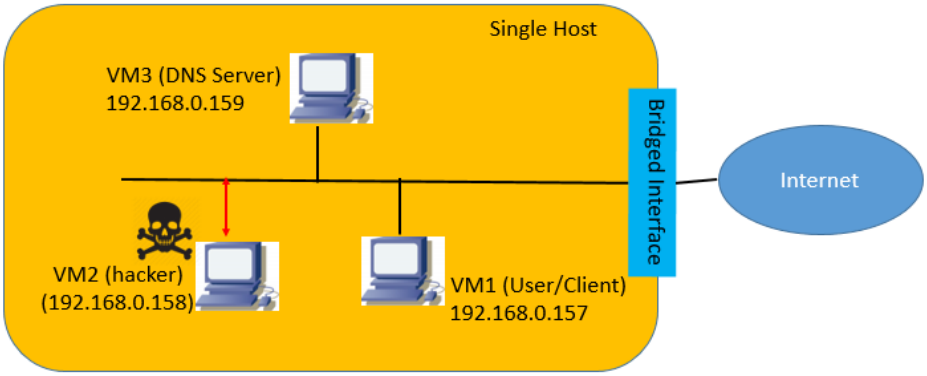
Lab06: DNS Attack (Local)

1. Learning Goals

- Learn to configure DNS server on Linux system
- Learn to use the **netwag** tool to launch a spoofing attack on DNS

2. Lab Environments



Note that the instruction is based on the IP addresses of the above diagram, and students should modify the instruction based on their own IP addresses.

The subnet of the lab instruction is 192.168.0 and your subnet would be different.

3. Lab Procedure of DNS Attack

3.1 Task 1: DNS Configuration and Test

Step 1: On VM3, download the DNS server package.

sudo apt-get install bind9

Step 2: On VM3, edit the file **named.conf.options** located at /etc/bind

```
options {
        directory "/var/cache/bind";
        dump-file "/var/cache/bind/dump.db";   // adding this line for the SEED DNS lab
```

Also in the same file, turn off DNSSEC

```
//------------------------------
# dnssec-validation auto;
# dnssec-enable yes;
dnssec-enable no;
```

Step 3: On VM3, edit the file **named.conf.local** located at /etc/bind. It is to create the DNS zone.

```
zone "example.com" {
        type master;
        file "/var/cache/bind/example.com.db";
};

zone "0.168.192.in-addr.arpa" {
        type master;
        file "/var/cache/bind/192.168.0";
};
```

Step 4: On VM3, create the file **example.com.db** at /var/cache/bind

```
[VM3] pwd
/var/cache/bind
[VM3] cat example.com.db
$TTL 3D
@       IN      SOA     ns.example.com. admin.example.com. (
        2018041601      ; Serial
        8H              ; Refresh
        2H              ; Retry
        4W              ; Expire
        1D)             ; Negative Cache TTL
;
@       IN      NS      ns.example.com.
@       IN      MX      10 mail.example.com.
;
www     IN      A       192.168.0.201
mail    IN      A       192.168.0.202
ns      IN      A       192.168.0.210
*.example.com.  IN      A       192.168.0.200
```

Step 5: on VM3, create another file **192.168.0** at /var/cache/bind

```
[VM3] pwd
/var/cache/bind
[VM3] cat 192.168.0
$TTL 3D

@       IN      SOA     ns.example.com. admin.example.com. (
        2018041601
        8H
        2H
        4W
        1D)
;
@       IN      NS      ns.example.com.

201     IN      PTR     www.example.com.
202     IN      PTR     mail.example.com.
210     IN      PTR     ns.example.com.
```

Step 6: On VM3, check the DNS status and then start (or restart) the service.

```
[VM3] sudo /etc/init.d/bind9 status
 * bind9 is running
[VM3] sudo /etc/init.d/bind9 restart
 * Stopping domain name service... bind9
 * Starting domain name service... bind9
```

Note: check /var/log/syslog to see if there is any message in loading DNS database.

[VM3] tail /var/log/syslog

Step 7. One VM1 (DNS client), edit the file **resolv.conf** at /etc to set the new DNS server.[1]

[1] The entries in the /etc/resolv.conf could be reset by the DHCP server. Therefore, we need to turn off DNS server in the DHCP setting. It is also recommended to add the DNS entry in /etc/resolvconf/resolv.conf.d/base

```
nameserver 192.168.0.159
```

Step 8. On VM1 (DNS client), set the DNS server

System Setting  then network  then the [option] button then the [ipv4 Settings] tab.

DHCP Address Only

Manually set DNS server

Step 9: On VM1 (DNS client), use the **dig** command to run DNS Test.

