Course Project: Milestone 2

Cesar Munguia

03/19/2024

—

IS-4543-001

—

Jose Mireles

Objectives

Each milestone submission will be a write-up that includes:

- What did you do?
- What did you learn?
- Documentation of your work
    - Screenshot
    - Picture
    - Video
    - Interpretive Dance

Project Proposal

The project involves in-depth analysis of a virtual machine infected with malware, focusing on understanding the malware's behavior, persistence mechanisms, and potential impact on the system. Through forensic techniques and dynamic analysis, I will dissect the malicious code, identify evasion tactics, and extract indicators of compromise. The project aims to enhance cybersecurity skills by providing hands-on experience in malware analysis and incident response, crucial for defending against and mitigating the effects of sophisticated cyber threats.

- Milestone 2: Download REMNUX to run fake services via INETSIM and check connectivity between victim machine and REMNUX

In Milestone 2 I will be mainly focusing on downloading the REMNUX ova file from their official website. REMUX is basically what digital foresters/SOC analysts use to help them in their malware analysis, just like penetration testers use Kali Linux for their own needs. This virtual machine comes with a special service that is very helpful, which is INETSIM. INETSIM is a software that simulates common internet services in a secure environment to analyze malware behavior.

Summary of Activities

I started by going into the official website to download the REMNUX ova file (www.remnux.org, refer to screenshot 1). This ova file was basically a Linux virtual machine, and I did not have to install it directly on VMWare. Instead, I double-clicked on the virtual machine configuration file and the VM opened up by itself. After I booted up the machine, I entered the credentials "remnux" for username and "malware" for the password. So, I decided to open up a terminal and go to the network configuration file, in which I used vi to edit the file. The command used was **sudo vi /etc/netplan/01-netcfg.yaml** (refer to screenshot 2). I was basically going to change the IP address to be static, and once I was in the file editor I pressed "I" to enter insert mode and changed dhcp4 to no. I also set the IPv4 address to be 192.168.10.3 with a netmask of 255.255.255.0 or /24 in CIDR notation, and set the gateway to be 192.168.10.1 (refer to screenshot 3). After making changes to the file I exited by pressing ESC and typing ":wq" to write and quit. Then, I used command **sudo netplan apply** (refer to screenshot 4) to apply the new network settings. No output error was shown and I was able to move on. The next step was to double-check that the new IP address was set correctly. For that I used commands **ip a**, **ip -br -c a**, and **ifconfig** to confirm that the IP address was correct and set to 192.168.10.3 (refer to screenshot 5). Then, I checked the connectivity of the REMNUX machine and the victim's machine (Win10 VM), and for that I used the **ping** command. Based on the ping outputs, I was able to confirm that both machines were able to communicate with each other (refer to screenshot 6 & 7).

Now, it was time to set up the INETSIM software that was already pre-installed in this Linux VM. For that, I used command **sudo vi /etc/inetsim/inetsim.conf** to edit the configuration file. Once inside the file editor I read the comment sections to understand a little bit about the settings. I noticed that there were many services available for me to choose from. When I say services I mean services like DNS, HTTP, HTTPS, FTP, POP3, etc. Therefore, I decided to choose only the ones I thought were the most significant and useful. The ones I did not want I commented them out by using "#" (refer to screenshot 8). Then, I scrolled down to change the service bind address section. I set the service_bind_address to 0.0.0.0 (refer to screenshot 9). Again, I scrolled down to the DNS default IP section. I set the dns_default_ip to 192.168.10.3 which is the static IP of REMNUX (refer to screenshot 10). I saved the exited the file and executed command **inetsim** to start the simulation (refer to screenshot 11).

Description of Learning Completed

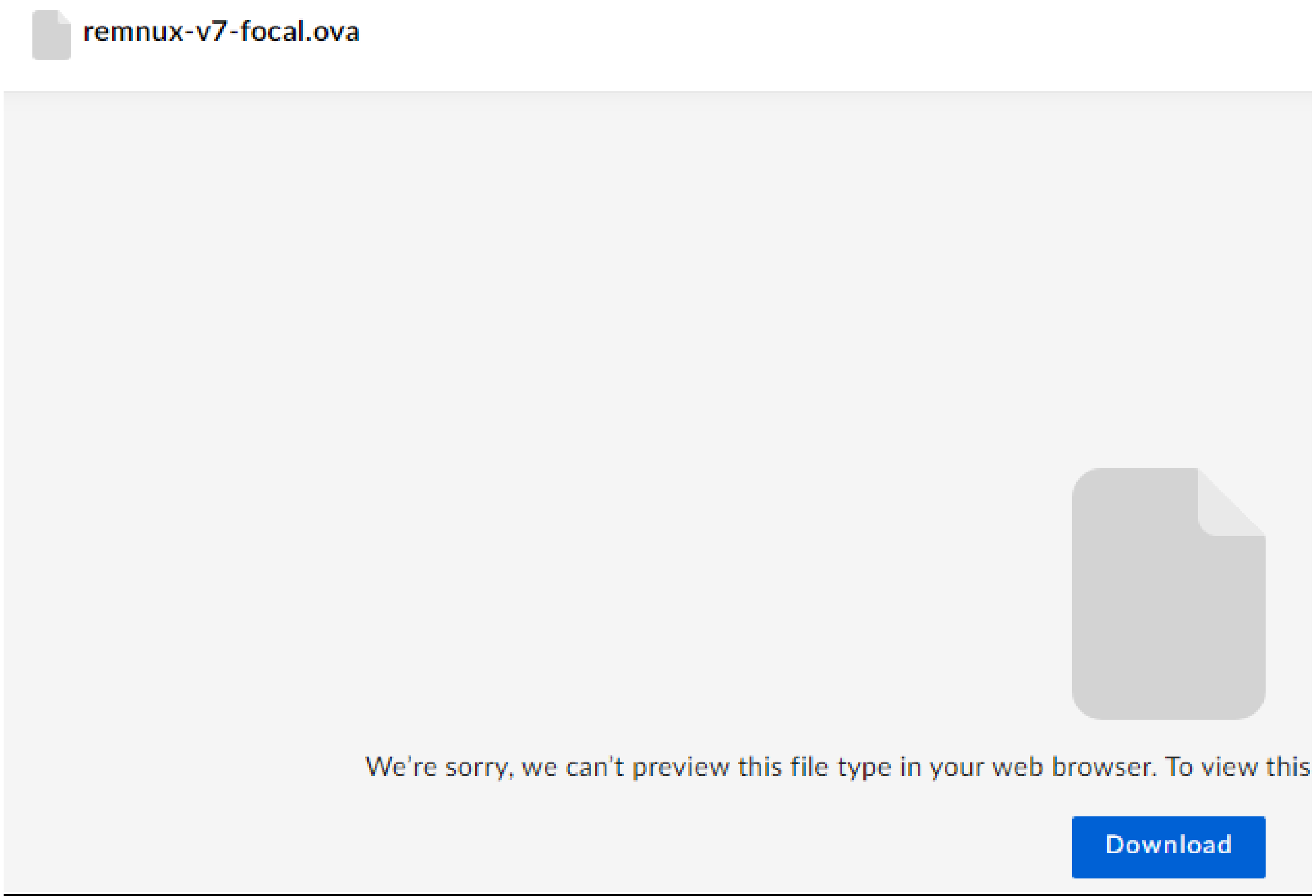
In this particular milestone, I learned that in order to be able to have communication between the victim's machine and the REMNUX machine I needed to make a couple of changes in both. I started by setting the network adapters of both machines to "Host-Only". This would make sure that they were not allowed to access the external world, a.k.a. the Internet. I then strategically changed the IP address of both VMs so that it would appear as if they were in a peer-to-peer network. The victim's IP was set to 192.168.10.2 and REMNUX to 192.168.10.3 so they're the only ones in the network.

