

# **LAB 1: Setting up Environment**

## **I. Purpose**

Set up the environment for malware testing by using Kali Linux as DNS server, and Windows Server 2008 as victim running on VMware to simulate and observe simulated network traffic using INetSim

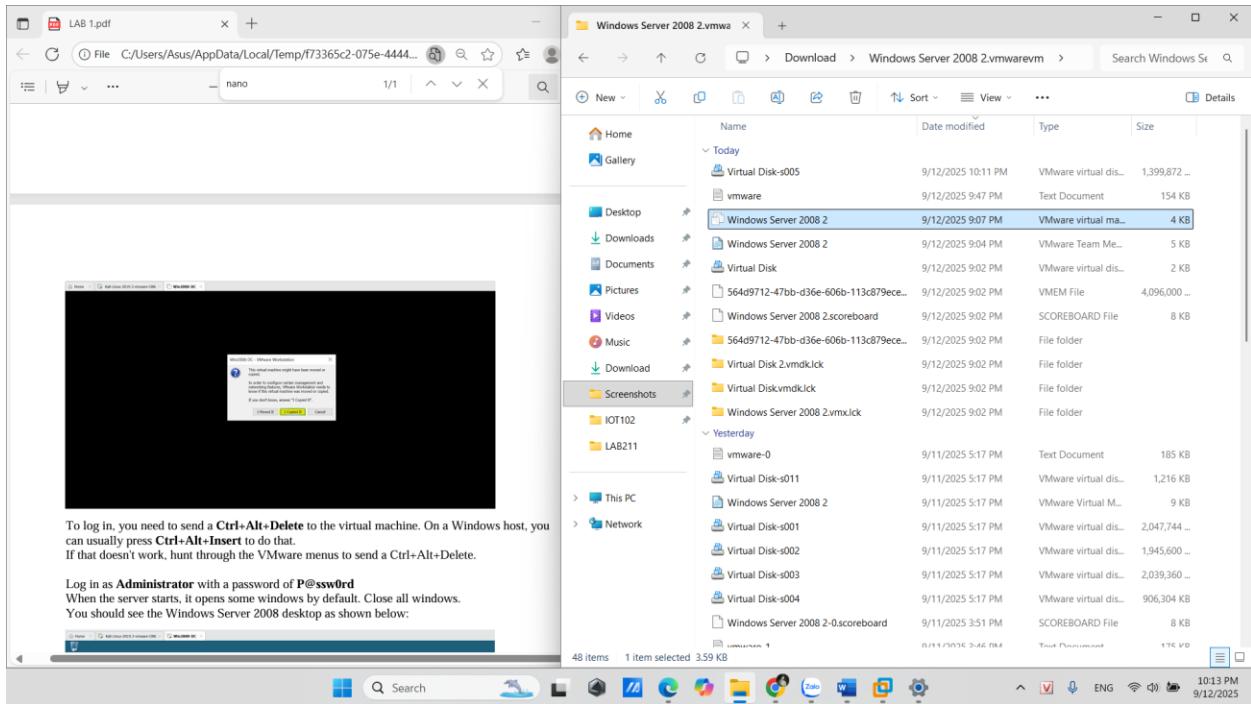
## **II. Target**

- Configure Kali (NAT) and INetSim to respond to DNS/HTTP queries from Windows machines
- Set up Window VM to point DNS to Kali and check InetSim page via browser
- Use nmap on Windows to scan fake domain (YOURNAME.com) and record services simulated by INetSim