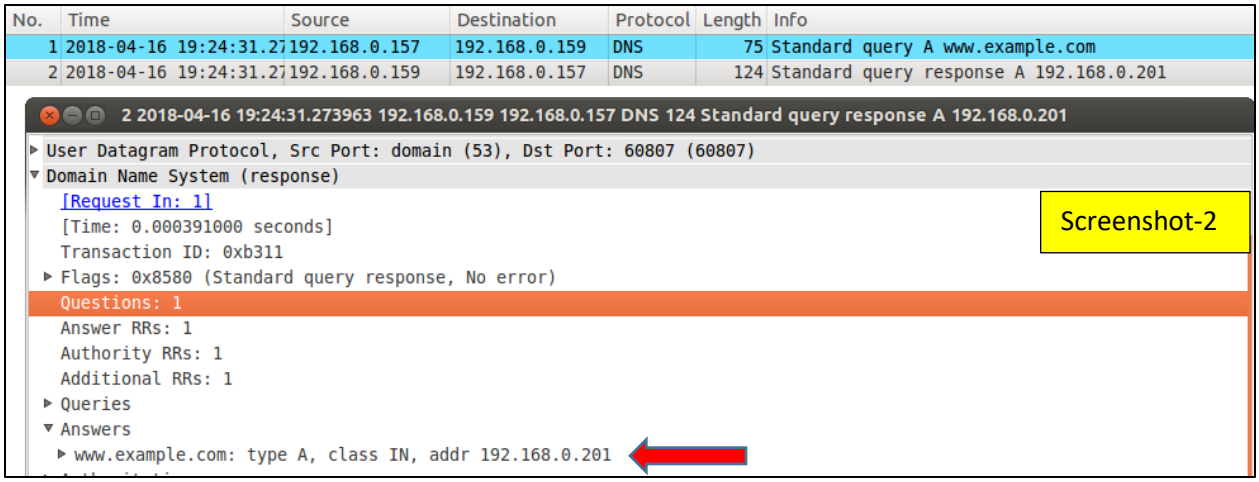
Screenshot-1

```
[VM1] dig www.example.com

; <<>> DiG 9.8.1-P1 <<>> www.example.com
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 12615
;; flags: qr aa rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 1, ADDITIONAL: 1

;; QUESTION SECTION:
;www.example.com.               IN      A

;; ANSWER SECTION:
www.example.com.        259200  IN      A       192.168.0.201

;; AUTHORITY SECTION:
example.com.            259200  IN      NS      ns.example.com.

;; ADDITIONAL SECTION:
ns.example.com.         259200  IN      A       192.168.0.210

;; Query time: 1 msec
;; SERVER: 192.168.0.159#53(192.168.0.159)
;; WHEN: Tue Jun 12 10:12:13 2018
;; MSG SIZE  rcvd: 82
```

The ping command can also be used to check the DNS query.

```
[VM1] ping www.example.com
PING www.example.com (192.168.0.201) 56(84) bytes of data.
```

Step 10: On VM1 (DNS client), use wireshark to capture the DNS traffic to and form the DNS server.



3.2 Task 2: DNS Attack on Local /etc/hosts File

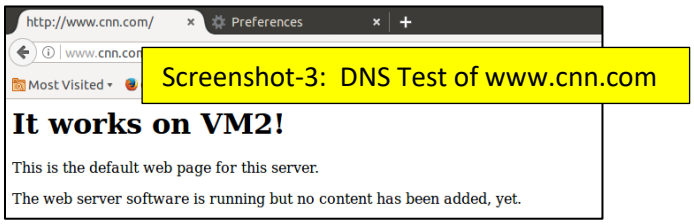
Step 1: On VM1, edit the /etc/hosts file by adding the following three entries. Do not change other entries in the file.

```
192.168.0.157    www.cis.syr.edu
192.168.0.158    www.cnn.com
192.168.0.159    www.depaul.edu
```

Step 2: Use the **ping** command to test the DNS service.

```
[VM1] ping -c 2 www.cis.syr.edu
PING www.cis.syr.edu (192.168.0.157) 56(84) bytes of data.
64 bytes from www.cis.syr.edu (192.168.0.157): icmp_req=1 ttl=64 time=0.014 ms
64 bytes from www.cis.syr.edu (192.168.0.157): icmp_req=2 ttl=64 time=0.015 ms
```

Step 3: Use the Web to test the DNS service.



Step 4: Remove the entries of step-1 from the /etc/hosts file. Test and confirm the entries are cleaned.

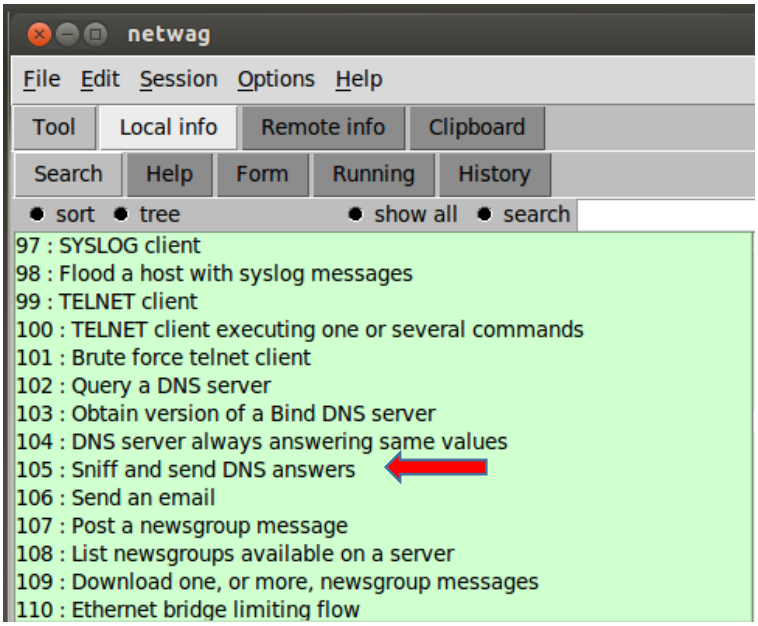
3.3 Task 3. Spoofing the DNS Response

Step 1: On VM2 (hacker), configure the interface in the promiscuous mode.