This was a peer-to-peer network because there was no centralized server, however, in this virtual environment the victim can be said to represent and act as a client and REMNUX represented and acted as the server. In the next milestone where the malware is going to be "accidently" executed by me I want to know exactly what this malicious is going to do and who it will try to reach in the external network. The whole purpose of having REMNUX is to see if the malware tries to, for example, download more malware from the web, or try to establish connectivity, for example, with a command-and-control server which are heavily used by malicious actors when trying to control the victim's computer. INETSIM will simulate all of this without having to access the "real" Internet.

Whenever I set the bind_address_address to 0.0.0.0 I did it because it basically binded all interfaces to the host (Linux VM). Once I ran the **inetsim** command in REMNUX, I quickly switched to the Win10 machine to confirm my previous arguments. I opened up Google Chrome and typed the IP address of REMNUX (192.168.10.3) and I was able to "reach" the website via the HTTP protocol (refer to screenshot 12). Then, on that same site I added /malware.exe to simulate a download and the browser actually downloaded a file called "malware.exe" (refer to screenshot 13 & 14). This file is actually not malicious, it's just a default file that REMNUX feeds to the "client". Finally, I typed thiswebsitedoesnotexits.com to confirm that the browser was able to "reach" the inexistent website because of the simulation software that is currently running in REMNUX (refer to screenshot 15). This VM is basically working as the DNS server, as well as other types of servers.

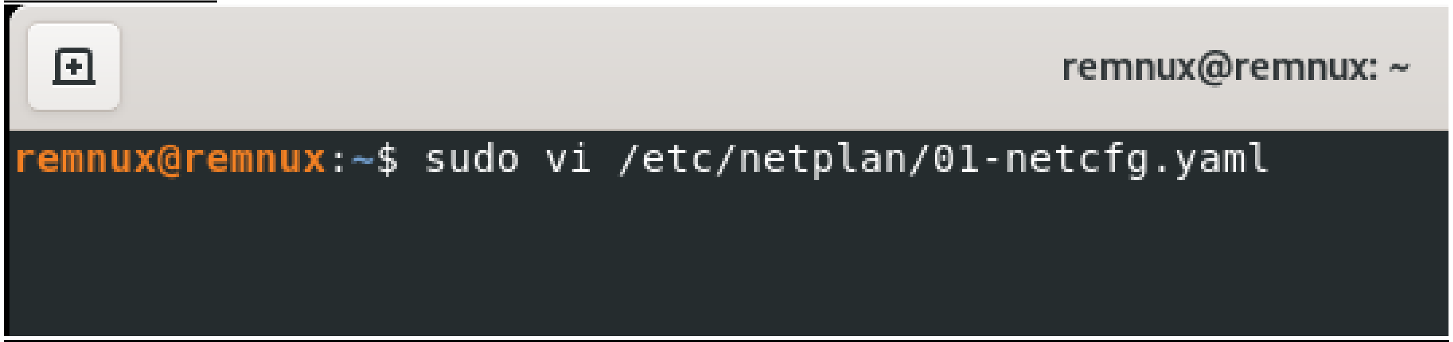
Documentation of Work Completed

Screenshot 1


remnux-v7-focal.ova

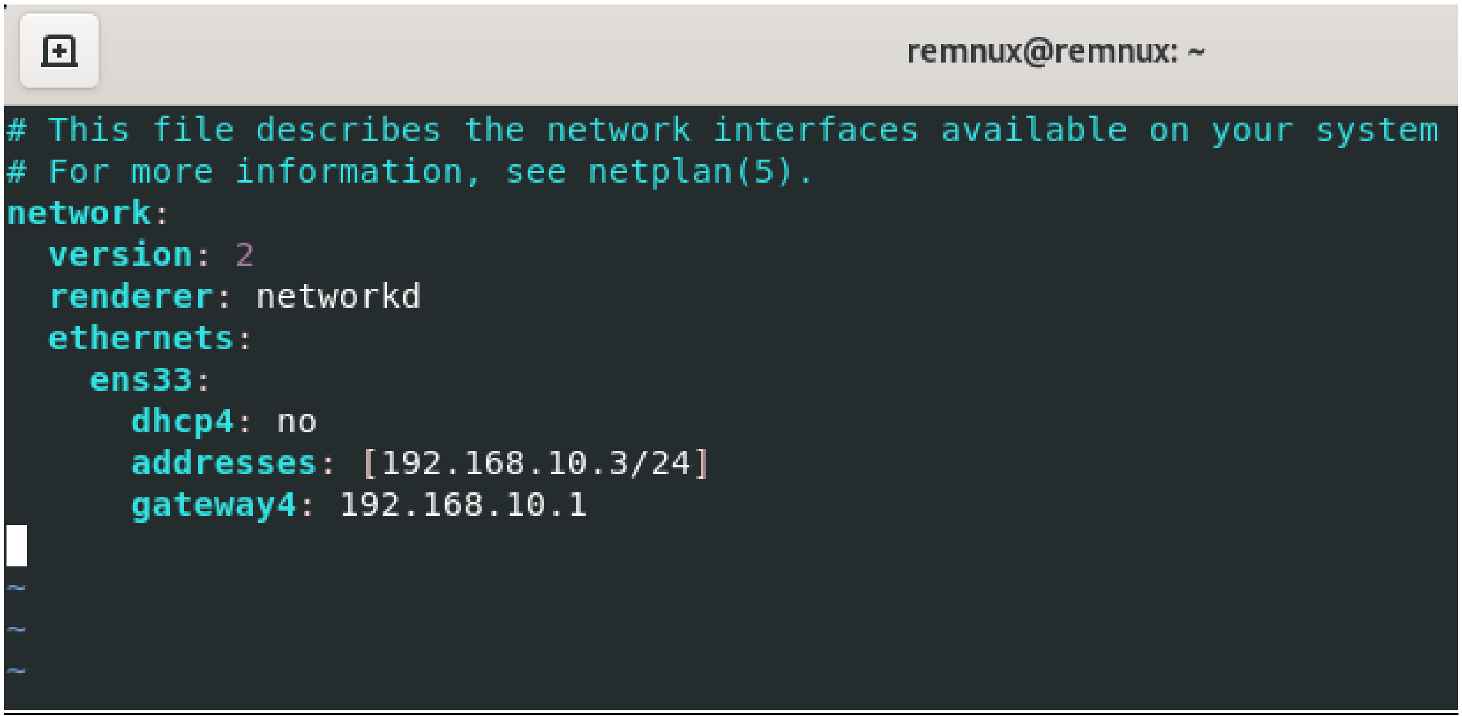
We're sorry, we can't preview this file type in your web browser. To view this