## **III. Lab instruction:**

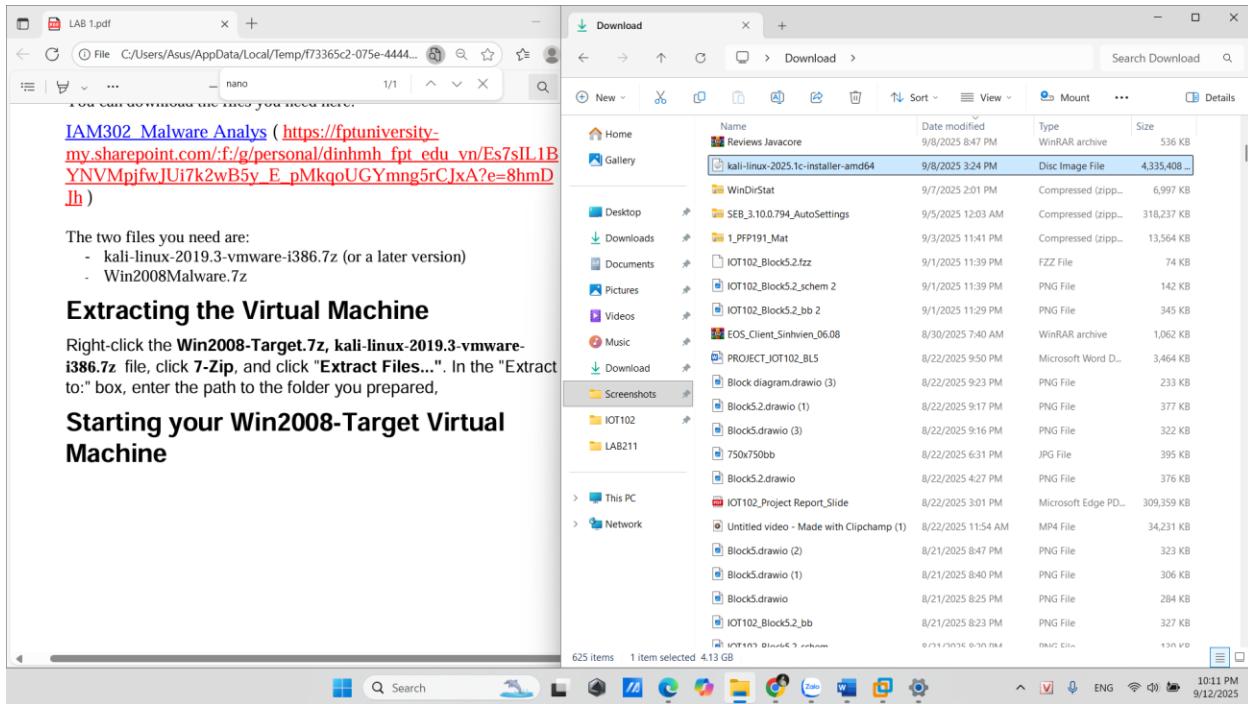
### **1. Set up virtual machine**

Firstly, download and unzip the file Window Server 2008



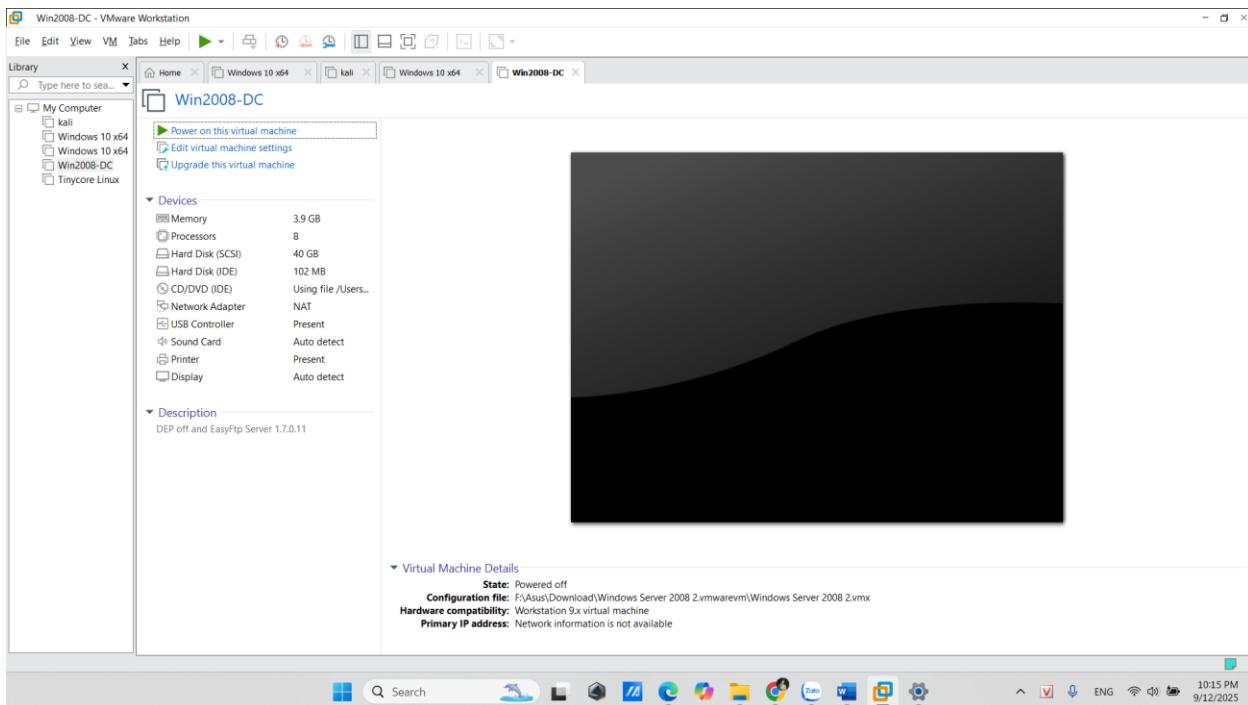
**Fig 1.1: Window Server 2008**

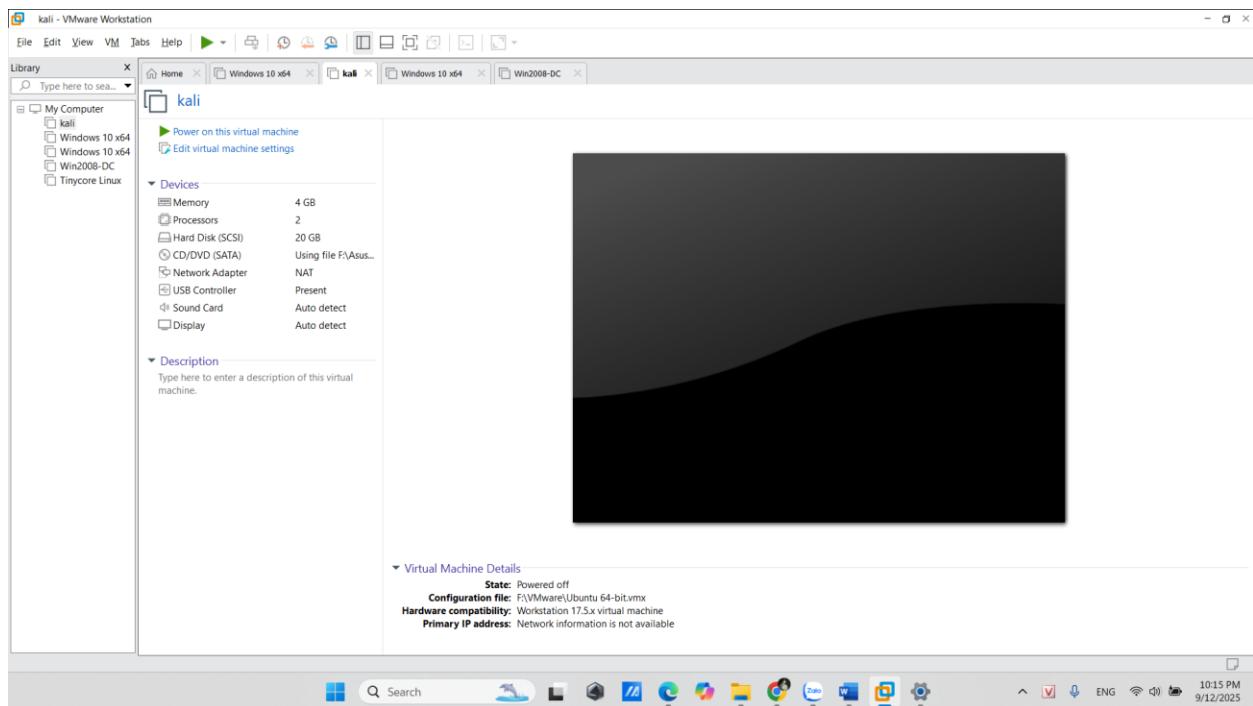
I also download the .iso file of Kali Linux



