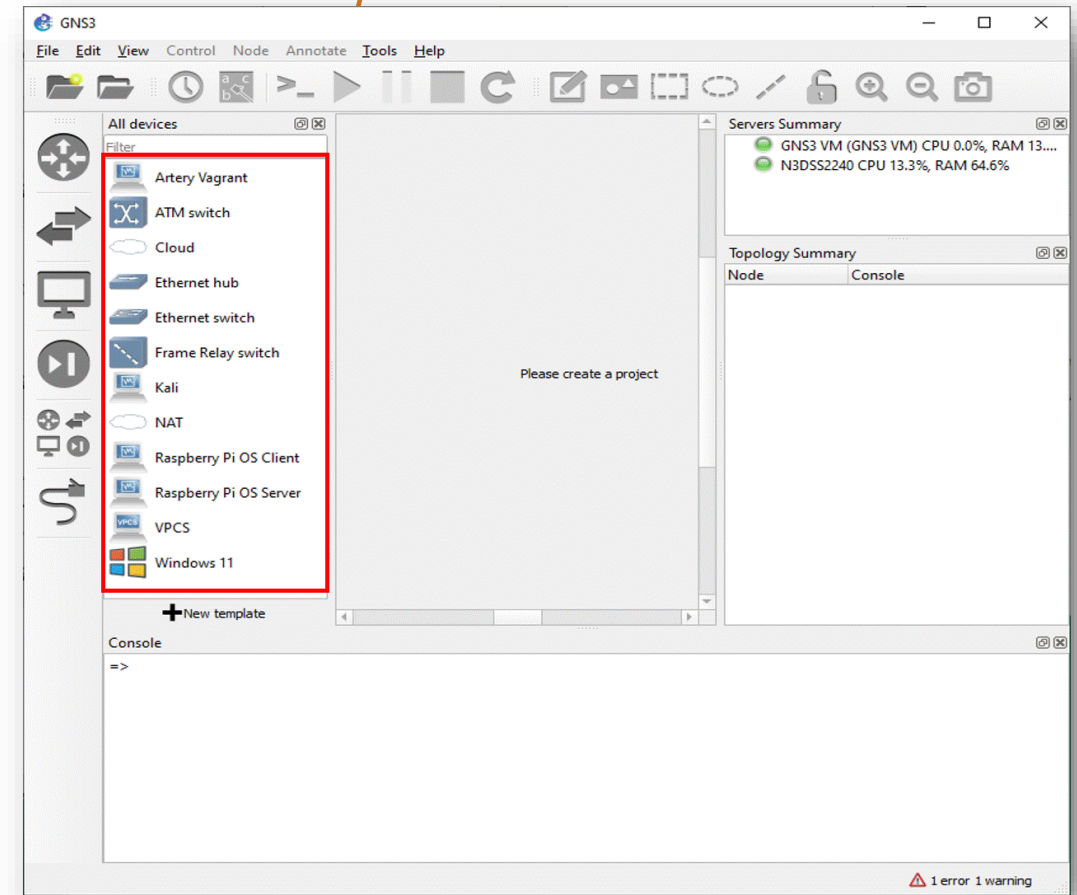
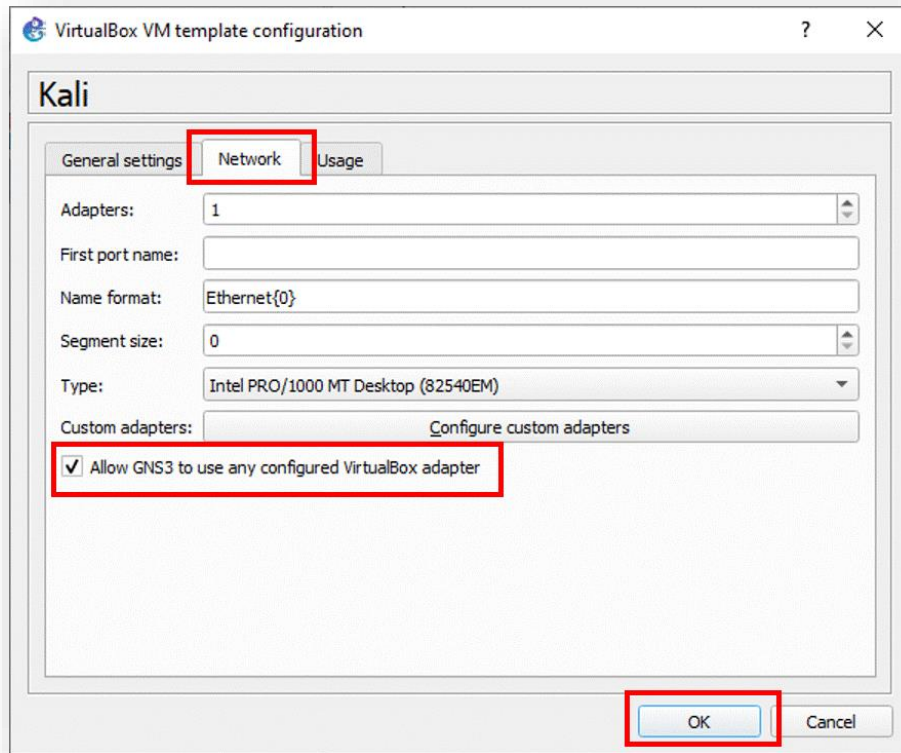
- Do the same for any VM you want to integrate with GNS3

Integrating VMs to GNS3



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

Integrating VMs to GNS3

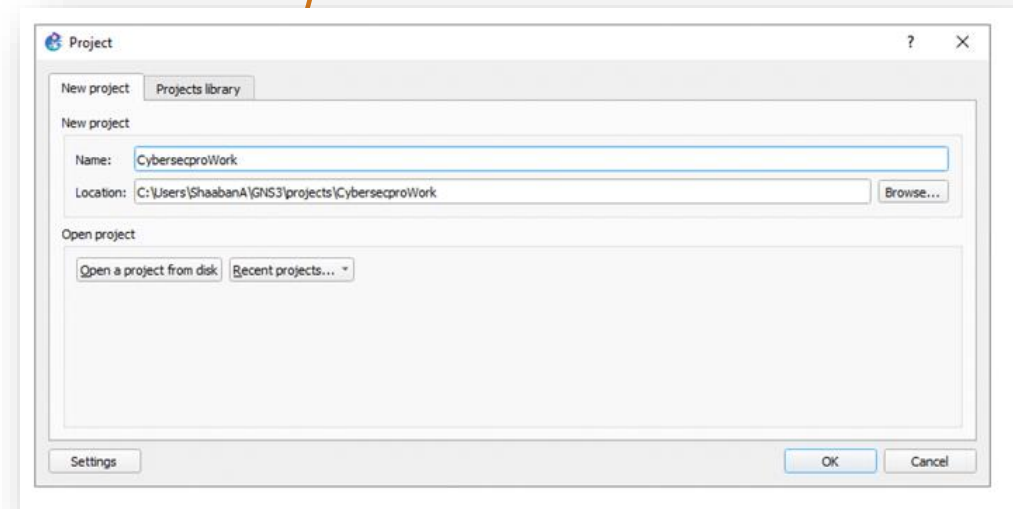
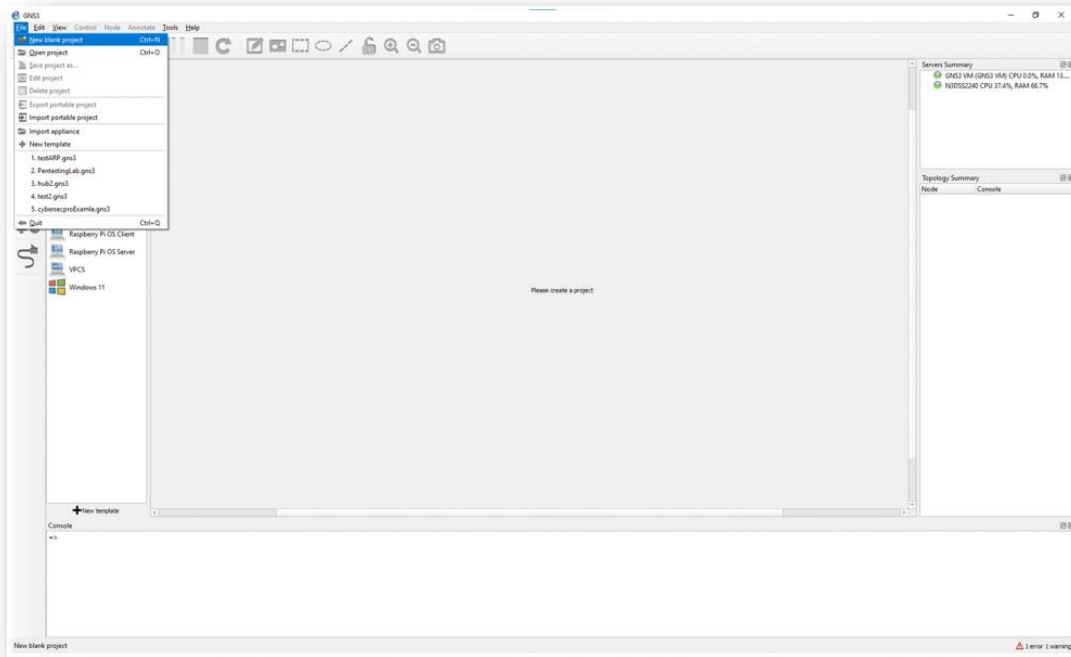
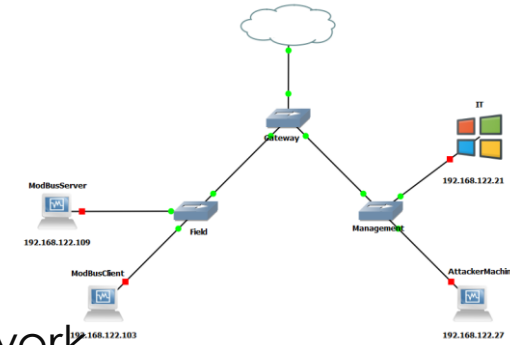