Step 2: The attacking tool, netwag, should already installed in the SEED image. Confirm and run it.

```
[VM2] which netwag
/usr/bin/netwag
[VM2] sudo netwag
```

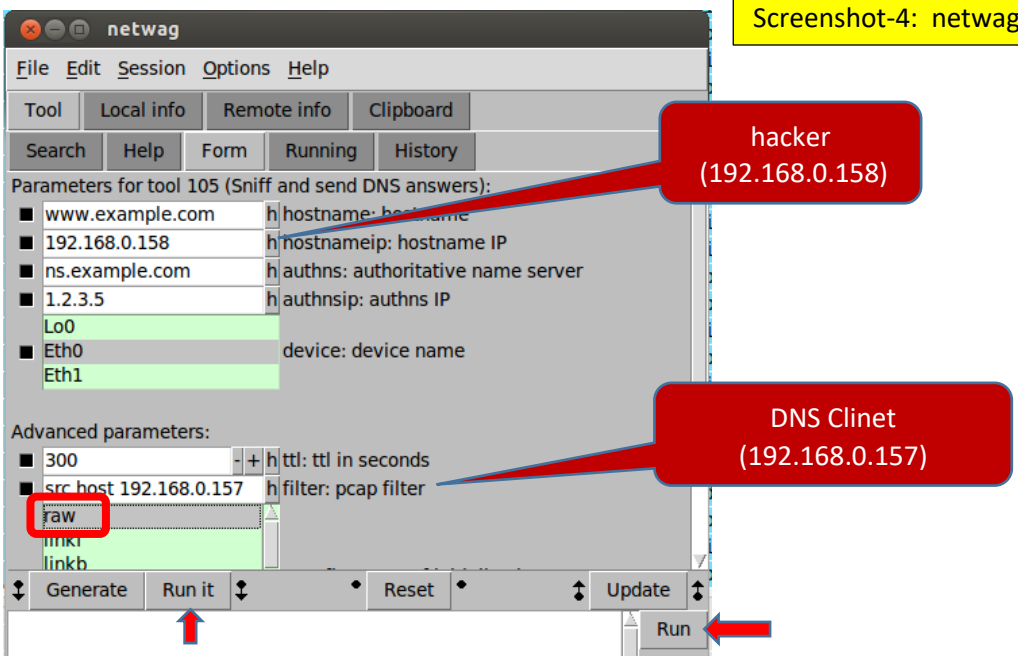
Step 3: the command netwag creates a new window. Scroll down to 105**: Sniff and sends DNS answers.**



Step 4: Configure **netwag** 105 for DNS spoofing attack. The attacking scenario is to change the IP address of hostname=www.example.com from 192.168.0.201 (on the DNS server) to 192.168.0.158 (hacker.) The source IP address is spoofed to the client address (192.168.0.157). Also select **raw** for the spoofed IP packet type.

After the configuration, run it (click the [run] button and then "**run it**" tab.

Screenshot-4: netwag configuration



Step 5: On VM1, run DNS query **multiple times** and check if the queried IP address for www.example.com is changed to the hacker.

```
[VM1] dig www.example.com

; <<>> DiG 9.8.1-P1 <<>> www.example.com
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 11217
;; flags: qr aa rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 1, ADDITIONAL: 1

;; QUESTION SECTION:
;www.example.com.               IN      A

;; ANSWER SECTION:
www.example.com.        300     IN      A       192.168.0.158

;; AUTHORITY SECTION:
ns.example.com.         300     IN      NS      ns.example.com.

;; ADDITIONAL SECTION:
ns.example.com.         300     IN      A       1.2.3.5

;; Query time: 4 msec
;; SERVER: 192.168.0.159#53(192.168.0.159)
;; WHEN: Tue Jun 12 10:46:23 2018
;; MSG SIZE  rcvd: 88
```

Screenshot-5: proof of DNS hacking (client)

Step 6: On VM1, start wireshark and observe the captured DNS traffic. Note that for each DNS query, there are two DNS responses. Also note that the source IP address from VM2 is spoofed.

Screenshot-6: Hacked DNS answer (client)

	Time	Source	Destination	Protocol	Length	Info
1	2018-06-12 11:46:37.08	192.168.0.157	192.168.0.159	DNS	75	Standard query A www.example.com
2	2018-06-12 11:46:37.08	192.168.0.159	192.168.0.157	DNS	130	Standard query response A 192.168.0.158
3	2018-06-12 11:46:37.08	192.168.0.159	192.168.0.157	DNS	124	Standard query response A 192.168.0.201