**Fig 1.2: Kali Linux**

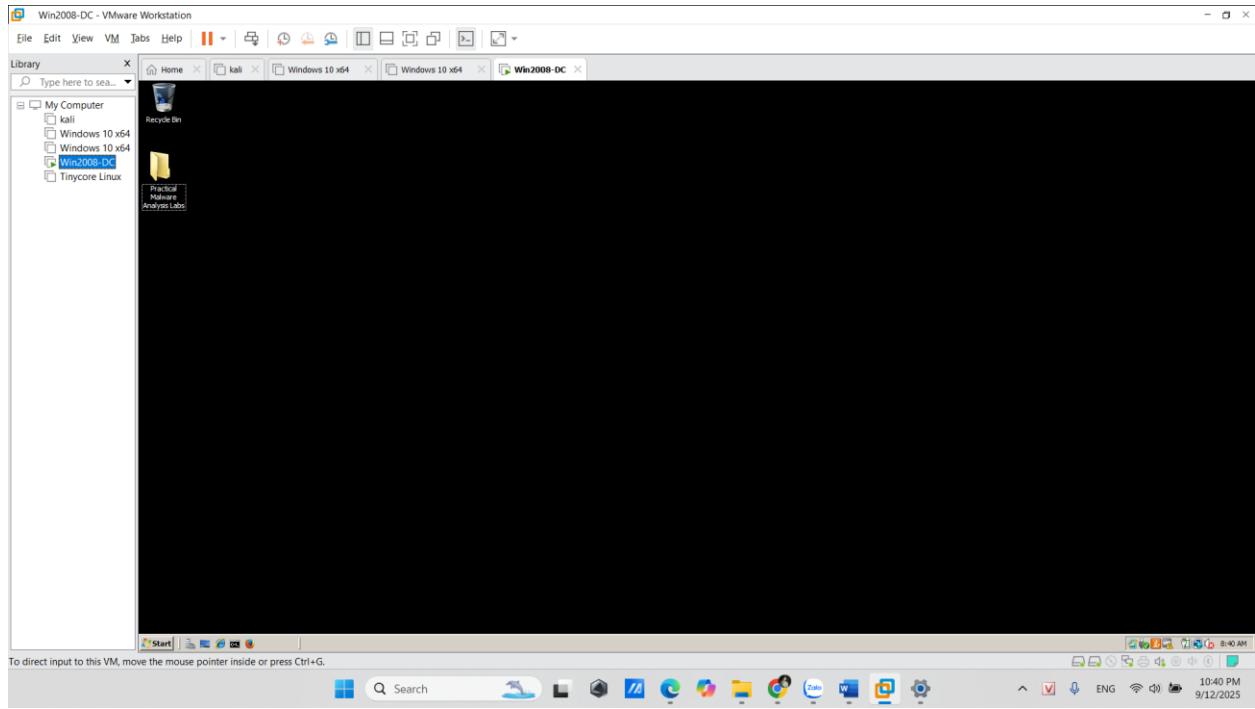
Next, I import two virtual machine into VMware Workstation





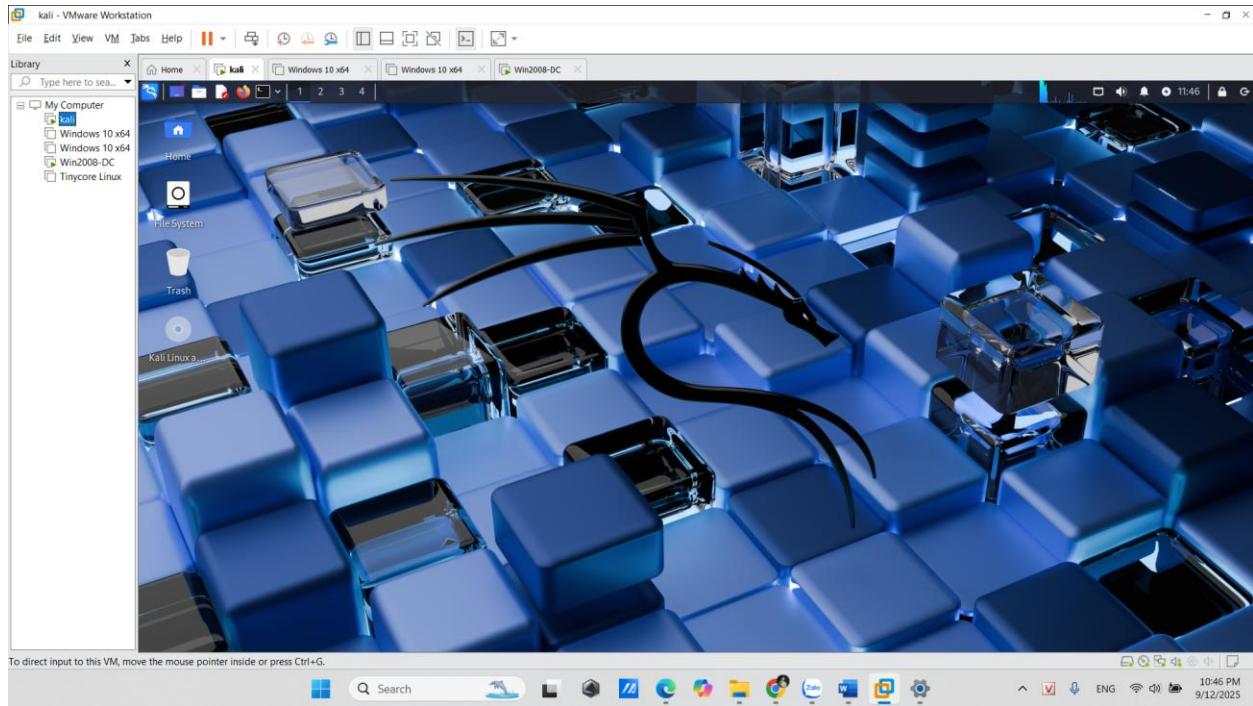
**Fig 1.3, 1.4: Setting virtual machine in VMware**