Screenshot 2



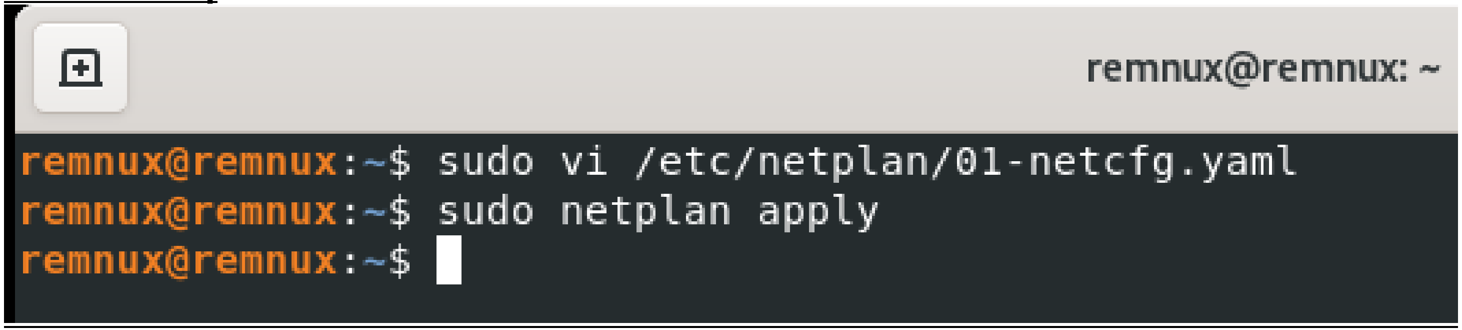
```
remnux@remnux:~$ sudo vi /etc/netplan/01-netcfg.yaml
```

Screenshot 3



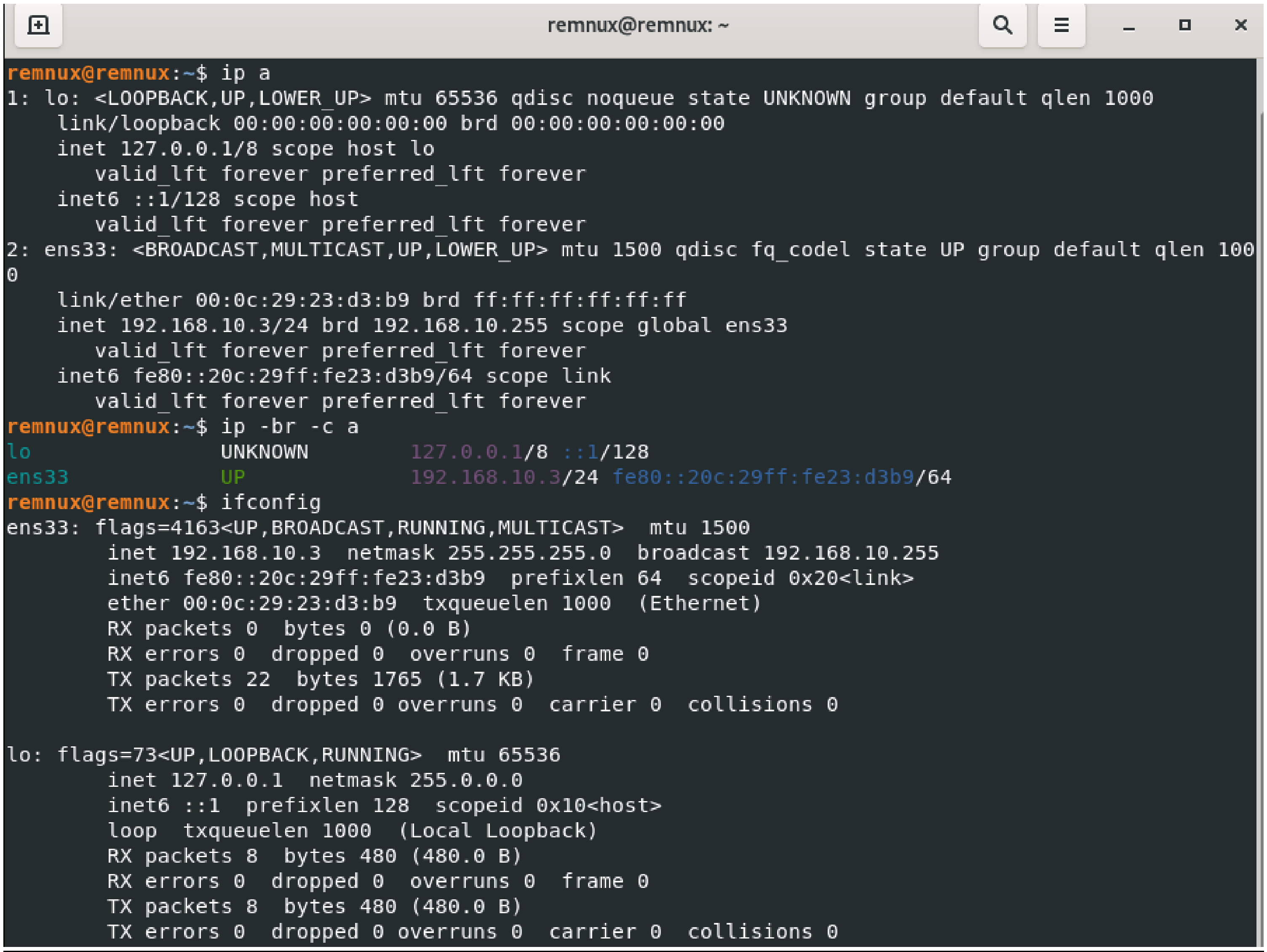
```
# This file describes the network interfaces available on your system
# For more information, see netplan(5).
network:
  version: 2
  renderer: networkd
  ethernets:
    ens33:
      dhcp4: no
      addresses: [192.168.10.3/24]
      gateway4: 192.168.10.1

~
~
~
```