3.4 Task 4. DNS Server Cache Poisoning

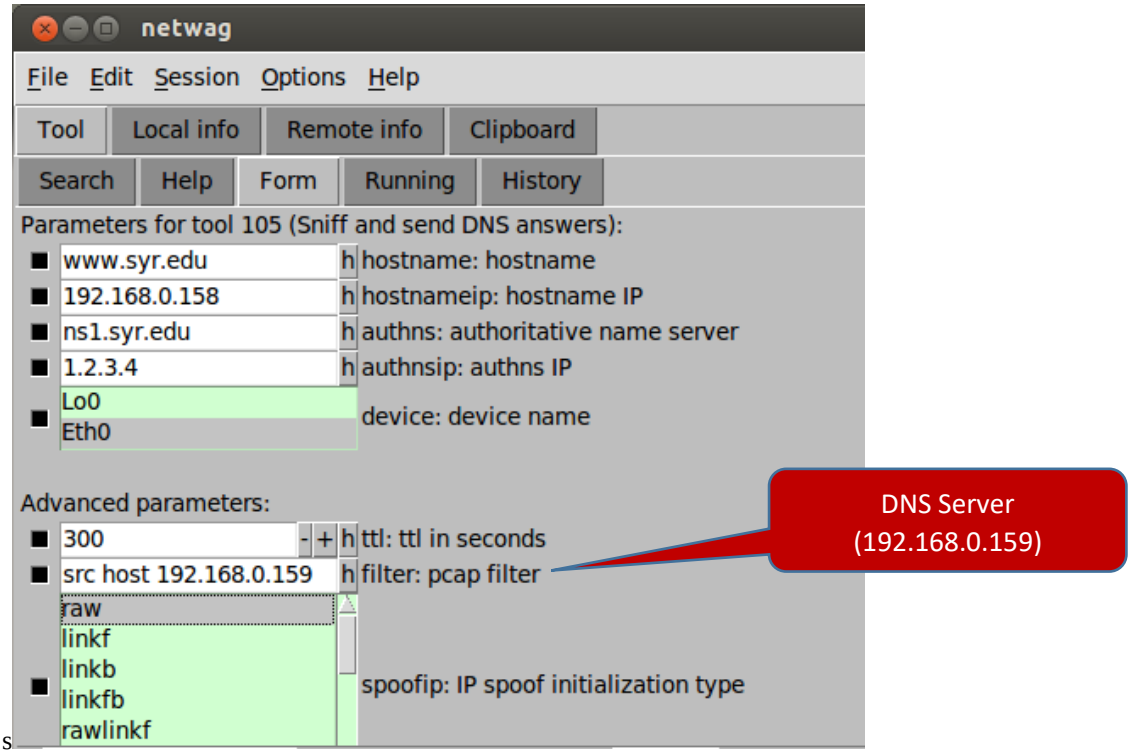
The lab procedure of Task 4 is similar to Task 3. The difference is to spoof the DNS response to the DNS server instead of to the DNS client.

Step 1: On VM3 (DNS server), clean the DNS cache.

```
[VM3] which rndc
/usr/sbin/rndc
[VM3] sudo rndc flush
```

Step 2: On VM3, use **dig** to find the authoritative name server of www.syr.edu. During my test, it is **ns1.syr.edu** and it is different from the published SEED lab manual.

Step 3: On VM2 (hacker), start netwag configuration (105) as Task 3.



Step 4: On VM1, run DNS queries multiple times to www.syr.edu.

```
[host1] dig www.syr.edu

; <<>> DiG 9.8.1-P1 <<>> www.syr.edu
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 11467
;; flags: qr rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 2, ADDITIONAL: 0

;; QUESTION SECTION:
;www.syr.edu.                   IN      A

;; ANSWER SECTION:
www.syr.edu.            269     IN      A       192.168.0.158

;; AUTHORITY SECTION:
syr.edu.                171729  IN      NS      ns1.syr.edu.
syr.edu.                171729  IN      NS      ns2.syr.edu.

;; Query time: 5 msec
;; SERVER: 127.0.0.1#53(127.0.0.1)
;; WHEN: Mon Apr 16 20:57:31 2018
;; MSG SIZE  rcvd: 81
```

Screenshot-7: proof of DNS hacking (server)

Step 4: On VM3 (DNS server), start wireshark to capture the DNS traffic. Note that the query response shows the IP address of www.syr.edu is 192.168.0.158.

Screenshot-8: Hacked DNS Response (Server)