## Log in Windows Server 2008 (target), log in as **Administrator**



**Fig 1.5: Window Server login screen**

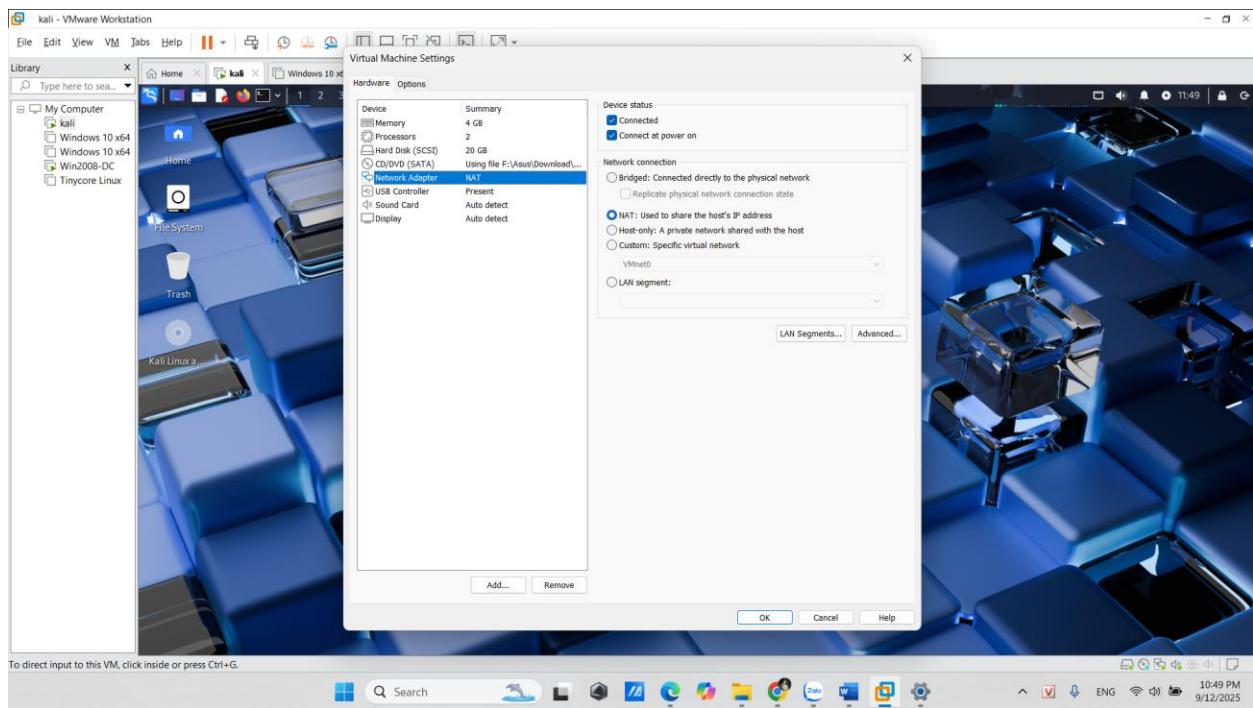
Log in Kali with the username **root** and a password of **toor**



**Fig 1.6: Kali Linux login screen**

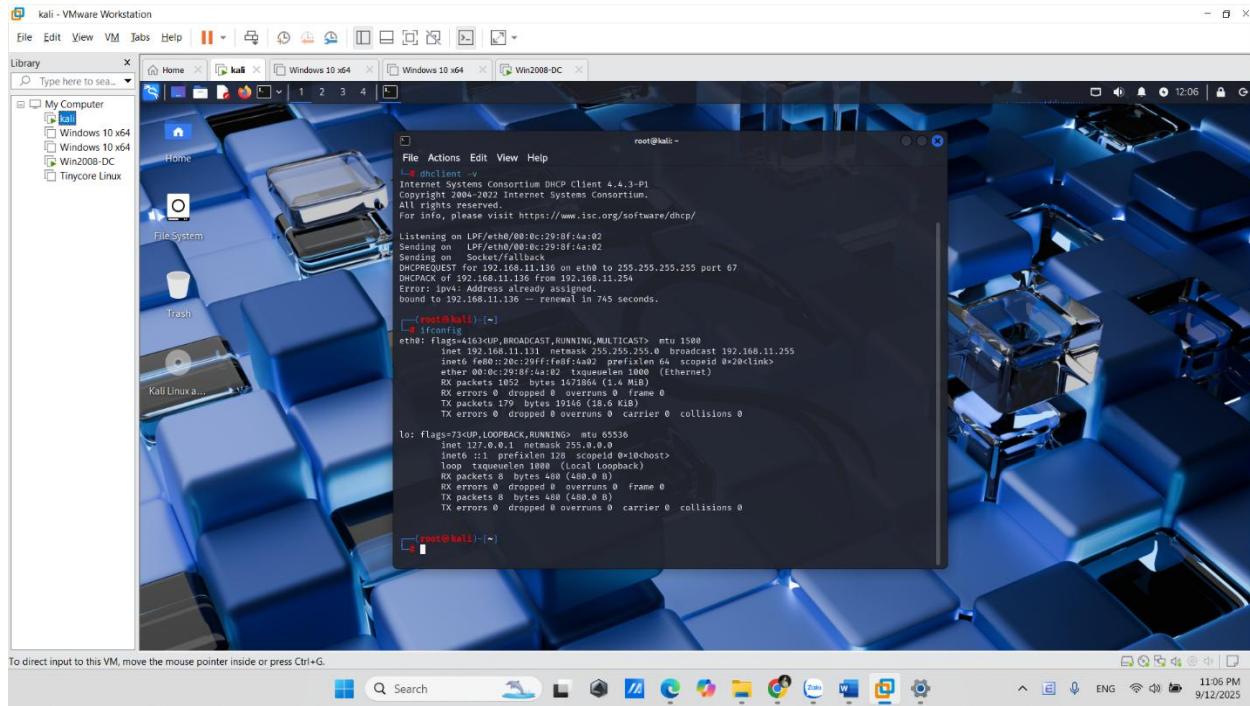
## 2. Setting the Kali Linux VM to NAT Networking

In the setting, change the **Network Adapter** of Kali into **NAT**, that let the VM can share their own specific IP Address with host machine.



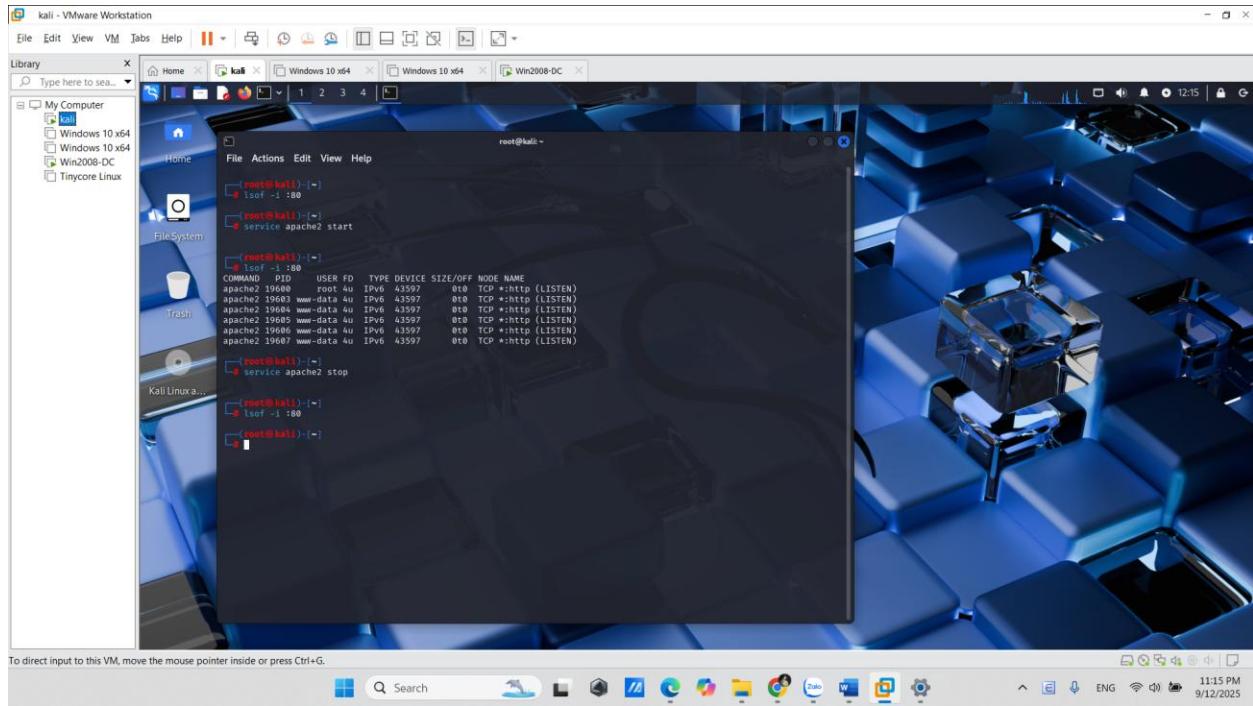
**Fig 2.1: Virtual Machine Settings – Network Adapter Tab**