<u>Screenshot 4</u>



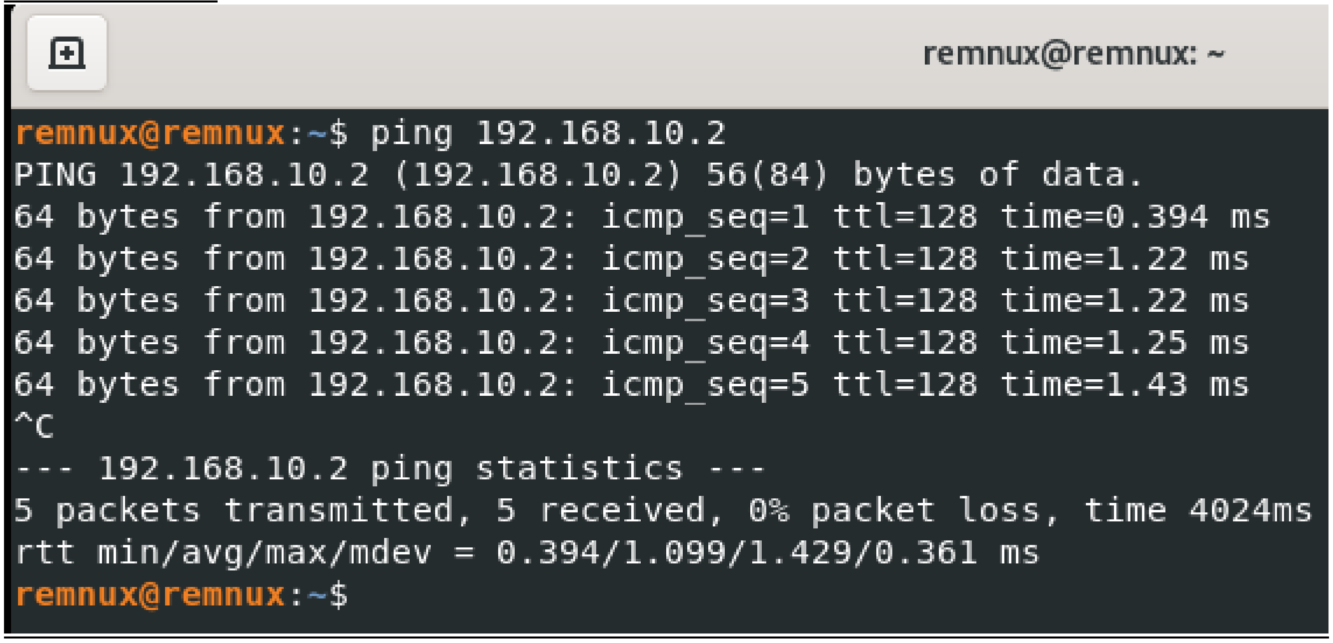
```
remnux@remnux:~$ sudo vi /etc/netplan/01-netcfg.yaml
remnux@remnux:~$ sudo netplan apply
remnux@remnux:~$ 
```

<u>Screenshot 5</u>



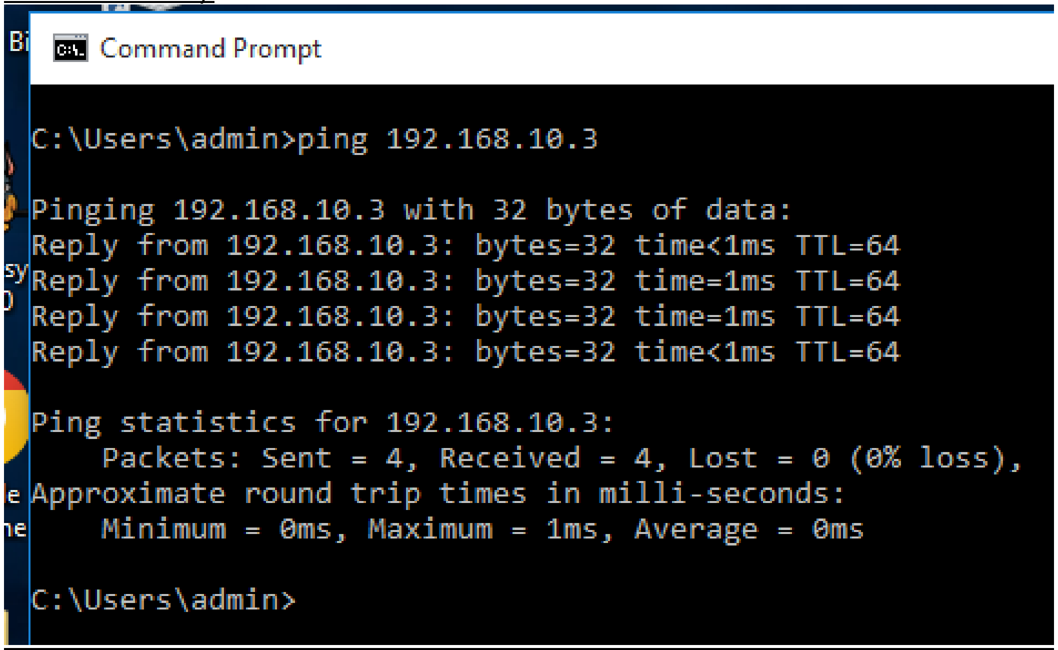
```
remnux@remnux:~$ ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
       valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host
       valid_lft forever preferred_lft forever
2: ens33: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen 100
0
    link/ether 00:0c:29:23:d3:b9 brd ff:ff:ff:ff:ff:ff
    inet 192.168.10.3/24 brd 192.168.10.255 scope global ens33
       valid_lft forever preferred_lft forever
    inet6 fe80::20c:29ff:fe23:d3b9/64 scope link
       valid_lft forever preferred_lft forever
remnux@remnux:~$ ip -br -c a
lo               UNKNOWN        127.0.0.1/8 ::1/128
ens33            UP             192.168.10.3/24 fe80::20c:29ff:fe23:d3b9/64
remnux@remnux:~$ ifconfig
ens33: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 192.168.10.3  netmask 255.255.255.0  broadcast 192.168.10.255
        inet6 fe80::20c:29ff:fe23:d3b9  prefixlen 64  scopeid 0x20<link>
        ether 00:0c:29:23:d3:b9  txqueuelen 1000  (Ethernet)
        RX packets 0  bytes 0 (0.0 B)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 22  bytes 1765 (1.7 KB)
        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 8  bytes 480 (480.0 B)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 8  bytes 480 (480.0 B)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0
```