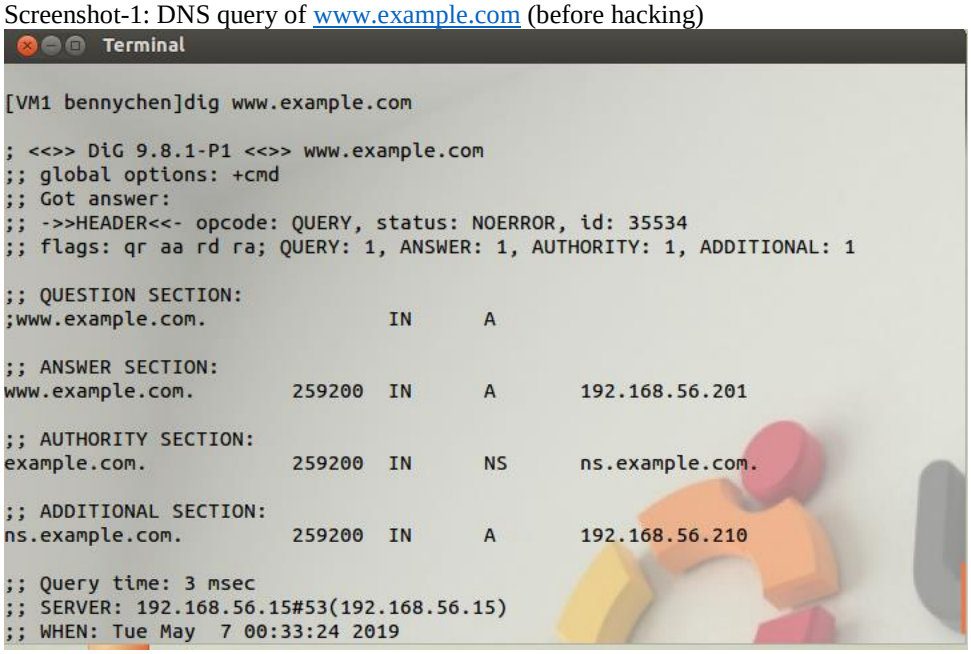
.	Time	Source	Destination	Protocol	Length	Info
1	2018-06-12 12:18:55.42	fe80::a00:27ff:f	2600:1401:2::f0	DNS	110	Standard query A incoming.telemetry.mozilla.org
2	2018-06-12 12:18:57.99	192.168.0.157	192.168.0.159	DNS	71	Standard query A www.syr.edu
3	2018-06-12 12:18:57.99	192.168.0.159	128.230.12.9	DNS	82	Standard query A www.syr.edu
4	2018-06-12 12:18:57.99	128.230.12.9	192.168.0.159	DNS	129	Standard query response A 192.168.0.158
5	2018-06-12 12:18:57.99	192.168.0.159	192.168.0.157	DNS	123	Standard query response A 192.168.0.158
6	2018-06-12 12:18:58.03	128.230.12.9	192.168.0.159	DNS	112	Standard query response CNAME syr.edu A 128.230.18.198

4. Lab Report

1. Your name
   陳邦元
2. Lab Log:
   - How long did you work on this lab?
     2 hrs
   - Any problems? How did you resolve the problem?
     Netwag dns cache poisoning doesn't work. Problem remain unsolved.
3. VM Host information

	Physical Interface	MAC Address	IP Address
VM host1 (client)	Eth15	08:00:27:61:f1:e0	192.168.56.13
VM host2 (hacker)	Eth16	08:00:27:6d:f0:3f	192.168.56.14
VM host3 (server)	Eth17	08:00:27:d0:a9:76	192.168.56.15

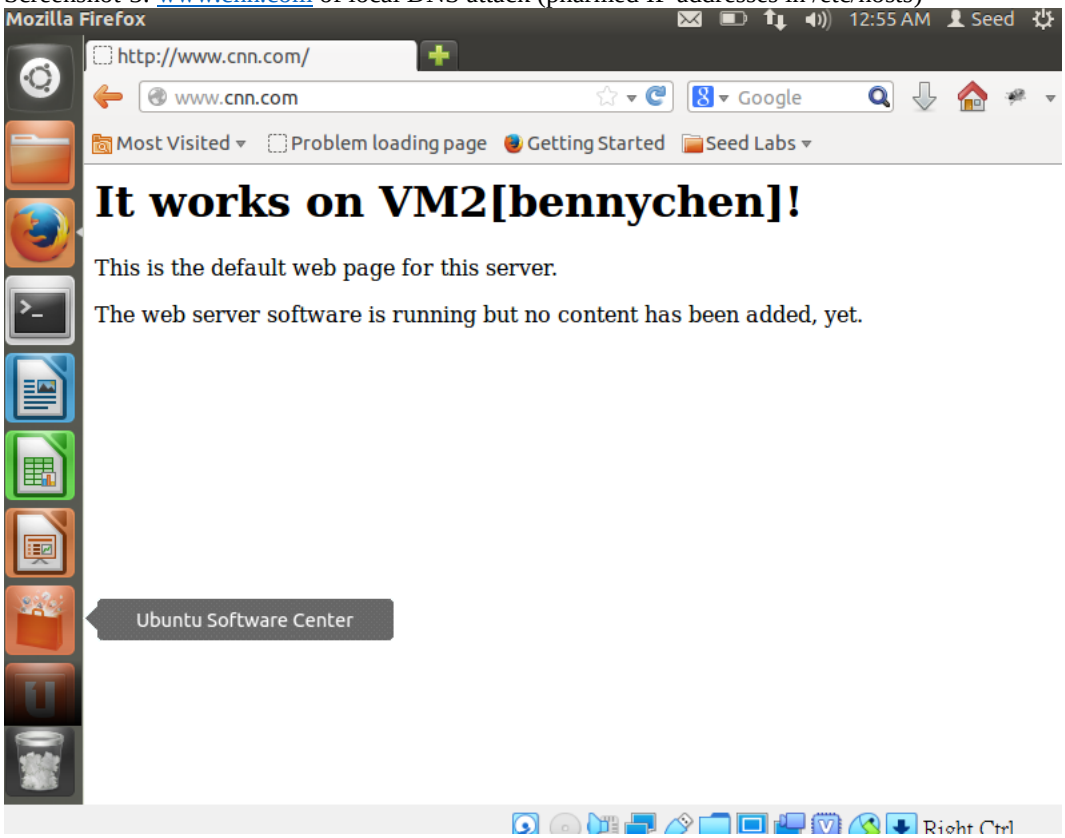
4. Proof of your lab work
   a. Screenshot-1: DNS query of www.example.com (before hacking)