- Do the same for any VM you want to integrate with GNS3

Network Topology Modeling

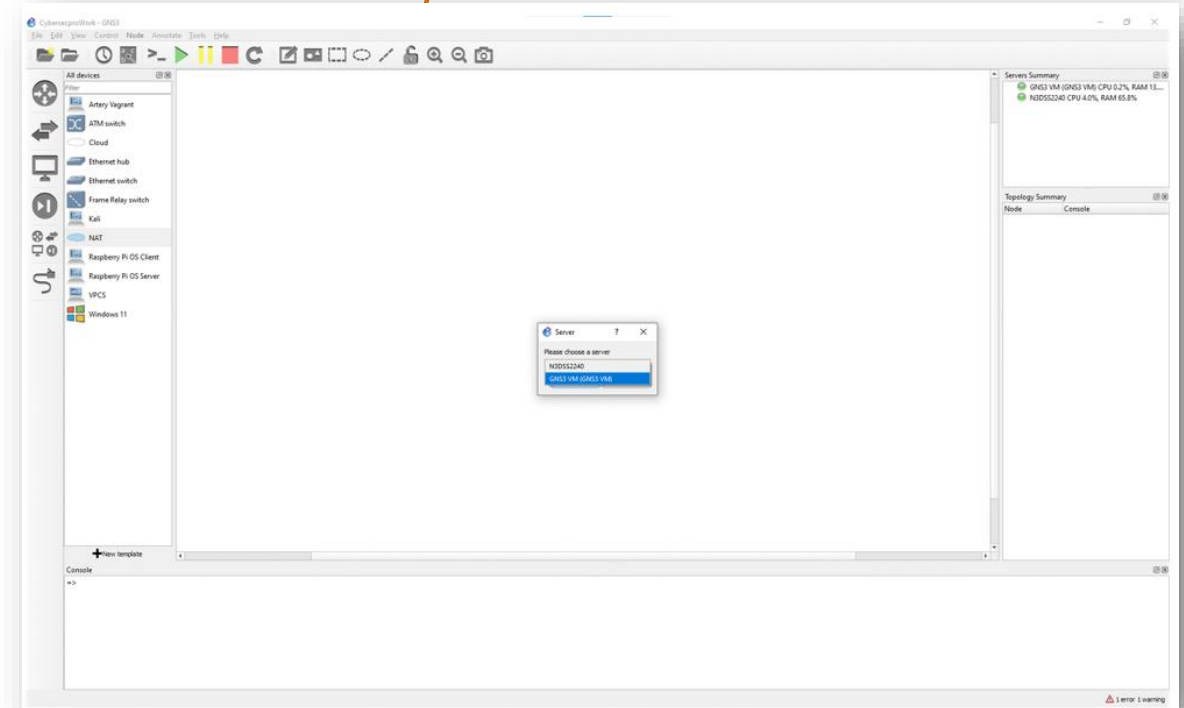
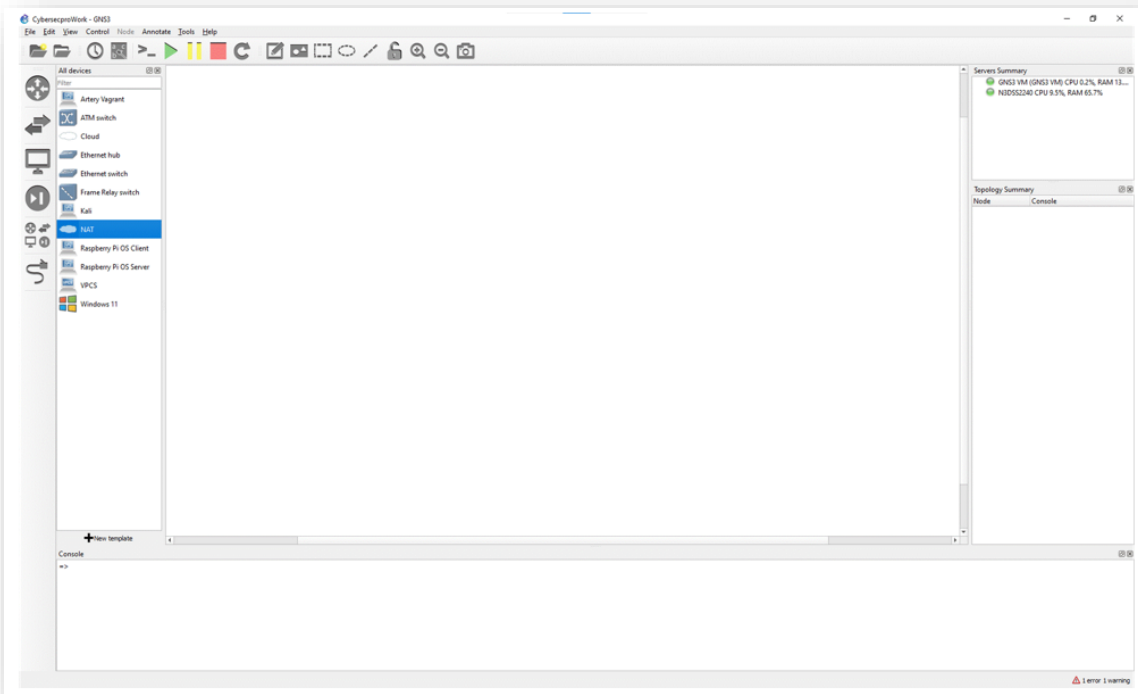
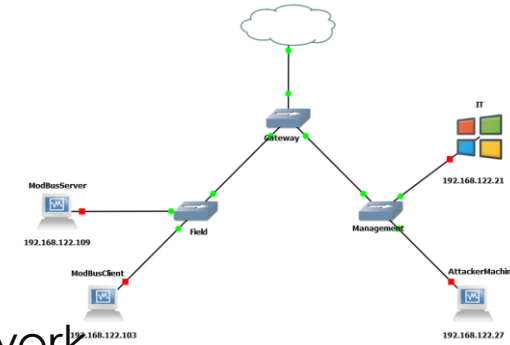
GNS3 Client for Network Modeling