Screenshot 6



```
remnux@remnux:~$ ping 192.168.10.2
PING 192.168.10.2 (192.168.10.2) 56(84) bytes of data.
64 bytes from 192.168.10.2: icmp_seq=1 ttl=128 time=0.394 ms
64 bytes from 192.168.10.2: icmp_seq=2 ttl=128 time=1.22 ms
64 bytes from 192.168.10.2: icmp_seq=3 ttl=128 time=1.22 ms
64 bytes from 192.168.10.2: icmp_seq=4 ttl=128 time=1.25 ms
64 bytes from 192.168.10.2: icmp_seq=5 ttl=128 time=1.43 ms
^C
--- 192.168.10.2 ping statistics ---
5 packets transmitted, 5 received, 0% packet loss, time 4024ms
rtt min/avg/max/mdev = 0.394/1.099/1.429/0.361 ms
remnux@remnux:~$
```

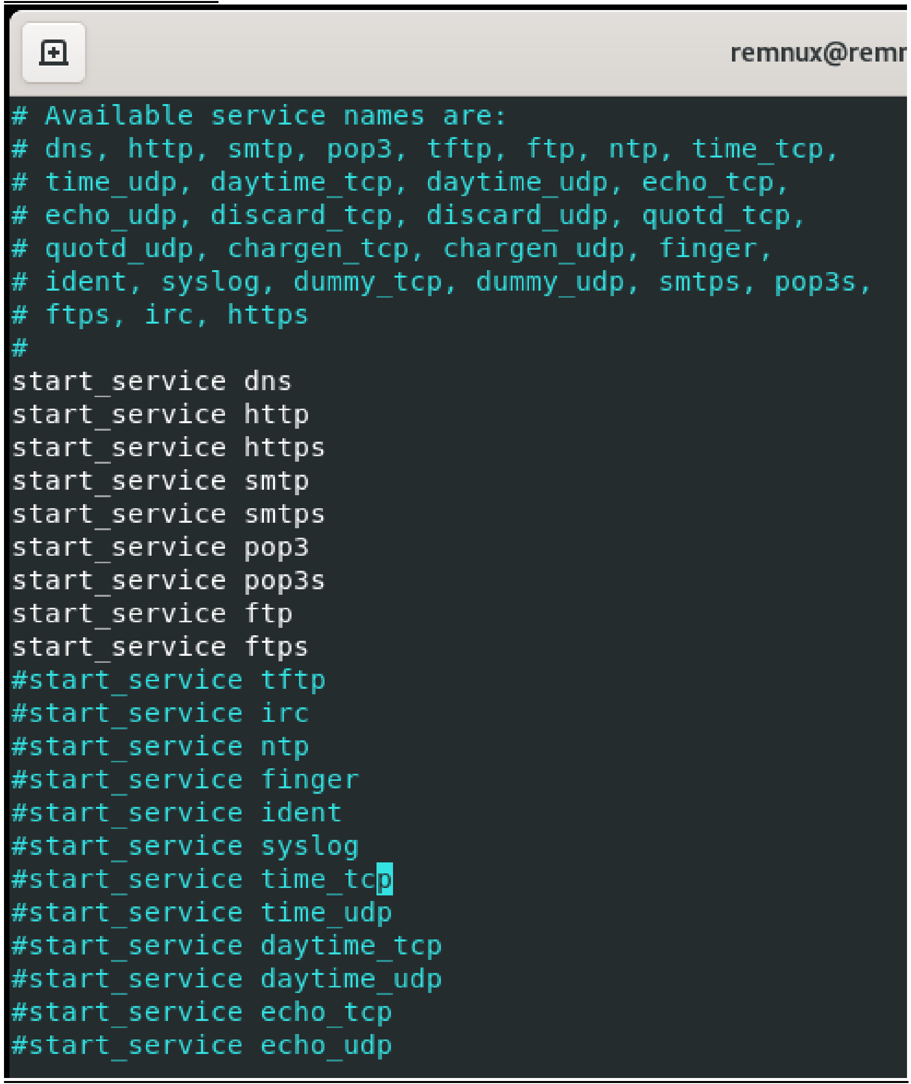
Screenshot 7



Command Prompt

```
C:\Users\admin>ping 192.168.10.3

Pinging 192.168.10.3 with 32 bytes of data:
Reply from 192.168.10.3: bytes=32 time<1ms TTL=64
Reply from 192.168.10.3: bytes=32 time=1ms TTL=64
Reply from 192.168.10.3: bytes=32 time=1ms TTL=64
Reply from 192.168.10.3: bytes=32 time<1ms TTL=64

Ping statistics for 192.168.10.3:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 0ms, Maximum = 1ms, Average = 0ms

C:\Users\admin>
```