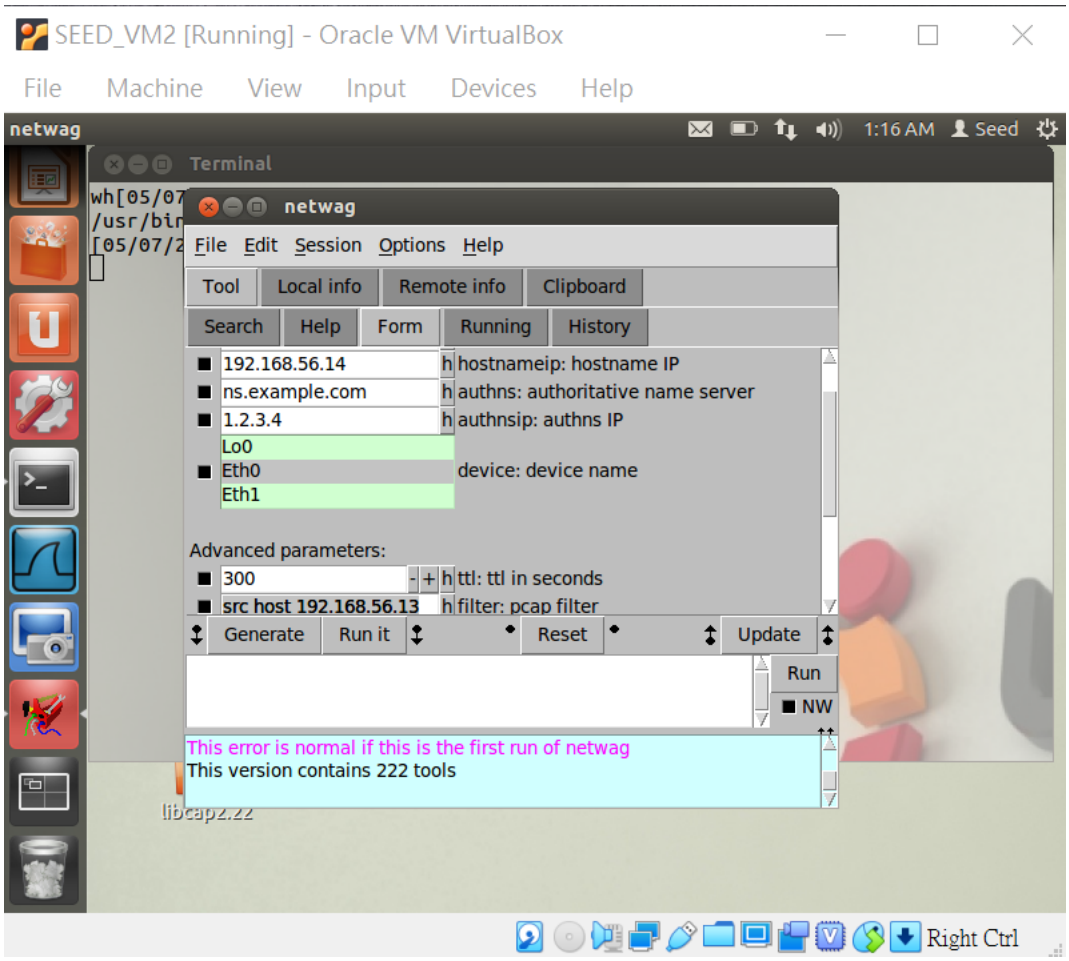
   b. Screenshot-2: wireshark of DNS query for www.example.com (before hacking)

192.168.56.13	192.168.56.15	DNS	75	Standard query A www.example
192.168.56.15	192.168.56.13	DNS	124	Standard query response A 19
CadmusCo_61:f1:e0	CadmusCo_d0:a9:76	ARP	42	Who has 192.168.56.15? Tell
CadmusCo_d0:a9:76	CadmusCo_61:f1:e0	ARP	60	192.168.56.15 is at 08:00:27
192.168.56.13	192.168.56.15	DNS	82	Standard query A videosearch
192.168.56.13	192.168.56.15	DNS	82	Standard query A videosearch
192.168.56.13	192.168.56.15	DNS	111	Standard query A videosearch
192.168.56.13	192.168.56.15	DNS	111	Standard query A videosearch
192.168.56.15	192.168.56.13	DNS	82	Standard query response, Ser