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



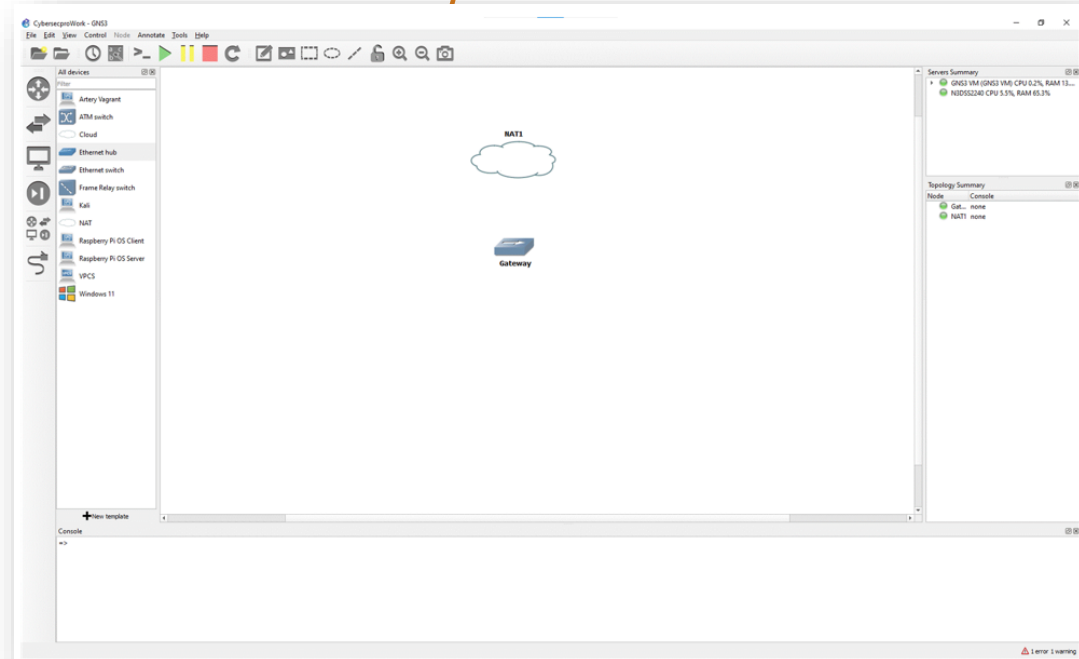
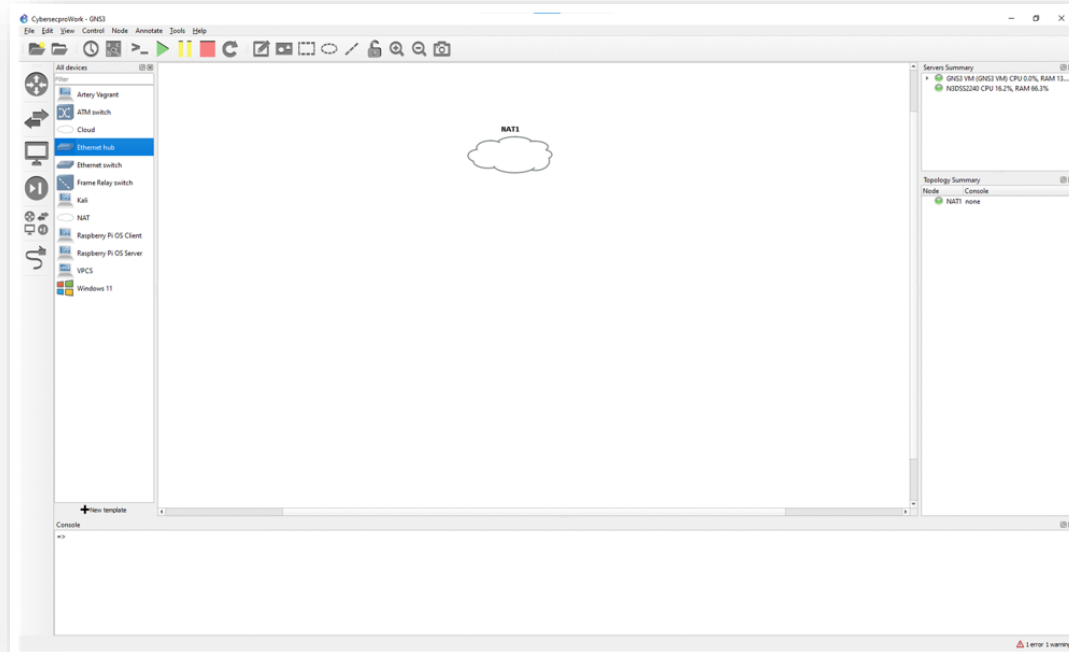
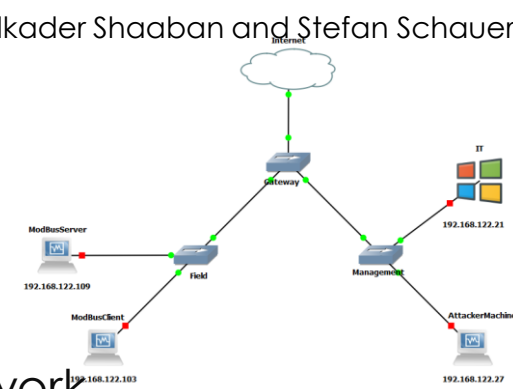
GNS3 Client for Network Modeling

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



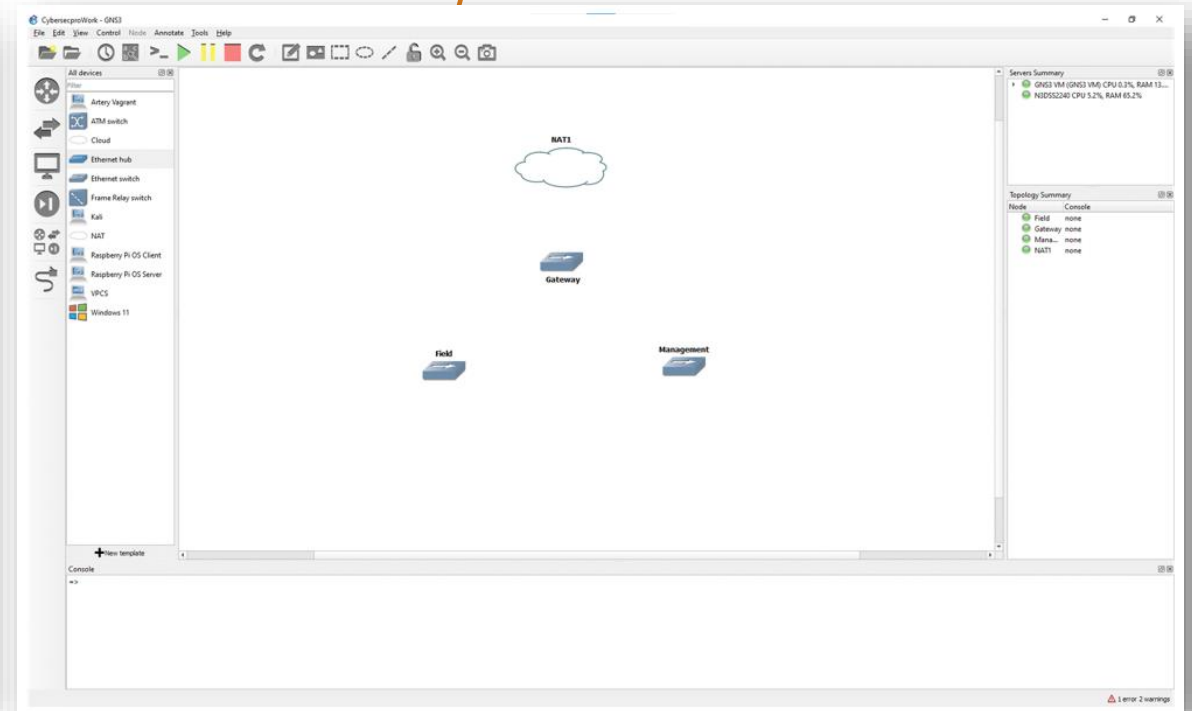
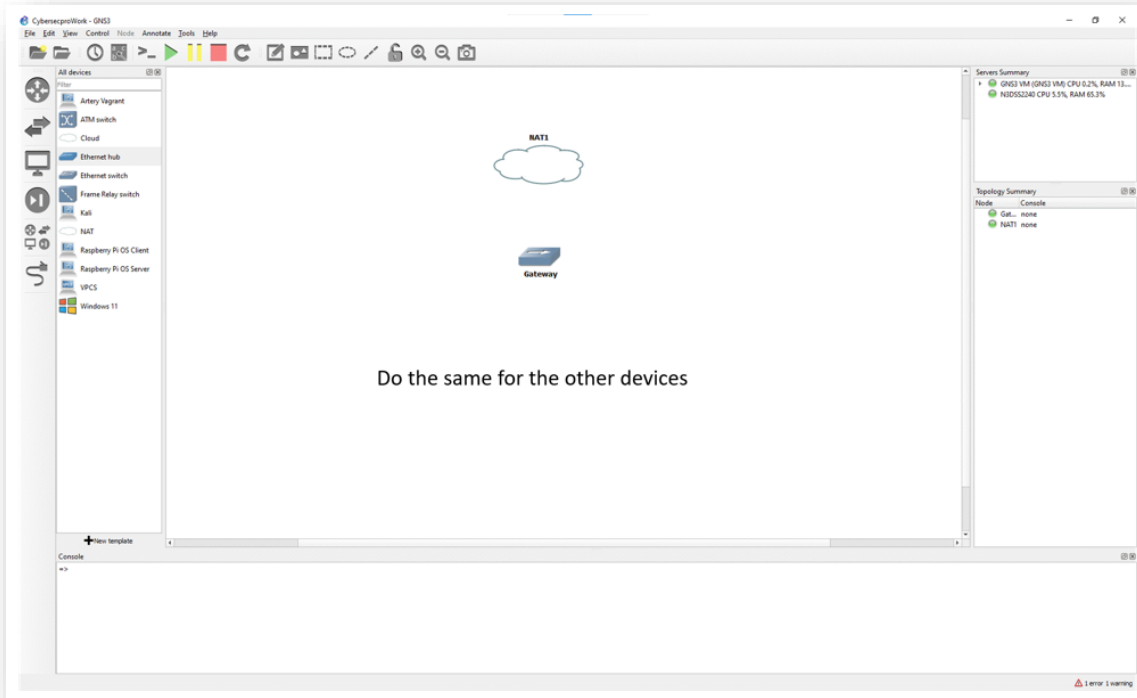
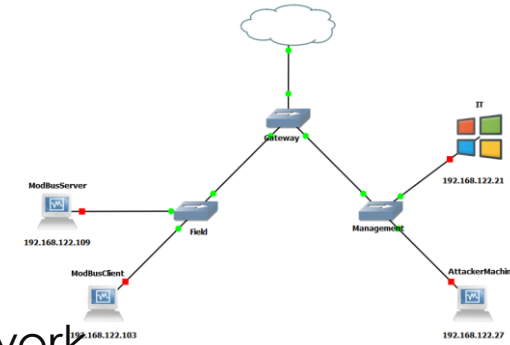
GNS3 Client for Network Modeling