### 3. Finding the Kali Machine’s IP Address



**Fig 3.1: Kali Linux run dhcp on each adapter**

## 4. Checking for a Web server and Configuring INetSim



**Fig 4.1: List of process that using port 80**

After starting apache2 for web server, we stop them and move to next step

First, backup config file: **cp /etc/inetsim/inetsim.conf /etc/inetsim/inetsim.conf.orig**

And then open config: **nano /etc/inetsim/inetsim.conf**

**Note:** Edit your IP address to your local IP you've got from previous step

```

root@kali: ~
GNU nano 8.3
/etc/inetsim/inetsim.conf *

start_service ftps
start_service tftp
start_service irc
start_service nntp
start_service finger
start_service ident
start_service auth_log
start_service time_tcp
start_service time_udp
start_service daytime_tcp
start_service daytime_udp
start_service echo_tcp
start_service echo_udp
start_service discard_tcp
start_service discard_udp
start_service quotd_tcp
start_service quotd_udp
start_service chargen_tcp
start_service chargen_udp
start_service dummy_tcp
start_service dummy_udp

#####
service_bind_address 0.0.0.0
#
# IP address to bind services to
#
# Syntax: service_bind_address <IP address>
#
# Default: 127.0.0.1
#
#service_bind_address 10.10.10.1

#####
#
# service_run_as_user
#
# User to run services
#
# Syntax: service_run_as_user <username>
#
# Default: inetsim
#
# Help      Write Out   Where Is   Cut Paste   Execute Justify   Location Go To Line   Undo   Set Mark   To Bracket   Where Was   Previous   Back Forward   Prev Word   Next Word
# Exit     Read File   Replace
# Search
# Library
# My Computer
# Windows 10 x64
# Windows 10 x64
# Win2008-DC
# Tinycore Linux
File Actions Edit View Help
root@kali: ~
GNU nano 8.3
/etc/inetsim/inetsim.conf *

dns_bind_port
#
# Port number to bind DNS service to
#
# Syntax: dns_bind_port <port number>
#
# Default: 53
#
#dns_bind_port 53

#####
dns_default_ip 192.168.11.131
#
# Default IP address to return with DNS replies
#
# Syntax: dns_default_ip <IP address>
#
# Default: 127.0.0.1
#
#dns_default_ip 10.10.10.1

#####
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
#
# Syntax: dns_default_domainname <domain name>
#
# Default: inetsim.org
#
#dns_default_domainname

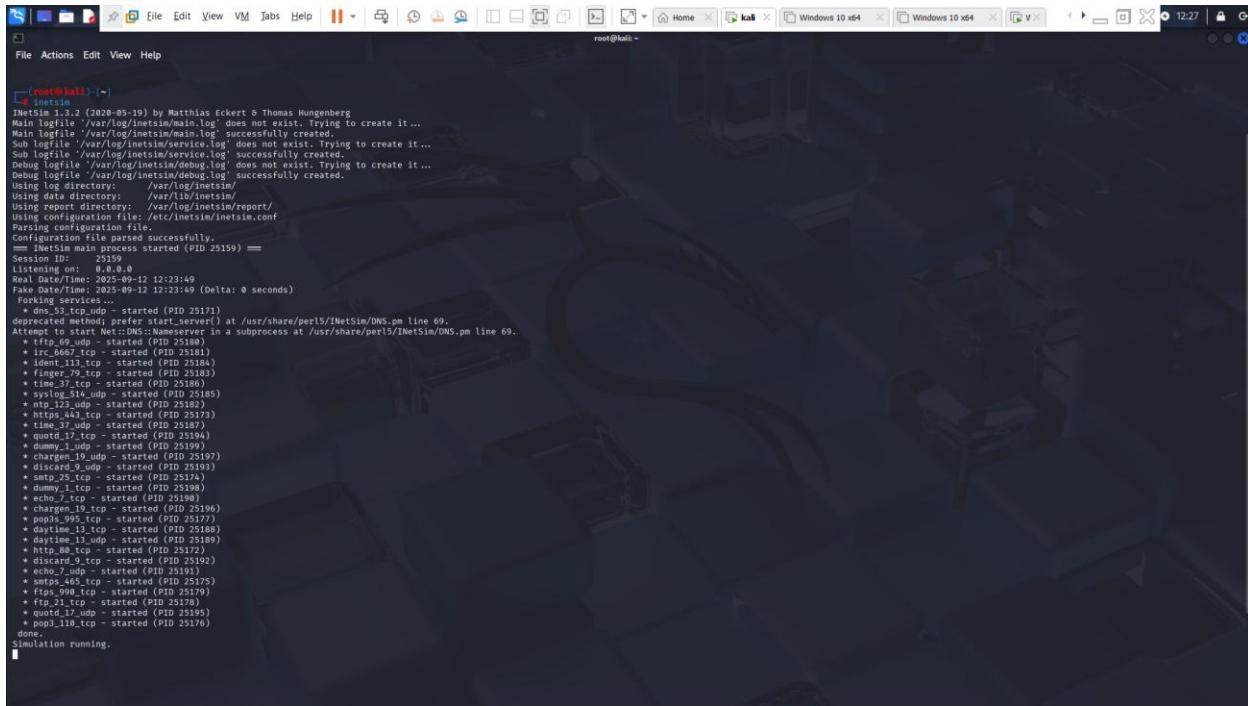
#
# Help      Write Out   Where Is   Cut Paste   Execute Justify   Location Go To Line   Undo   Set Mark   To Bracket   Where Was   Previous   Back Forward   Prev Word   Next Word
# Exit     Read File   Replace
# Search
# Library
# My Computer
# Windows 10 x64
# Windows 10 x64
# Win2008-DC
# Tinycore Linux
File Actions Edit View Help

```

To direct input to this VM, move the mouse pointer inside or press Ctrl+G.