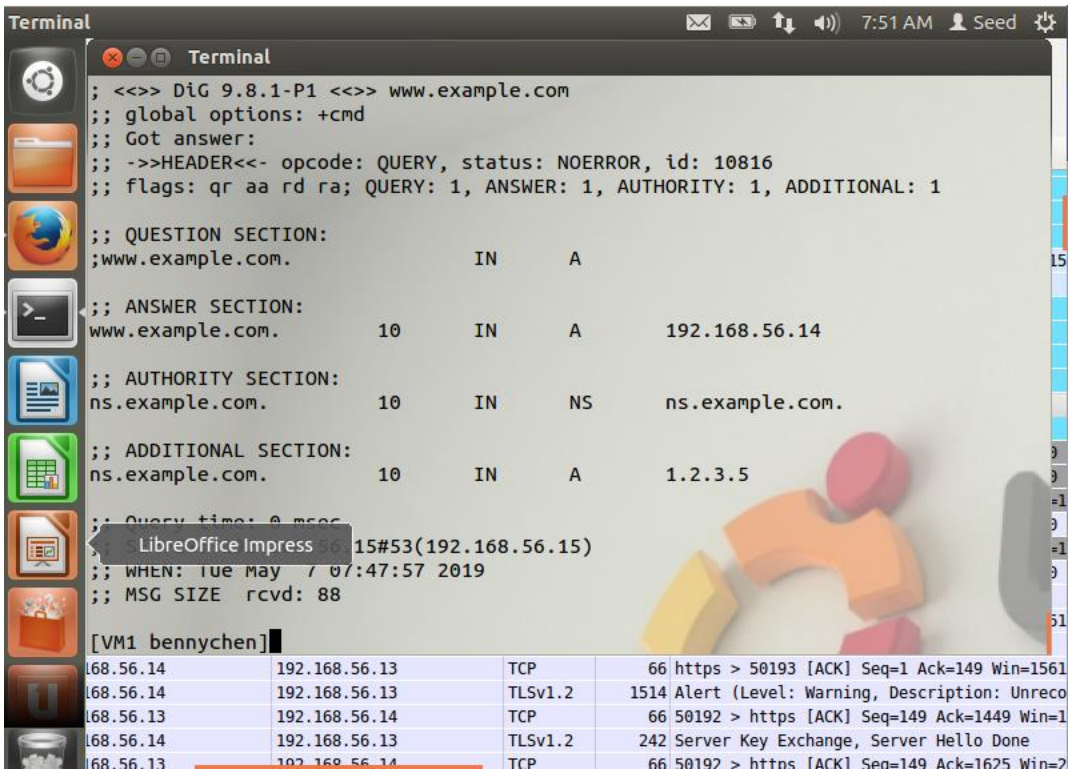
c. Screenshot-3: www.cnn.com of local DNS attack (pharmed IP addresses in /etc/hosts)



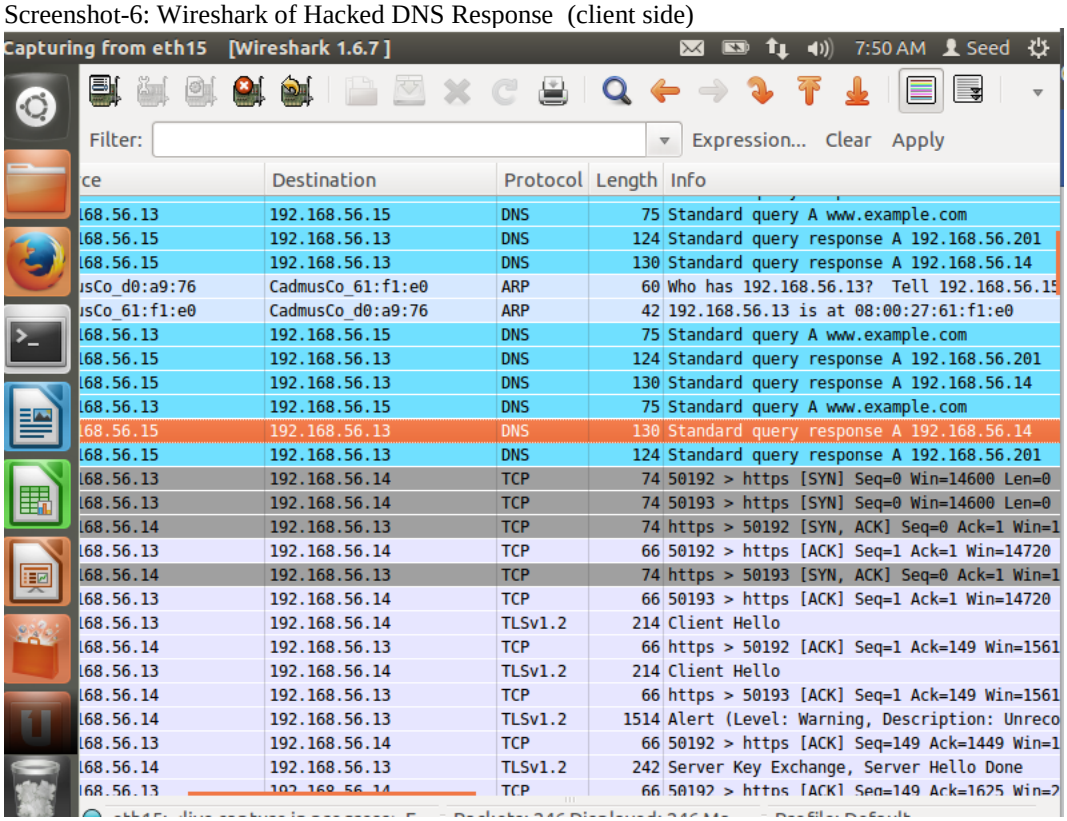
d. Screenshot-4: netwag configuration for DNS Spoofing (client side)