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



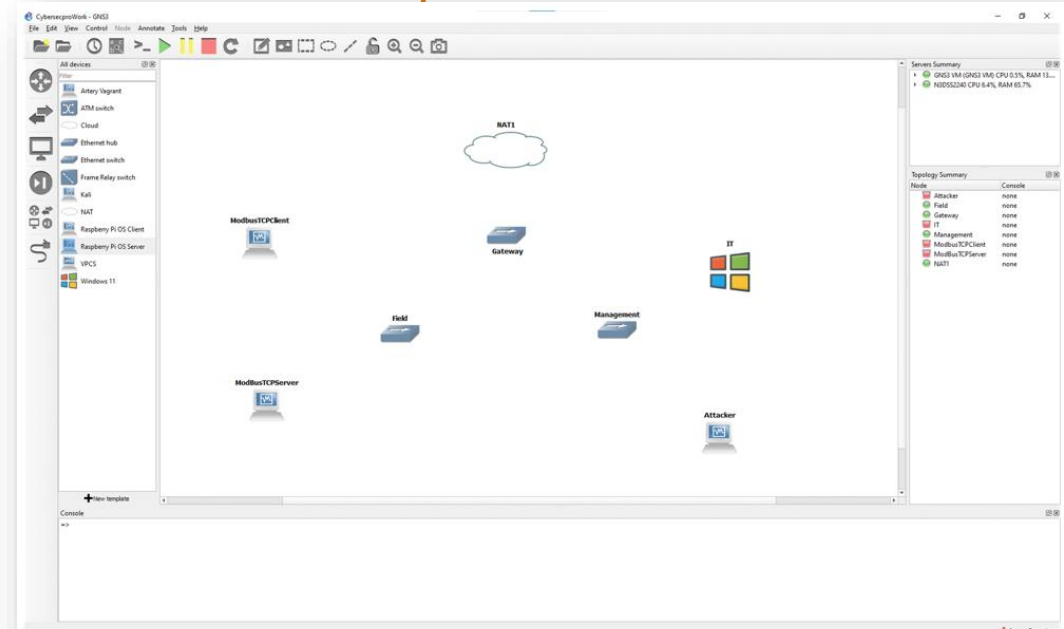
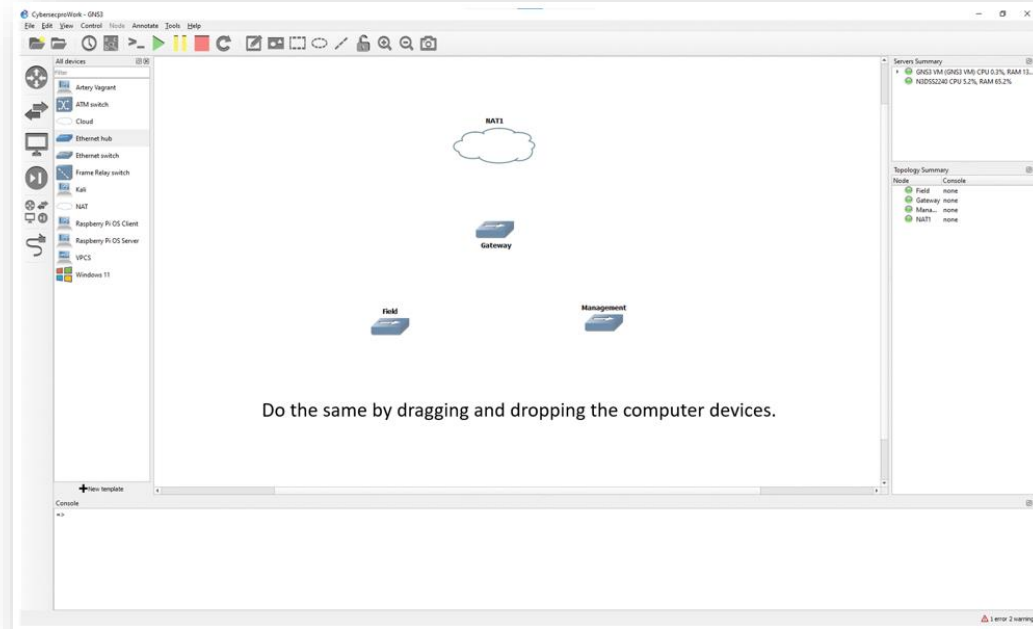
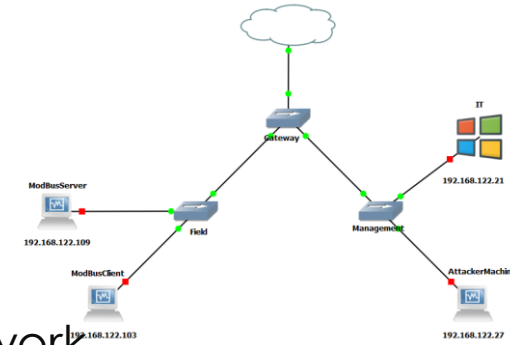
GNS3 Client for Network Modeling