**Fig 4.2: inetsim.conf**

## Safe it and then execute inetsim



```
(root@kali:)-[~]
InetSim 1.1.2 (2020-05-19) by Matthias Eckert & Thomas Hungenberg
Main logfile '/var/log/inetsim/main.log' does not exist. Trying to create it ...
Main logfile '/var/log/inetsim/main.log' successfully created.
Sub logfile '/var/log/inetsim/service.log' does not exist. Trying to create it ...
Sub logfile '/var/log/inetsim/debug.log' successfully created.
Debug logfile '/var/log/inetsim/debug.log' does not exist. Trying to create it ...
Debug logfile '/var/log/inetsim/debug.log' successfully created.
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
Reading configuration file...
Configuration File parsed successfully.
== InetSim main process started (PID 25159) ==
Session ID: 25159
Little Endian: 0x0
Real Date/Time: 2025-09-12 12:23:49
Fake Date/Time: 2025-09-12 12:23:49 (Delta: 0 seconds)
Forwarding service started
+ dgram_53_tcp_udp - started (PID 25171)
deprecated method; prefer_start_server() at /usr/share/perl5/iNetSim/DNS.pm line 69.
Attempt to start Net::DNS::Nameserver in a subprocess at /usr/share/perl5/iNetSim/DNS.pm line 69.
+ dgram_67_tcp - started (PID 25181)
+ irc_69_tcp - started (PID 25181)
+ ident_113_tcp - started (PID 25184)
+ finger_79_tcp - started (PID 25183)
+ time_37_tcp - started (PID 25180)
+ simon_514_udp - started (PID 25185)
+ http_123_udp - started (PID 25182)
+ https_443_tcp - started (PID 25173)
+ telnet_23_tcp - started (PID 25177)
+ quoted_17_tcp - started (PID 25194)
+ dummy_1_udp - started (PID 25199)
+ chargen_19_udp - started (PID 25197)
+ echo_20_tcp - started (PID 25193)
+ snat_25_tcp - started (PID 25174)
+ dummy_1_tcp - started (PID 25198)
+ echo_7_tcp - started (PID 25190)
+ rfc1394_1024_udp - started (PID 25196)
+ pop3_995_tcp - started (PID 25177)
+ daytime_13_tcp - started (PID 25188)
+ daytyme_13_udp - started (PID 25189)
+ telnet_520_tcp - started (PID 25171)
+ discard_9_tcp - started (PID 25192)
+ echo_7_udp - started (PID 25191)
+ smtp_465_tcp - started (PID 25175)
+ smtp_25_tcp - started (PID 25179)
+ ftp_21_tcp - started (PID 25178)
+ quoted_17_udp - started (PID 25195)
+ pop3_110_tcp - started (PID 25176)
done
Simulation running.
```

Fig 4.3: inetsim running

When running INetSim, an error occurred due to a missing or outdated **Net::DNS** Perl module. To fix it, we manually downloaded and installed the correct version using the following commands:

```
curl -LO https://www.net-dns.org/download/Net-DNS-1.22.tar.gz
```

```
tar -xvzf Net-DNS-1.22.tar.gz
```

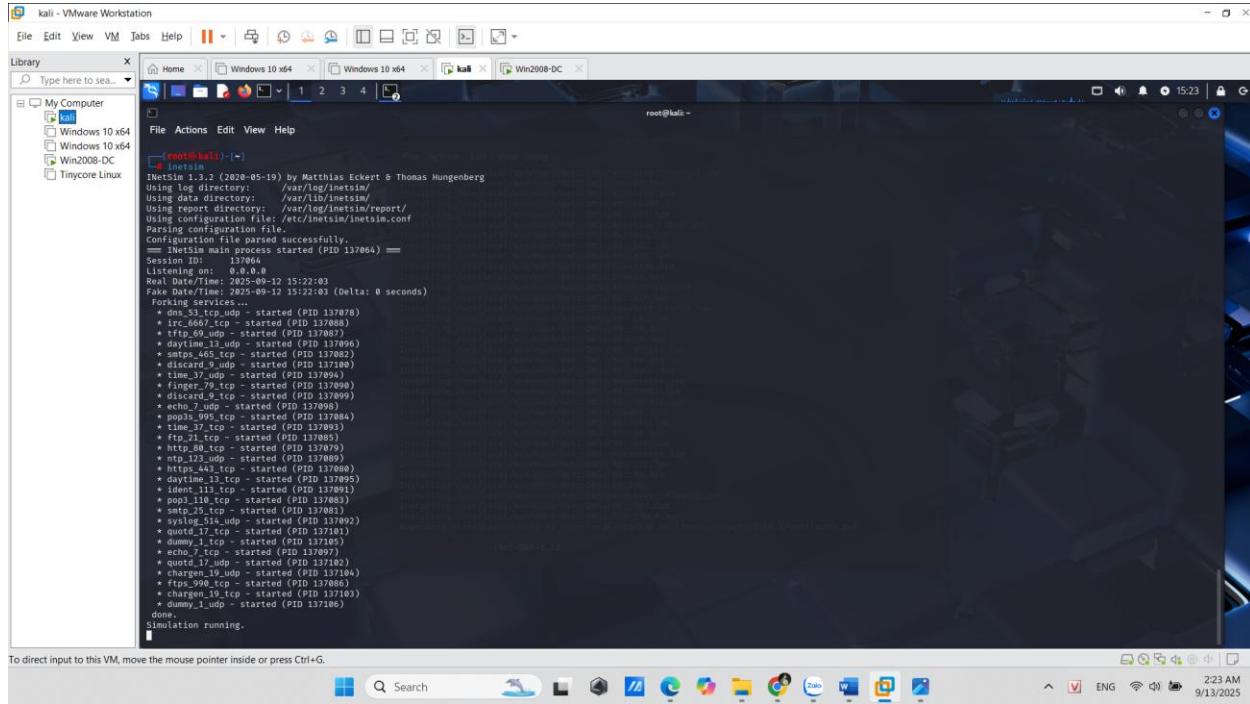
```
cd Net-DNS-1.22
```

```
perl Makefile.PL
```

```
make
```

```
sudo make install
```