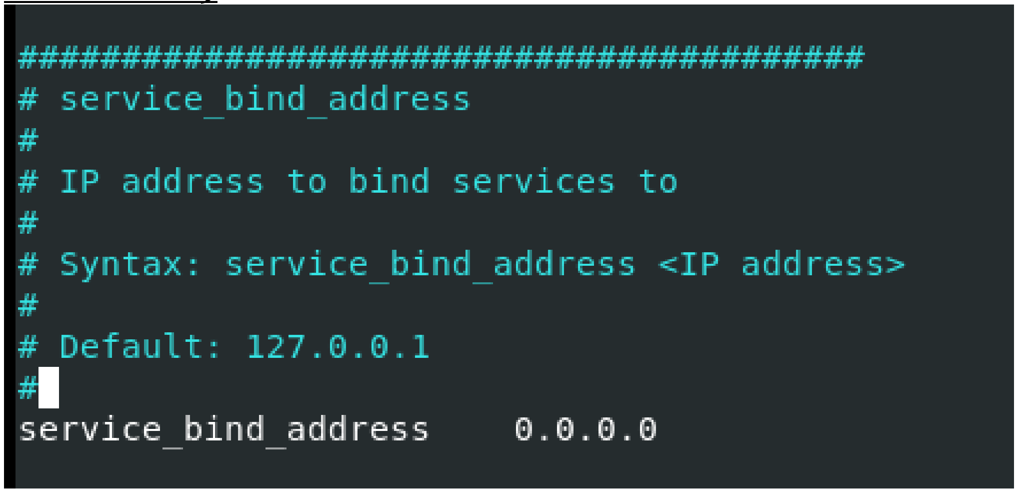
Screenshot 8



remnux@remr

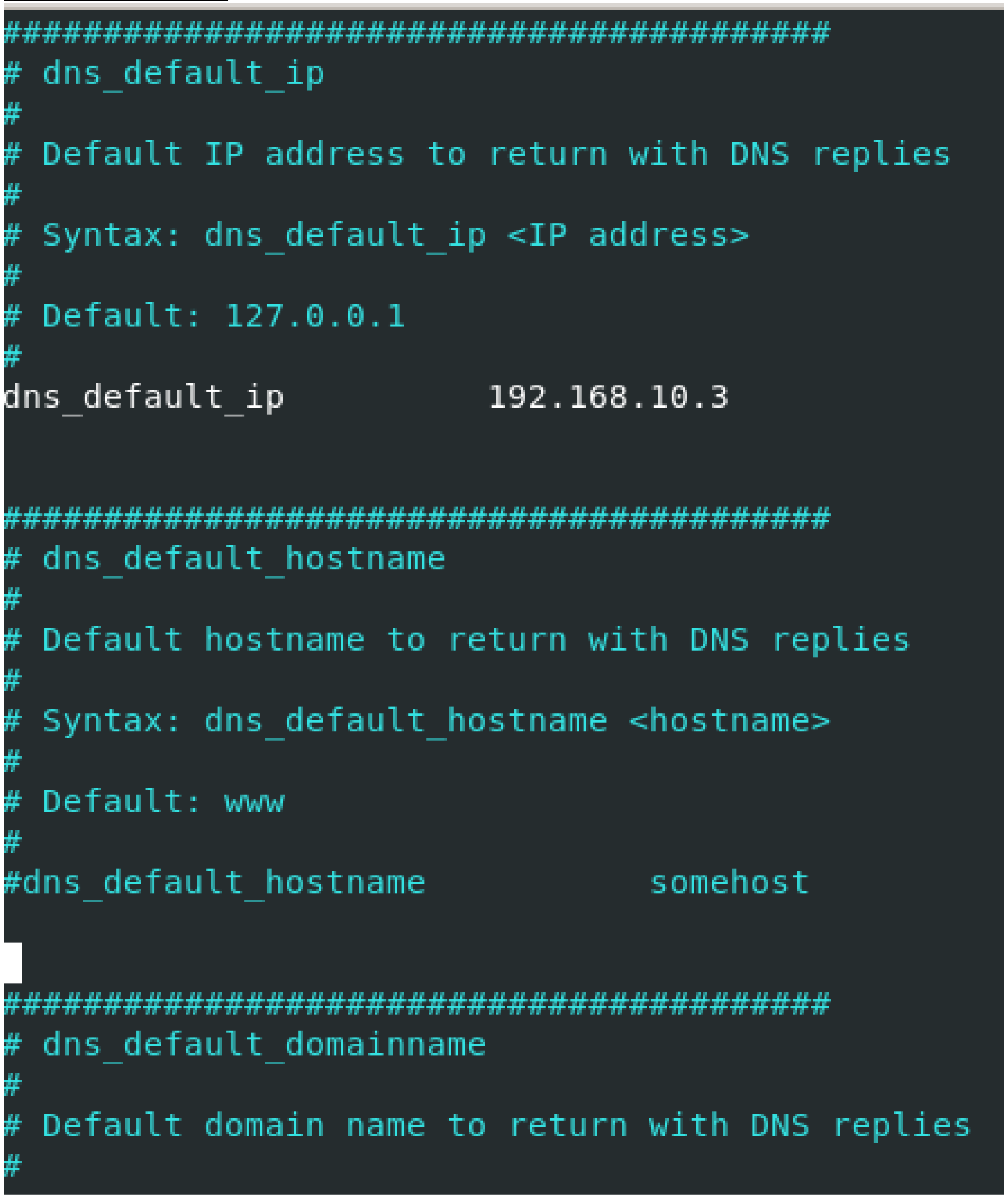
```
# Available service names are:
# dns, http, smtp, pop3, tftp, ftp, ntp, time_tcp,
# time_udp, daytime_tcp, daytime_udp, echo_tcp,
# echo_udp, discard_tcp, discard_udp, quotd_tcp,
# quotd_udp, chargen_tcp, chargen_udp, finger,
# ident, syslog, dummy_tcp, dummy_udp, smtps, pop3s,
# ftps, irc, https
#
start_service dns
start_service http
start_service https
start_service smtp
start_service smtps
start_service pop3
start_service pop3s
start_service ftp
start_service ftps
#start_service tftp
#start_service irc
#start_service ntp
#start_service finger
#start_service ident
#start_service syslog
#start_service time_tcp
#start_service time_udp
#start_service daytime_tcp
#start_service daytime_udp
#start_service echo_tcp
#start_service echo_udp
```