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



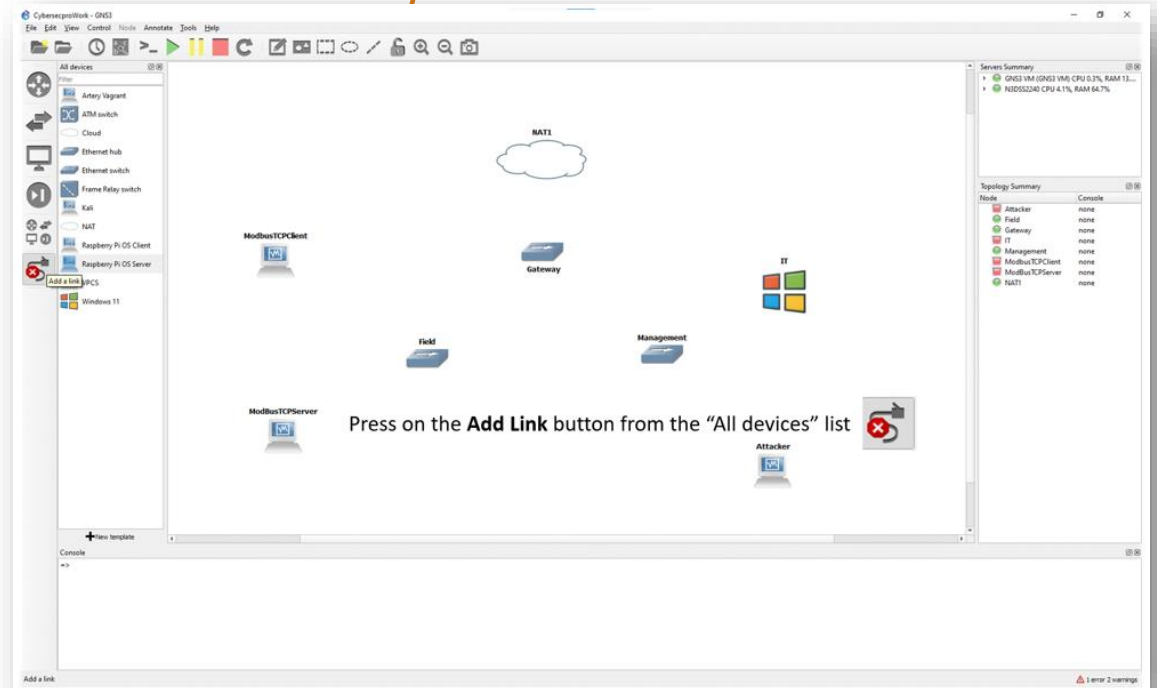
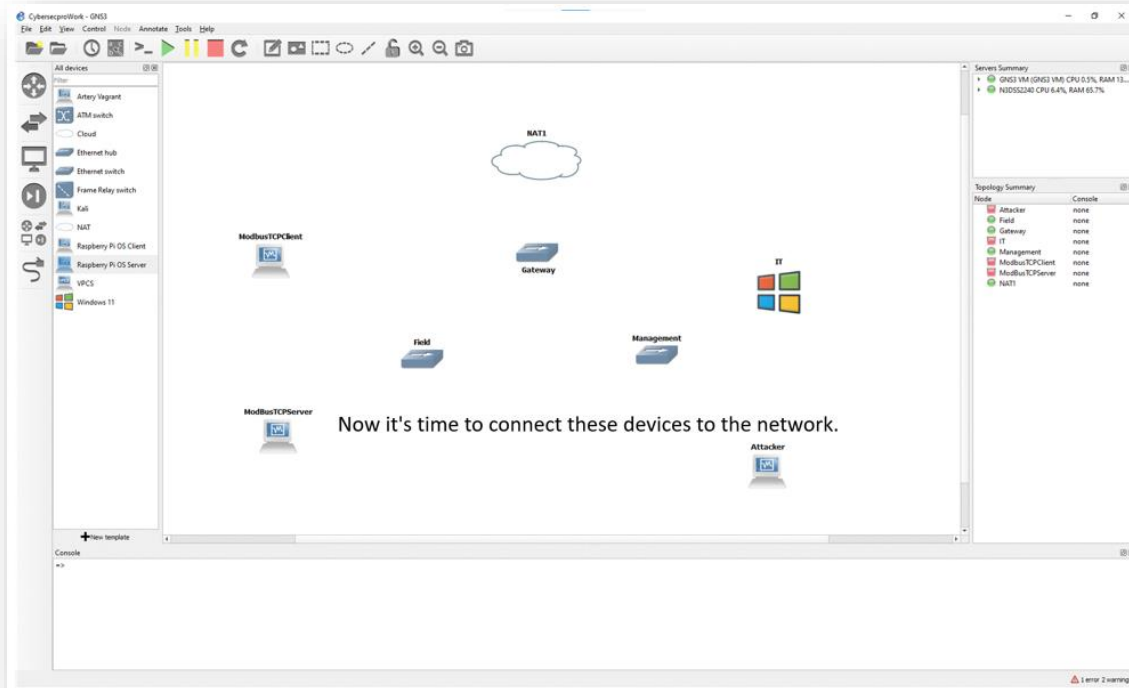
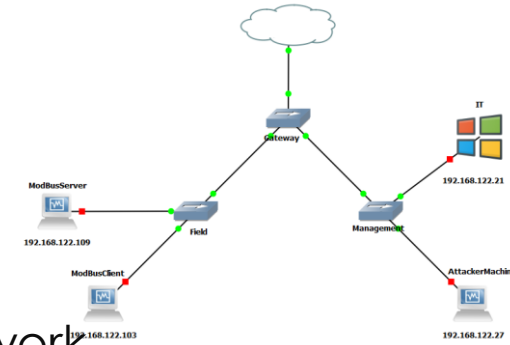
GNS3 Client for Network Modeling

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



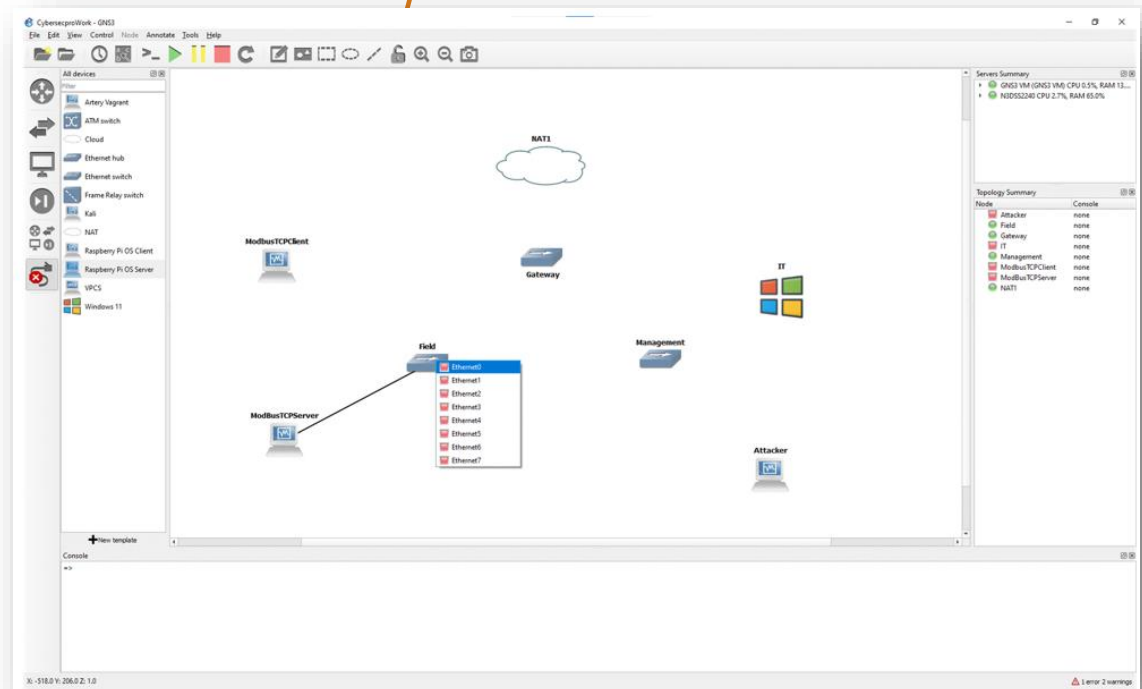
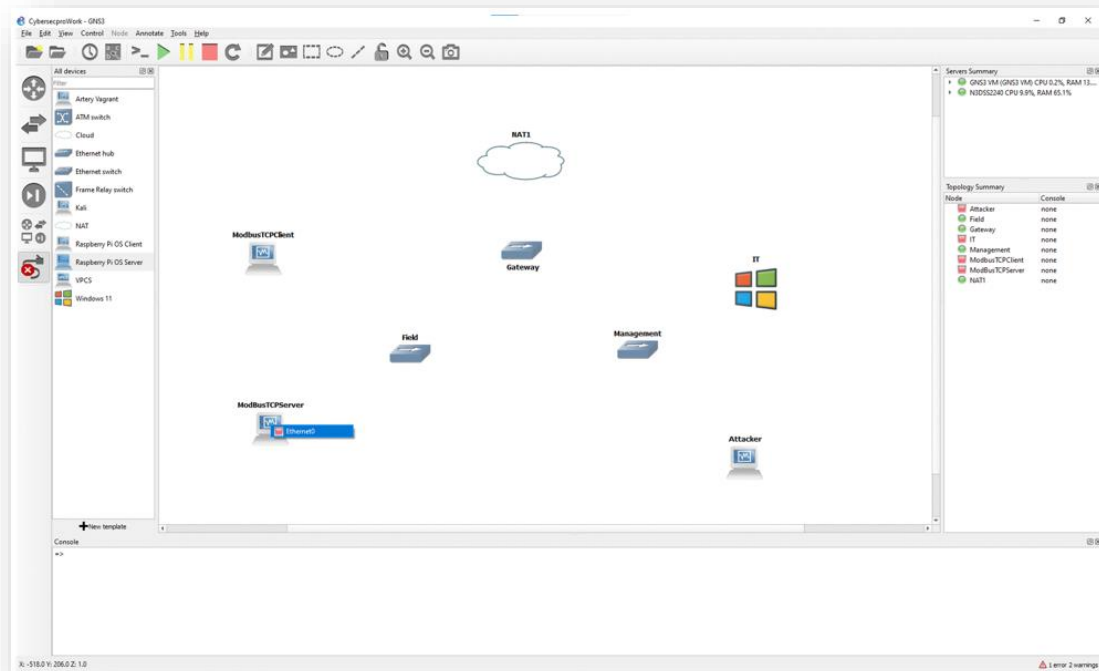
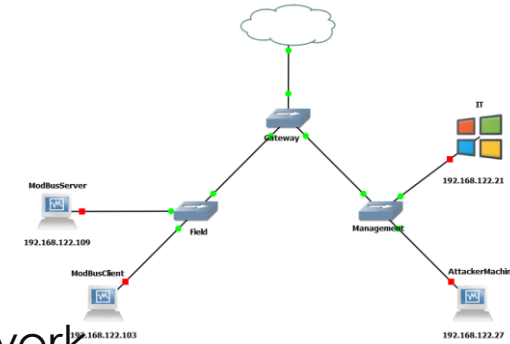
GNS3 Client for Network Modeling