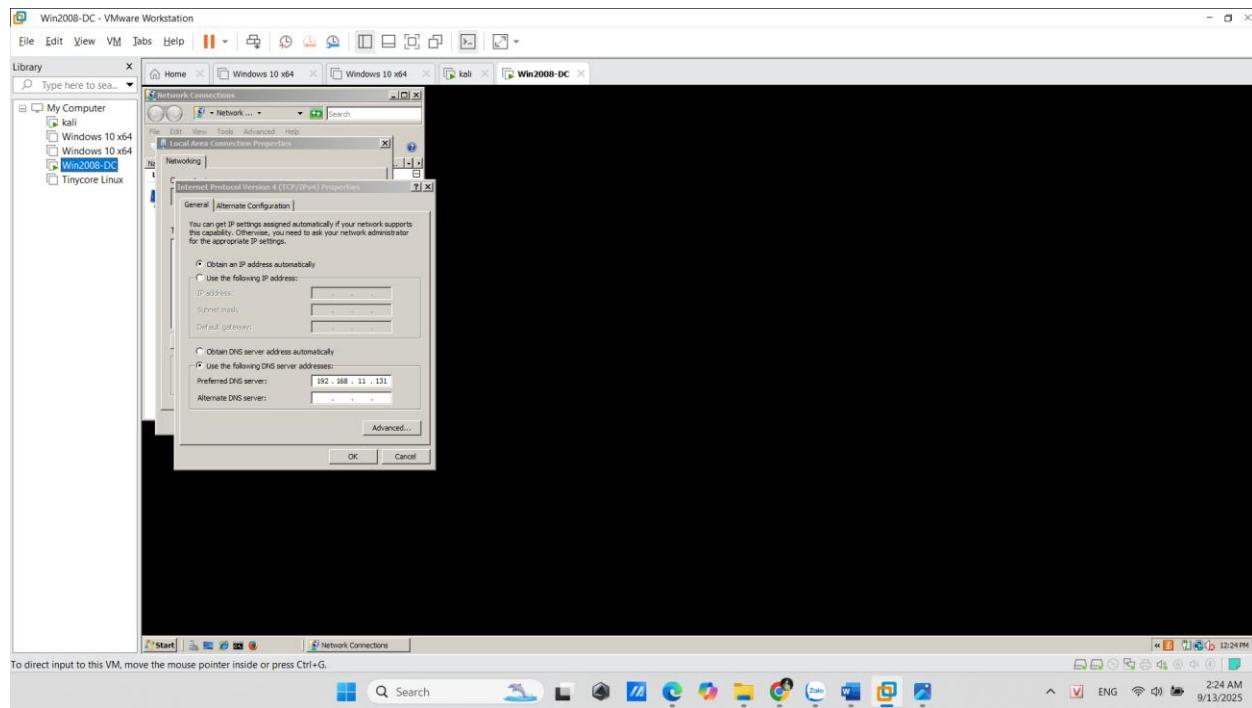
```
perl -MNet::DNS -e 'print "$Net::DNS::VERSION\n"'
```



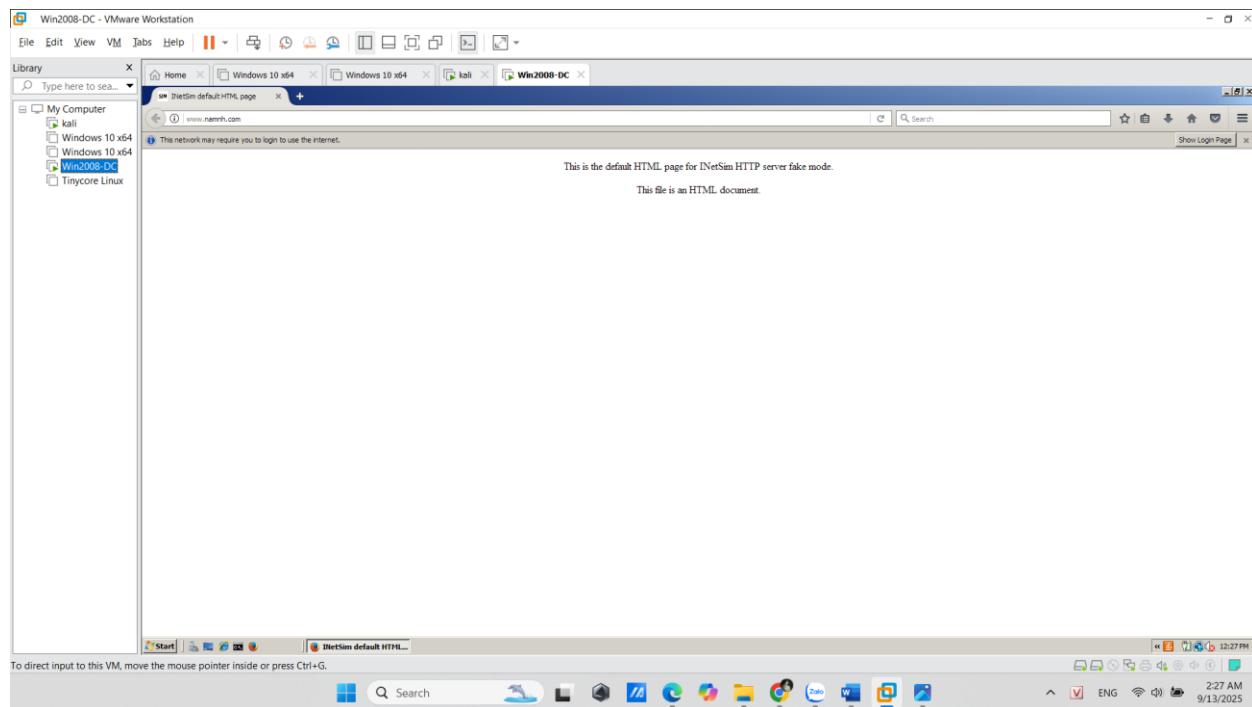
```
root@kali: ~# ./inetSim
InetSim 1.3.2 (2020-05-19) by Matthias Eckert & Thomas Hungenberg
Using log directory: /var/log/inetsim/
Using report directory: /var/log/inetsim/report/
Using configuration file: /etc/inetsim/inetsim.conf
Parsing configuration file...
Configuration file parsed successfully.
== InetSim main process started (PID 137064) ==
Session ID: 137064
Listening on: 0.0.0.0
Real Date/Time: 2025-09-12 15:22:03
Fake Date/Time: 2025-09-12 15:22:03 (Delta: 0 seconds)
Forcing services...
* http_80_udp - started (PID 137078)
* irc_6697_tcp - started (PID 137068)
* tftp_69_udp - started (PID 137087)
* daytime_13_udp - started (PID 137096)
* echo_7_udp - started (PID 137092)
* discard_9_udp - started (PID 137100)
* time_37_udp - started (PID 137094)
* finger_79_tcp - started (PID 137090)
* xmodem_45_tcp - started (PID 137099)
* echo_7_tcp - started (PID 137098)
* pop3_995_tcp - started (PID 137084)
* time_37_tcp - started (PID 137093)
* xmodem_45_udp - started (PID 137095)
* http_80_tcp - started (PID 137079)
* ntp_123_udp - started (PID 137089)
* https_443_tcp - started (PID 137080)
* http_8000_tcp - started (PID 137095)
* ident_113_tcp - started (PID 137091)
* pop3_110_tcp - started (PID 137083)
* xmodem_45_tcp - started (PID 137081)
* syslog_514_udp - started (PID 137092)
* quodt_17_tcp - started (PID 137101)
* dummy_1_tcp - started (PID 137105)
* echo_7_tcp - started (PID 137107)
* xmodem_45_tcp - started (PID 137109)
* ident_17_udp - started (PID 137102)
* chargen_19_udp - started (PID 137104)
* ftps_998_tcp - started (PID 137086)
* chargen_19_tcp - started (PID 137103)
* xmodem_1_udp - started (PID 137106)
done.
Simulation running.
```