Screenshot 9



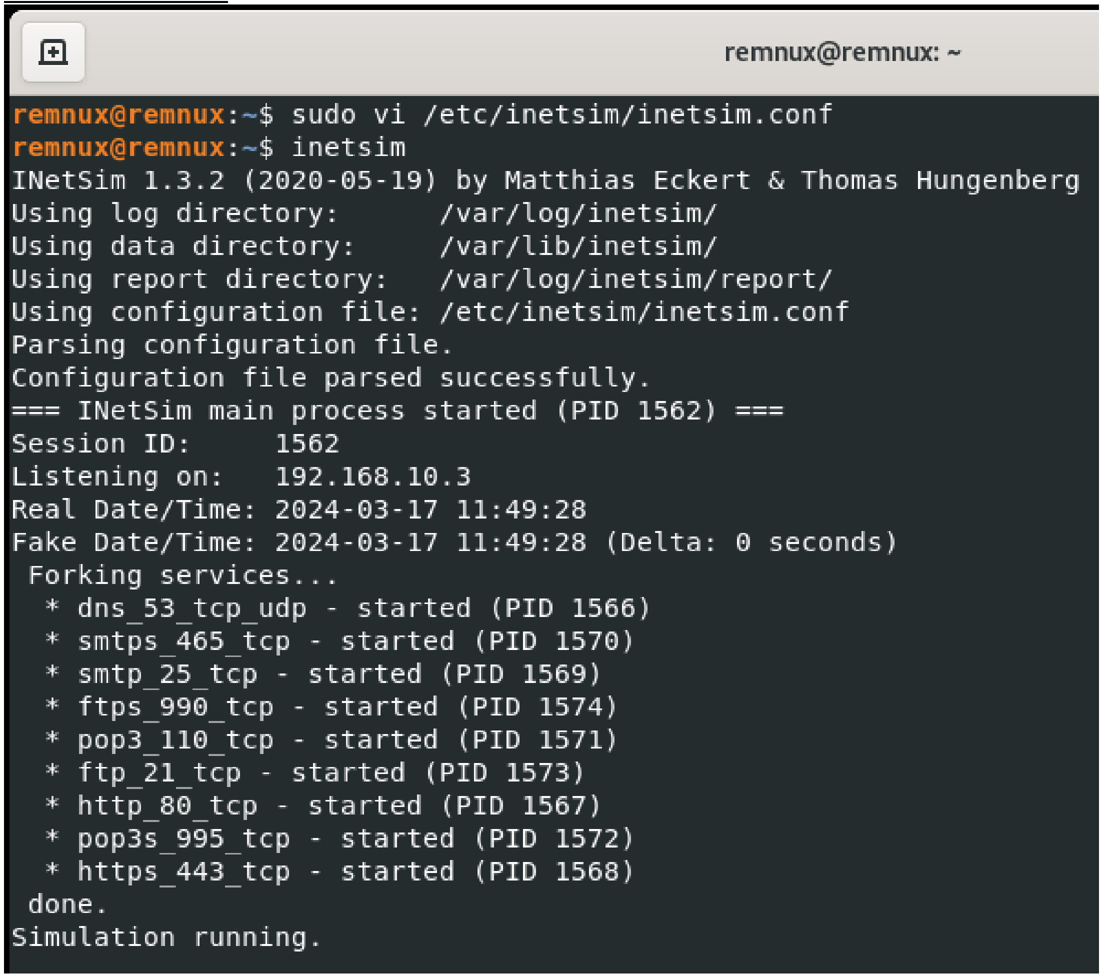
```
############################################
# service_bind_address
#
# IP address to bind services to
#
# Syntax: service_bind_address <IP address>
#
# Default: 127.0.0.1
#
service_bind_address    0.0.0.0
```

Screenshot 10



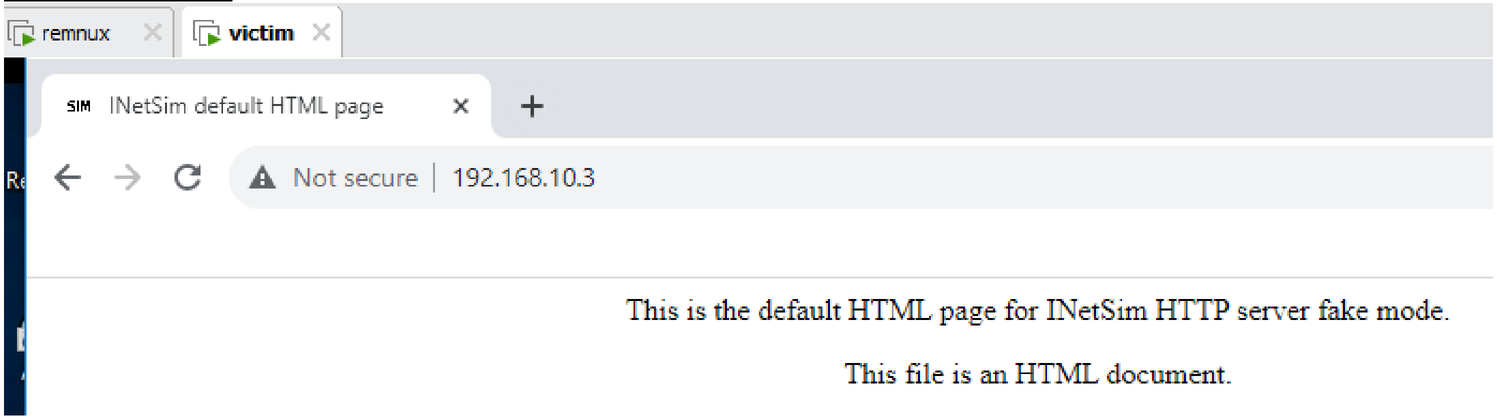
```
############################################
# dns_default_ip
#
# Default IP address to return with DNS replies
#
# Syntax: dns_default_ip <IP address>
#
# Default: 127.0.0.1
#
dns_default_ip          192.168.10.3


############################################
# dns_default_hostname
#
# Default hostname to return with DNS replies
#
# Syntax: dns_default_hostname <hostname>
#
# Default: www
#
#dns_default_hostname           somehost



############################################
# dns_default_domainname
#
# Default domain name to return with DNS replies
#
```

Screenshot 11



```
remnux@remnux:~$ sudo vi /etc/inetsim/inetsim.conf
remnux@remnux:~$ inetsim
INetSim 1.3.2 (2020-05-19) by Matthias Eckert & Thomas Hungenberg
Using log directory:      /var/log/inetsim/
Using data directory:     /var/lib/inetsim/
Using report directory:   /var/log/inetsim/report/
Using configuration file: /etc/inetsim/inetsim.conf
Parsing configuration file.
Configuration file parsed successfully.
=== INetSim main process started (PID 1562) ===
Session ID:     1562
Listening on:   192.168.10.3
Real Date/Time: 2024-03-17 11:49:28
Fake Date/Time: 2024-03-17 11:49:28 (Delta: 0 seconds)
 Forking services...
  * dns_53_tcp_udp - started (PID 1566)
  * smtps_465_tcp - started (PID 1570)
  * smtp_25_tcp - started (PID 1569)
  * ftps_990_tcp - started (PID 1574)
  * pop3_110_tcp - started (PID 1571)
  * ftp_21_tcp - started (PID 1573)
  * http_80_tcp - started (PID 1567)
  * pop3s_995_tcp - started (PID 1572)
  * https_443_tcp - started (PID 1568)
 done.
Simulation running.
```