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



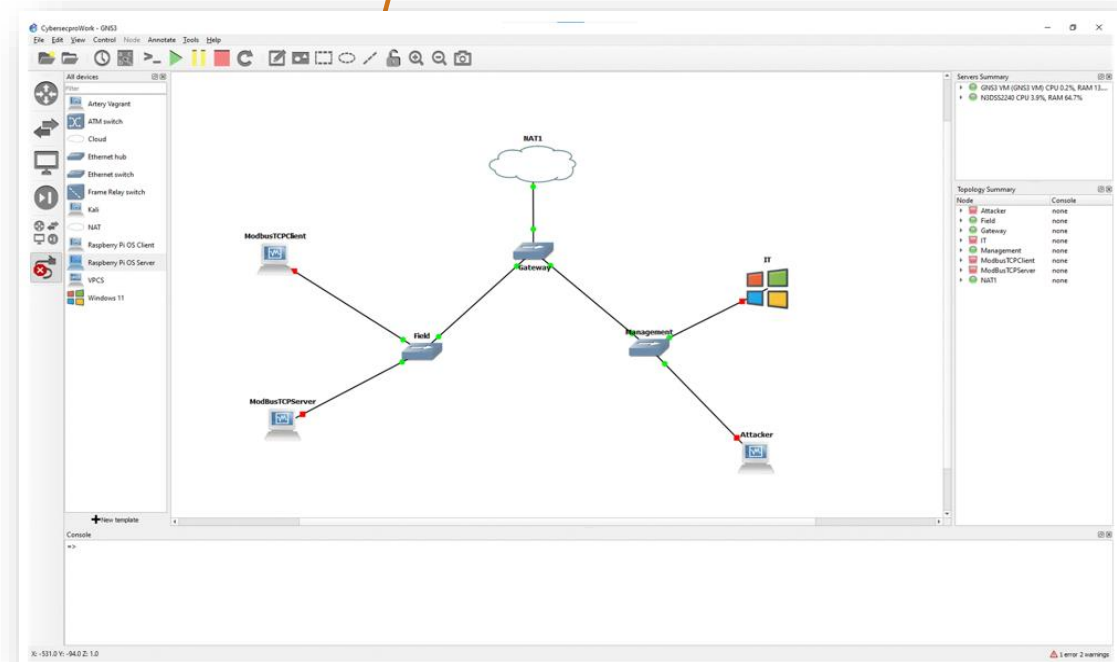
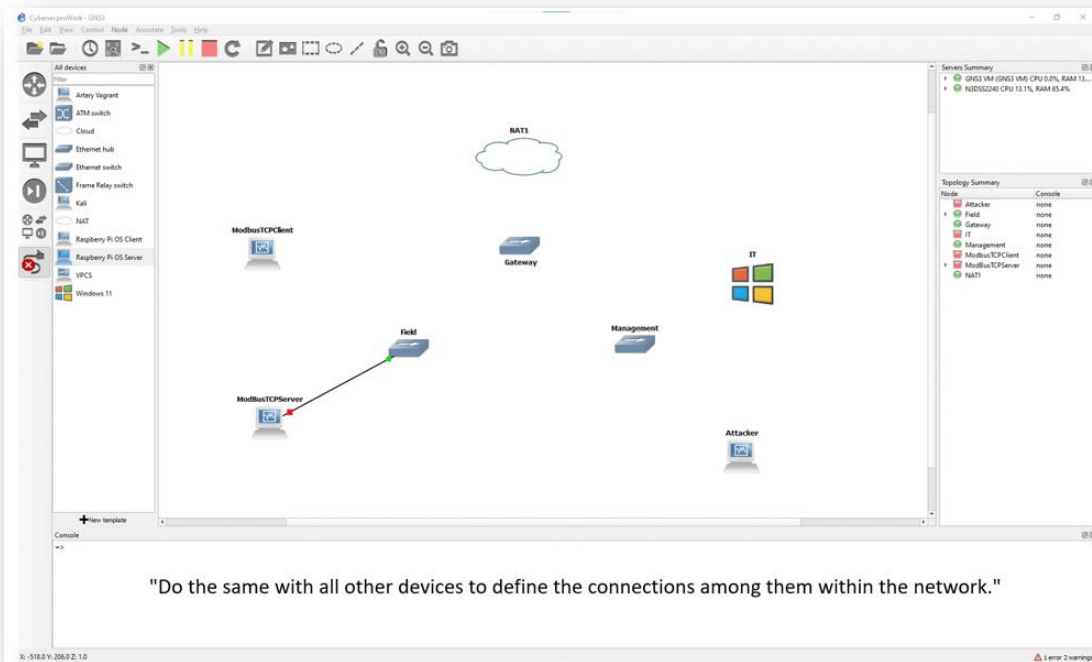
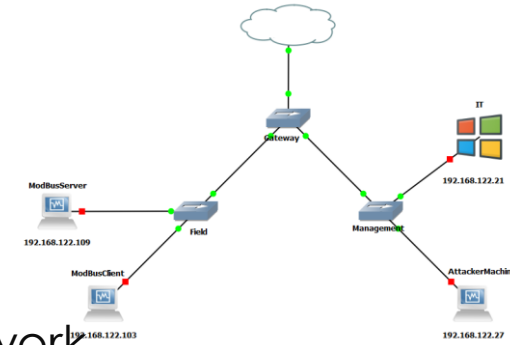
GNS3 Client for Network Modeling