After this, it's working perfectly, no more error

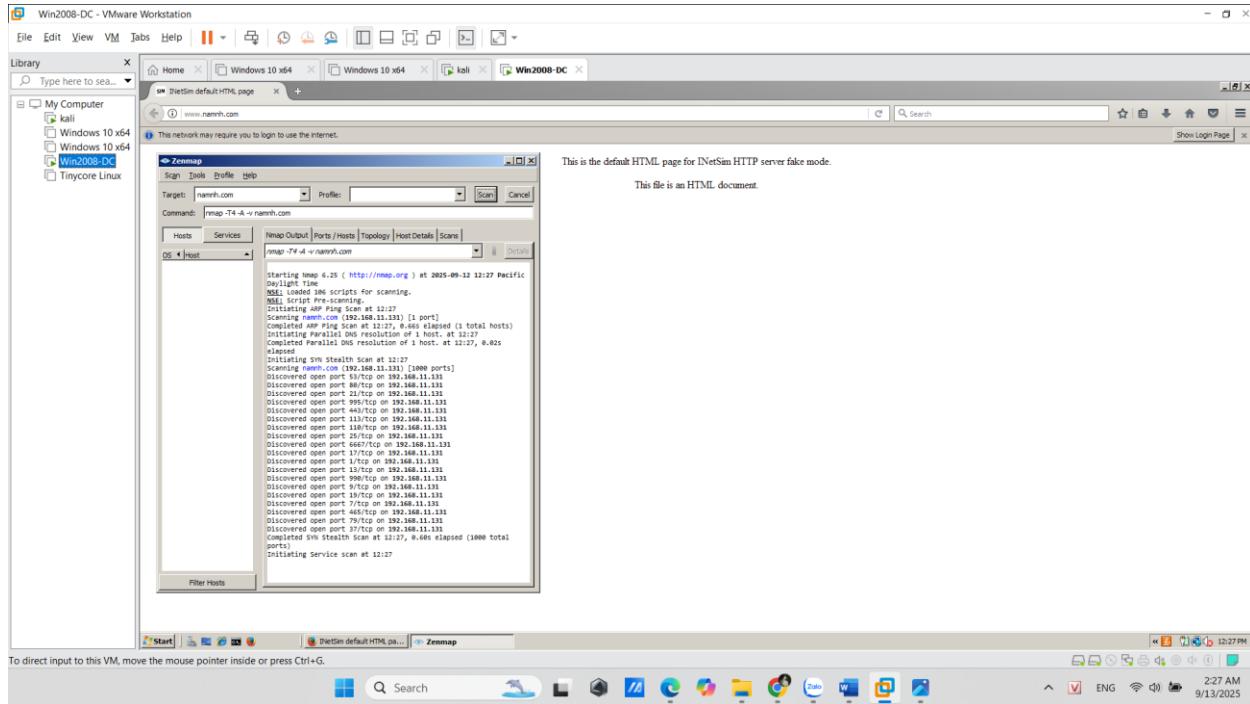
## 5. Network Configuration for Windows VM (DNS)



## Test HTTP on Windows (open YOURNAME.com)



## Scanning YOURNAME.com



## IV. Summary

- This lab focuses solely on configuring a self-hosted DNS responder on Kali Linux.
- All steps proceeded as expected except **Step 3**: on newer Kali releases you must downgrade/install Net::DNS (use Net-DNS-1.22) to resolve compatibility errors so INetSim runs correctly.

## Additional Question

### 1. PE (Portable Executable) Structure of Windows

- PE is a file format for executable files used in the Windows operating system, based on the COFF (Common Object File Format) file format

- A PE file is a data structure that contains the information needed for the operation system loader to load that executable file into memory and execute it
- **Structure:**
  - o **DOS Header:** Every PE file begins with a 64-byte structure, which is what makes the PE file an MS-DOS executable
  - o **DOS Stub:** It's a small MS-DOS 2.0, compatible executable that simply print the error message when the program is run in DOS mode
  - o **NT Header:** consists of three main part:
    - **PE Signature:** A 4-byte signature that identifies the file as a PE file
    - **File Header:** A Standard COFF. It contains some information about the PE files
    - **Optional Header:** The most important header of the NT Header, because some file like object files do not have this header, however, this header is required for image files. Provide important information to the operating system loader
  - o **Section Table:** located right after the Optional Header, is an array of Image Section Header, each section in the PE file has a section header; Each header contains information about the section it references
  - o **Sections:** where the actual contents of the file are stored, including things like data and resources that the program uses, as well as the actual code of the program. There are several sections, each with its own purpose.

## **2. What is the differences between NAT, Host-only and Bridge in VMware Network Adapter?**

- **NAT (Network Address Translation):**
  - o Connect VM to Internet via host, more secure, not directly accessed
  - o Used for personal lab
- **Bridged:**
  - o VM as a device on LAN
  - o Used when needing to access/be accessed from real network
- **Host-only**
  - o Isolate VM from host or between VMs
  - o Used for malware analysis without allowing access to the Internet