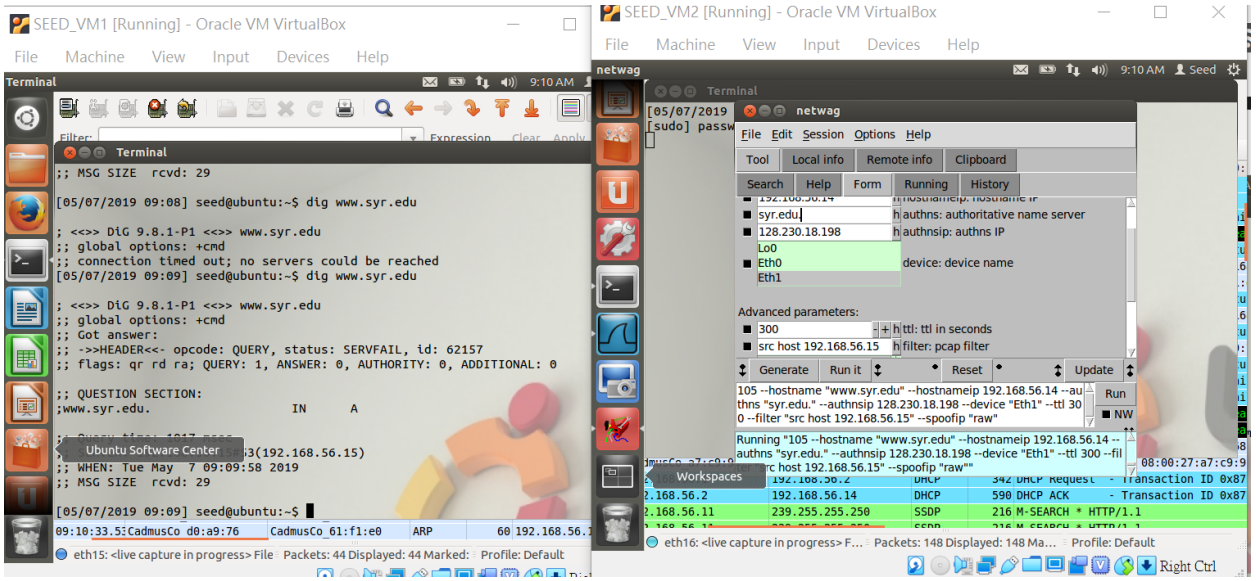
e. Screenshot-5: Proof of DNS hacking (www.exammple.com, client side)



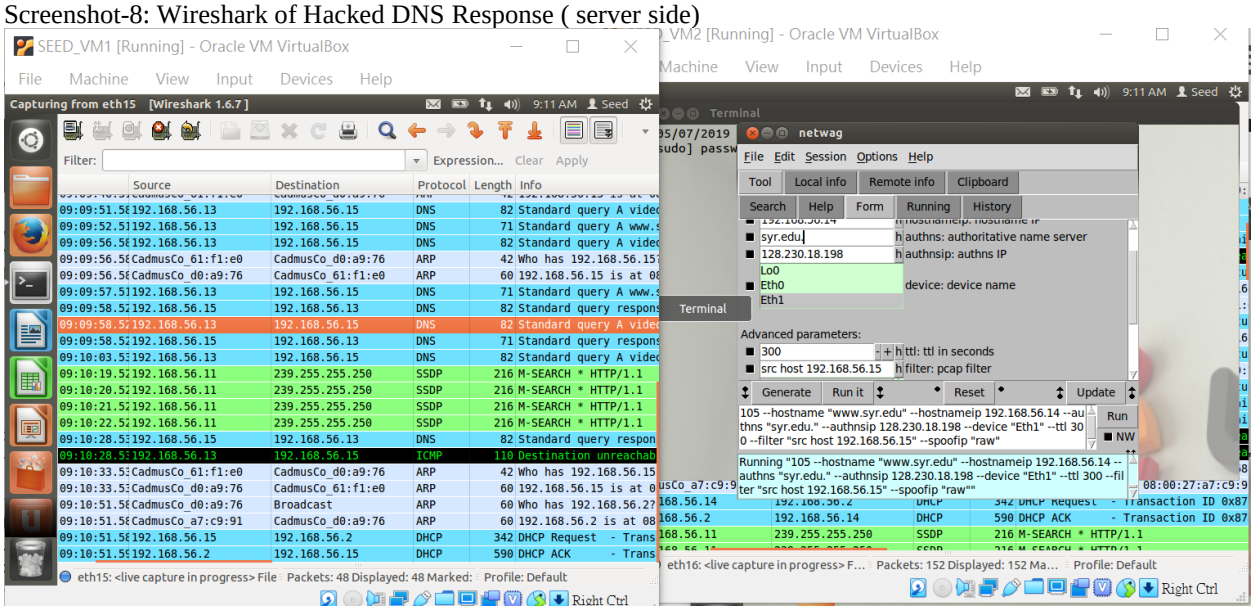
f. Screenshot-6: Wireshark of Hacked DNS Response (client side)



g. Screenshot-7: Proof of DNS hacking (www.syr.edu, server side)



h. Screenshot-8: Wireshark of Hacked DNS Response (server side)



5. Question:
Comparing Task-3 and Task-4, which DNS attack is more effective? Why?
Effectiveness is defined as the percentage of successful attacks.

Task 4 is more effective, cause successful DNS spoofing by directly producing fake response usually depends on the transmission speed difference, and chances are not that high rather than DNS cache poisoning, which means directly spoof the DNS server, and when a client queries, it will definitely gets a spoofed ip.

6. Lab reflection
Describe if the lab learning goals are met and also any interesting observation from this lab exercise.

Barely met the learning goals. Still not be acquainted with netwag. The user interface wasn't friendly.

And still not sure the detail of performing dns cache poisoning.