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



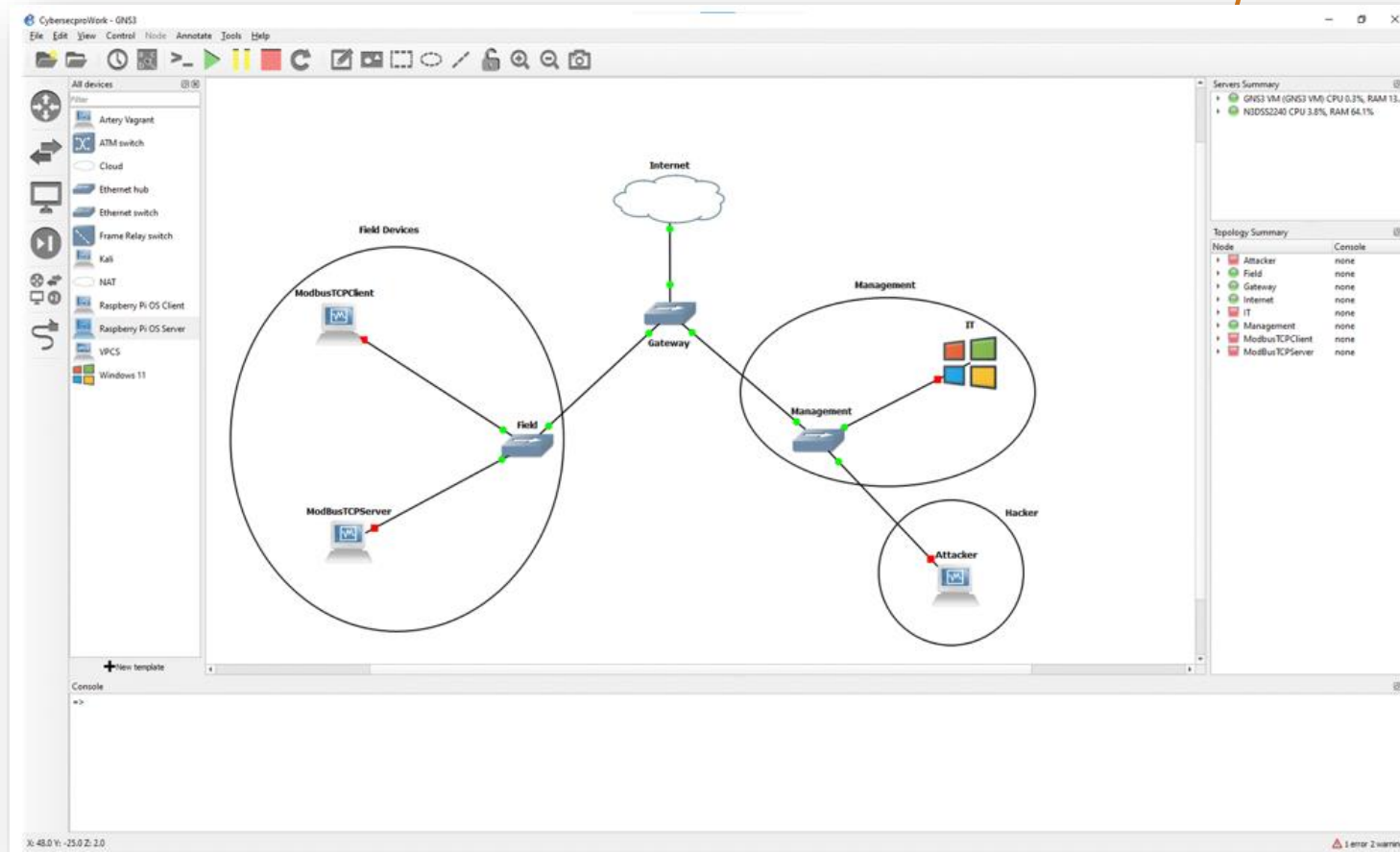
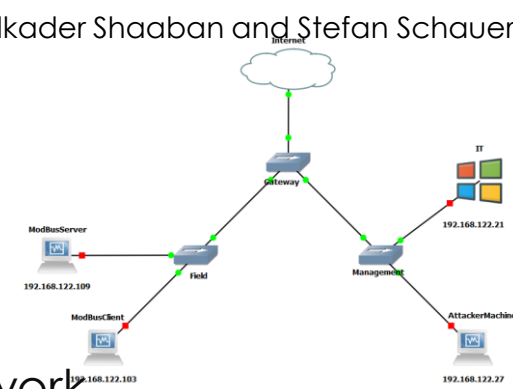
GNS3 Client for Network Modeling