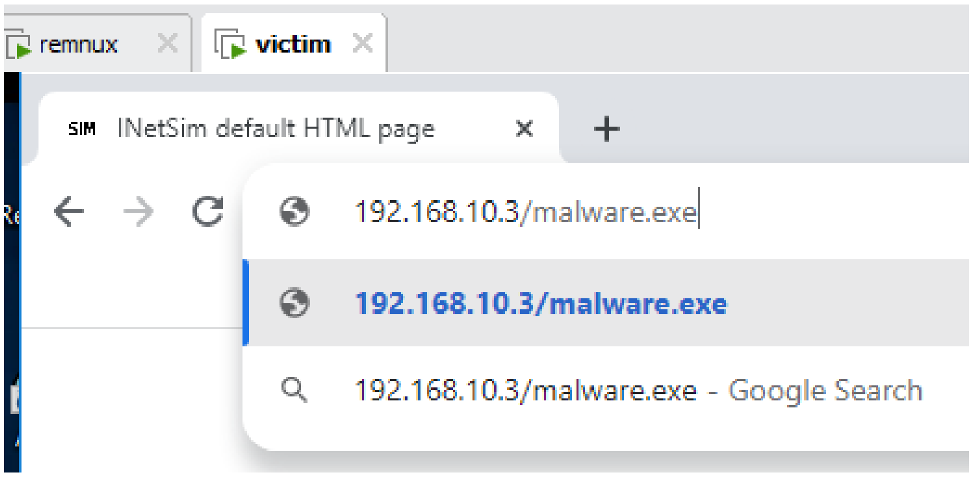
Screenshot 12



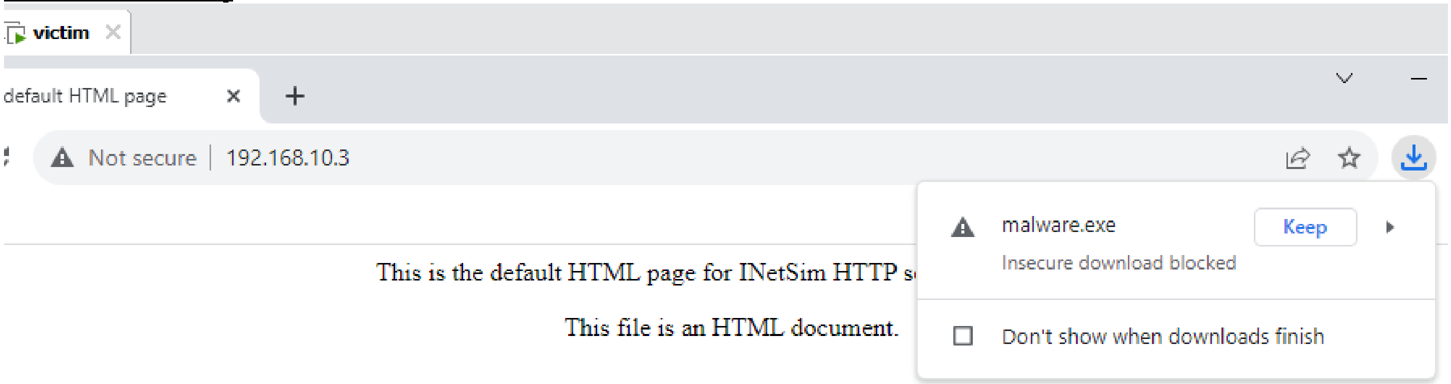
This is the default HTML page for INetSim HTTP server fake mode.

This file is an HTML document.

Screenshot 13


remnux
victim
INetSim default HTML page
192.168.10.3/malware.exe
192.168.10.3/malware.exe
192.168.10.3/malware.exe - Google Search

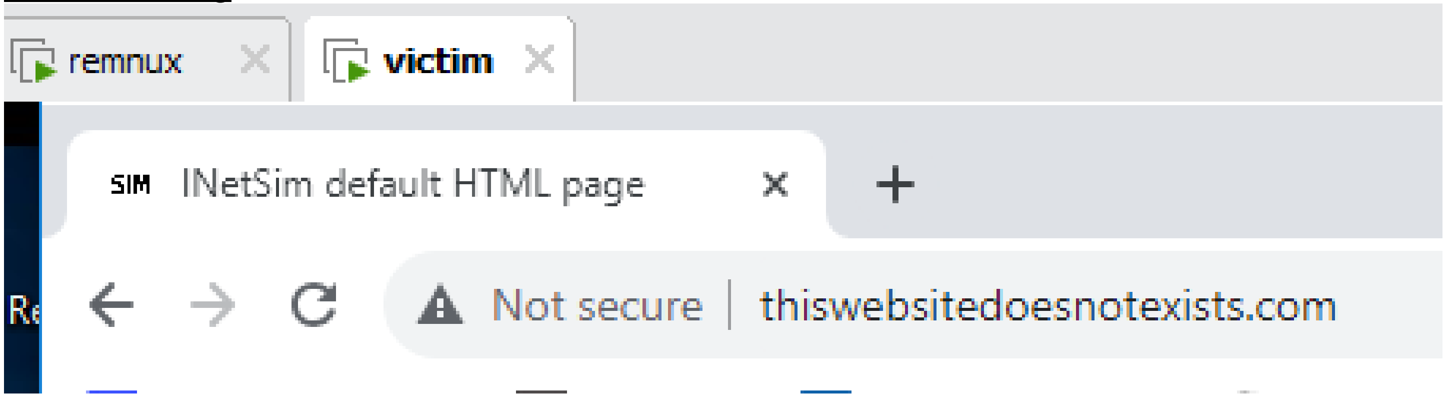
Screenshot 14


victim
default HTML page
Not secure | 192.168.10.3
malware.exe
Insecure download blocked
Keep
Don't show when downloads finish

This is the default HTML page for INetSim HTTP s

This file is an HTML document.

Screenshot 15


remnux
victim
INetSim default HTML page
Not secure | thiswebsitedoesnotexists.com

References

INetSim. (n.d.). https://www.inetsim.org/

REMnux. (n.d.). https://remnux.org/