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



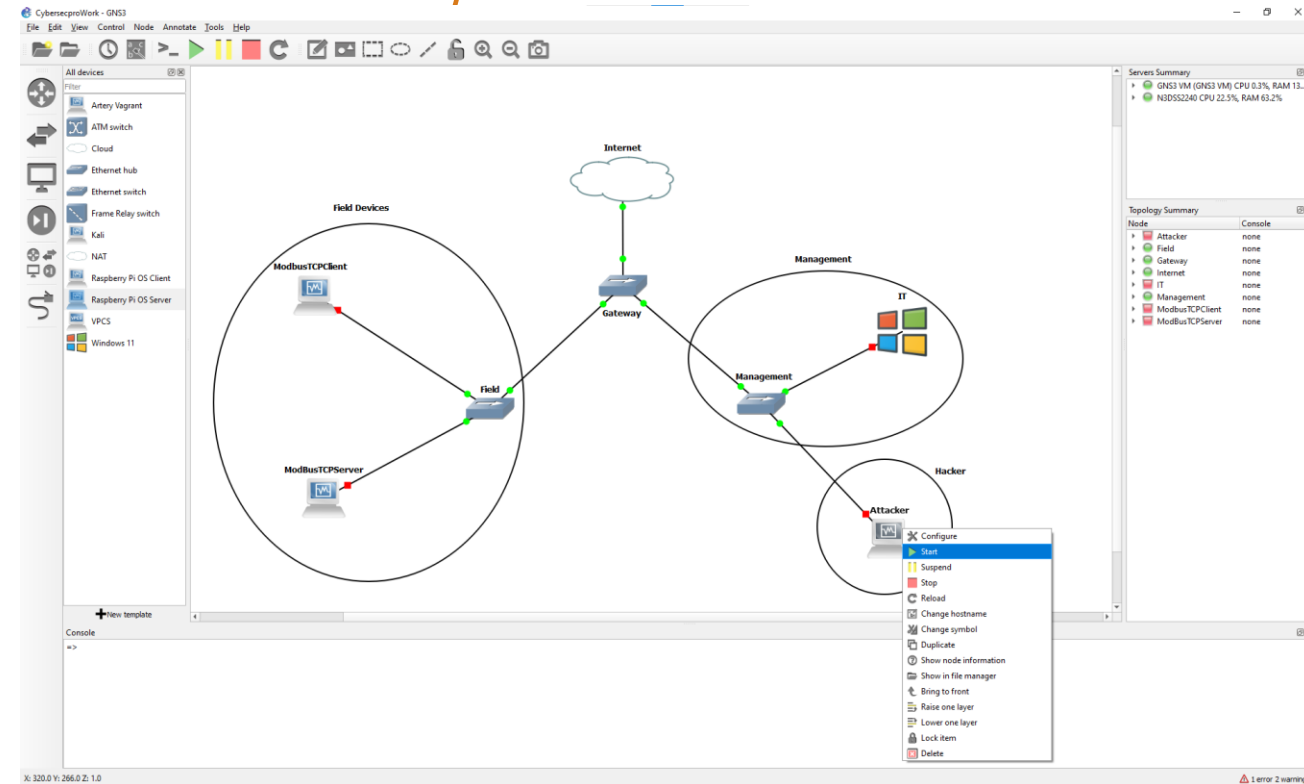
GNS3 Client for Network Modeling

- 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

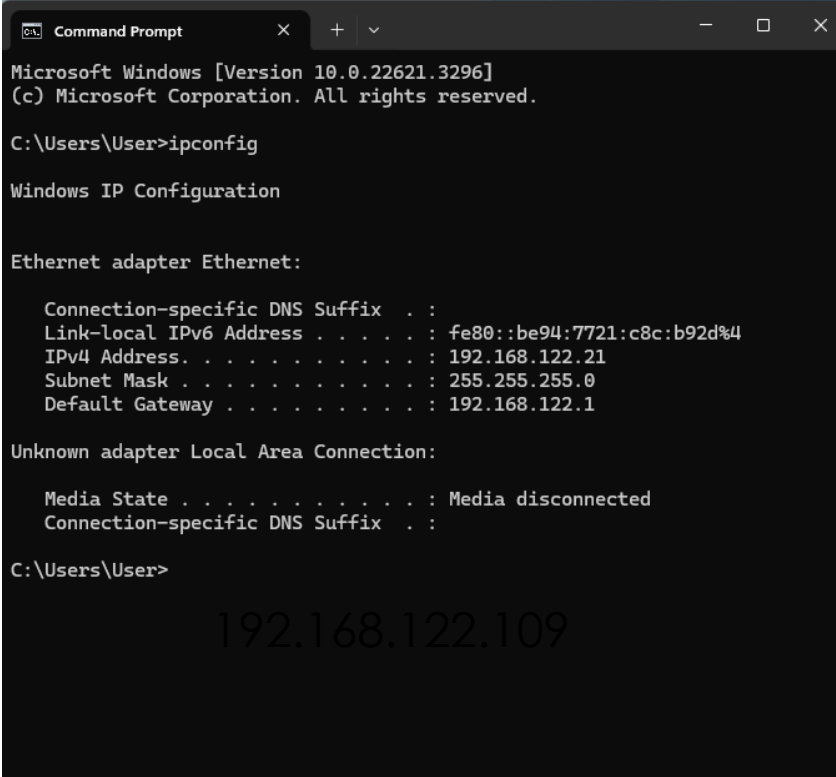
pi@raspberrypi:~$
  
```

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

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

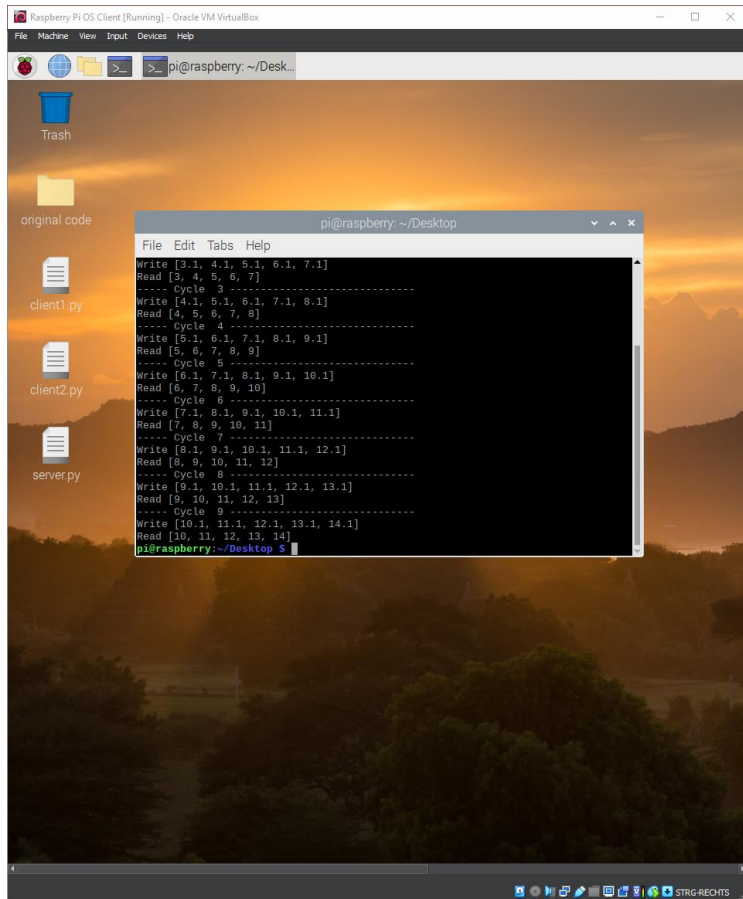
The screenshot shows the Wireshark interface with the following details:

- Packet List:** A table of captured packets. Packet 7772 is highlighted in black (TCP Out-Of-Order), packet 7778 is dark blue (DNS), and packet 7779 is green (TCP).
- Packet Details:** Shows the structure of a DNS query for NetBIOS Name Service, including Ethernet II, Internet Protocol Version 4, User Datagram Protocol, and NetBIOS Name Service.
- Packet Bytes:** Shows the raw hex and ASCII data of the captured packet.
- Status Bar:** Shows 12284 packets displayed (100.0%) and 0 dropped (0.0%).

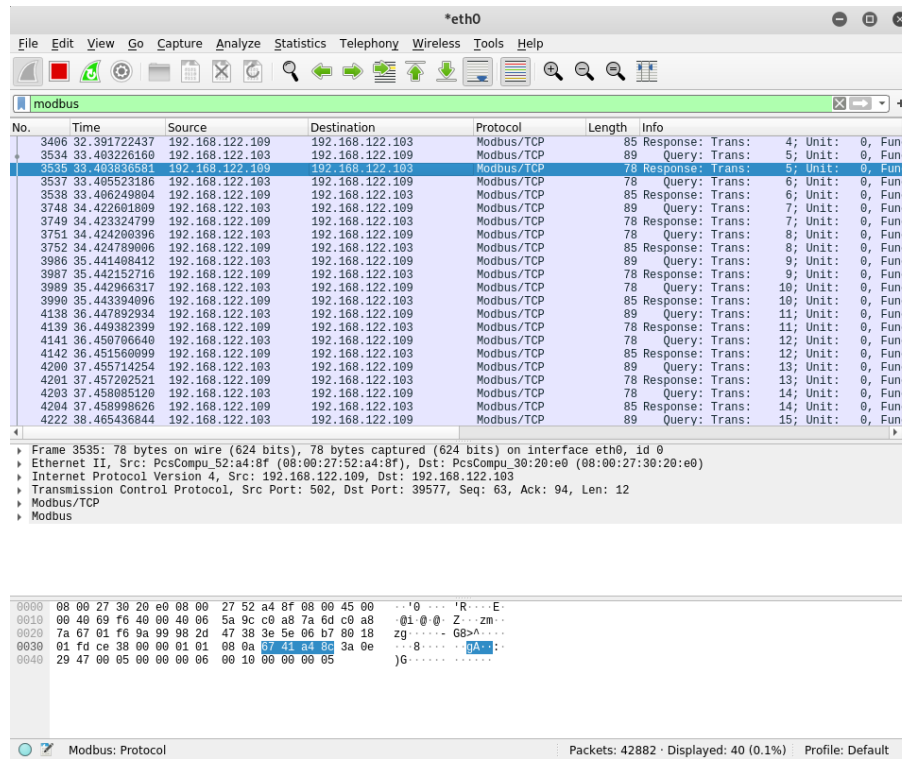
Wireshark – Client/Server Example

- 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

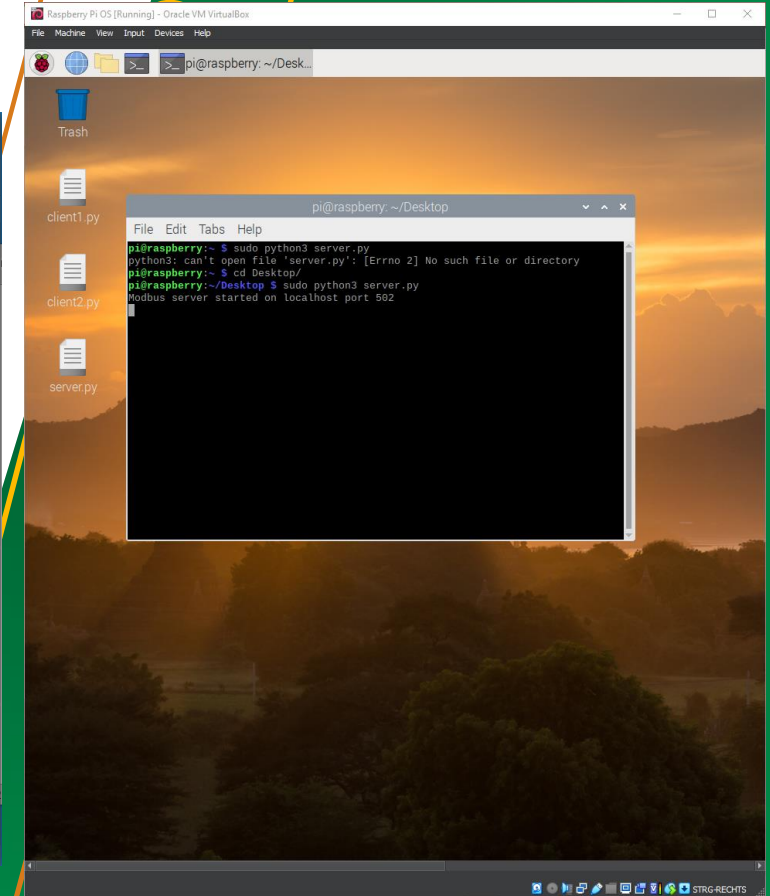
Wireshark - Client/Server Example



Client



Kali



Server

Thank you

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