

Rafid
Shaon
CSC 4222
Nov 07 2022

CSc 4222/6222 Assignment 3

Due: **Nov. 4, 11:59pm**

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 Wi-Fi 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?

There are multiple security risks, and some are malwares/viruses, network spoofing, and session hijacking. With malwares and viruses, attackers are able to install malicious software onto the Wi-Fi and infiltrate all devices connected on the infected Wi-Fi. Attackers are able to monitor every device's activities through a third-party device via network spoofing. With session hijacking, attackers are able to obtain information and data by exploiting a legitimate web browsing session.

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?

There are a total of 18 bytes devoted to head (14 bytes) and footer (4 bytes) information.

Due to the Ethernet frame, the size may increase with the payload/data amount.

With a UDP packet, there are 8 bytes.

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.

Attack: Attacker can send a normal SYN, receiving an ACK-SYN back from the web server. This ACK-SYN has the source address and port with a valid cookie of the attacker's, validating for a minute before allowing the attacker to continue sending packets to the same source address and port. This enables the attacker to maintain millions of open connections from the web server.

Defend: Have an administrator (in this case is you) implement a firewall or detection system in order to detect this DOS attack. This allows the administrator to monitor the network traffic and packet drops while identifying the source of where the traffic load is occurring via an alert. The DOS attacks will be detected through an alarming system.

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 a probability of 99%?

Using the RFC-5452 Formula: $P_s = \frac{D \cdot R \cdot W}{N \cdot P \cdot I}$ based on # of fake requests,

$P_s = 99\%$ or 0.99 (given)

$N = 2.5$

$D = 1$

$P = 1$

$$W = 0.1$$

$$I = 65,536 \text{ (given)}$$

$$0.99 = \frac{1 * R * 0.1}{2.5 * 1 * 65536} \rightarrow 0.99 = \frac{0.1 * R}{163840} \rightarrow R = \frac{0.99 * 163840}{0.1} \rightarrow R = 1622016$$

$$R * \frac{W}{\# \text{ of replies}} \rightarrow 1622016 * \frac{0.1}{2048} = 79.2$$

The attacker needs ~ 79 attempts/requests.

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?

The sequence number will be $156955003 + 1 = \underline{156955004}$.

The acknowledgement number will be $883790339 + 1 = \underline{883790340}$.

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.

A third party attacker needs to be able to find the sequence number of the TCP connection in order to have access to the session. They can then send a packet with one of the sequence numbers and reset with the RST bit at 1.

Case 1: When the third party can sniff the packets from the TCP connections, they will have access to change the IP address. This kills off the session's connection.

Case 2: When the third party cannot sniff the packets from the TCP connections, they will not be able to infiltrate the session. The packet's sequence numbers will need to be determined accurately in order to terminate the session. The third party will not be able to connect to the session, hence closing the connection.

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

Creating a firewall with a tunneling protocol can prevent the IP spoofing on outgoing packets from its internal network. The firewall will identify the source address of the packet entering in from the outside. Through tunneling, the packet itself is encrypted, creating a new header that will be added to the packet. The packet will then be sent to its destination within a virtual private network in order to protect its contents.

8. Suppose you are interested in detecting the number of hosts behind a NAT. You observe that the IP layer stamps an identification number sequentially on each IP packet. The identification number of the first IP packet generated by a host is a large random number, and the identification numbers of the subsequent IP packets are sequentially assigned. Assume all IP packets generated by hosts behind the NAT are sent to the outside world.

a. Based on this observation, and assuming you can sniff all packets sent by the NAT

- to the outside, can you outline a simple technique that detects the number of unique hosts behind a NAT? Justify your answer.
- b. If the identification numbers are not sequentially assigned but randomly assigned, would your technique work? Justify your answer.
 - c. Explain why we say NAT can work as a natural Firewall.
 - a) Yes. A packet sniffer can be used to record all IP packets generated by hosts behind the NAT. Every host will create a sequence of IP packets with numbers in a succession and a distinct initial identification number. These unique IDs are then grouped in a consecutive order, showing the number of hosts behind the NAT.
 - b) The technique would not work if identification numbers are randomly assigned. Since the identification numbers are not grouped in order when randomized, using a packet sniffer to find the group of IP packets with its ID will be ineffective.
 - c) NAT can work as a natural firewall by only allowing what enters in and out of a device through a private network. This allows requests or data packets that are unsolicited to be discarded when attempting to enter the device. The NAT firewall can detect any unrequested traffic without private IP addresses to be discarded.

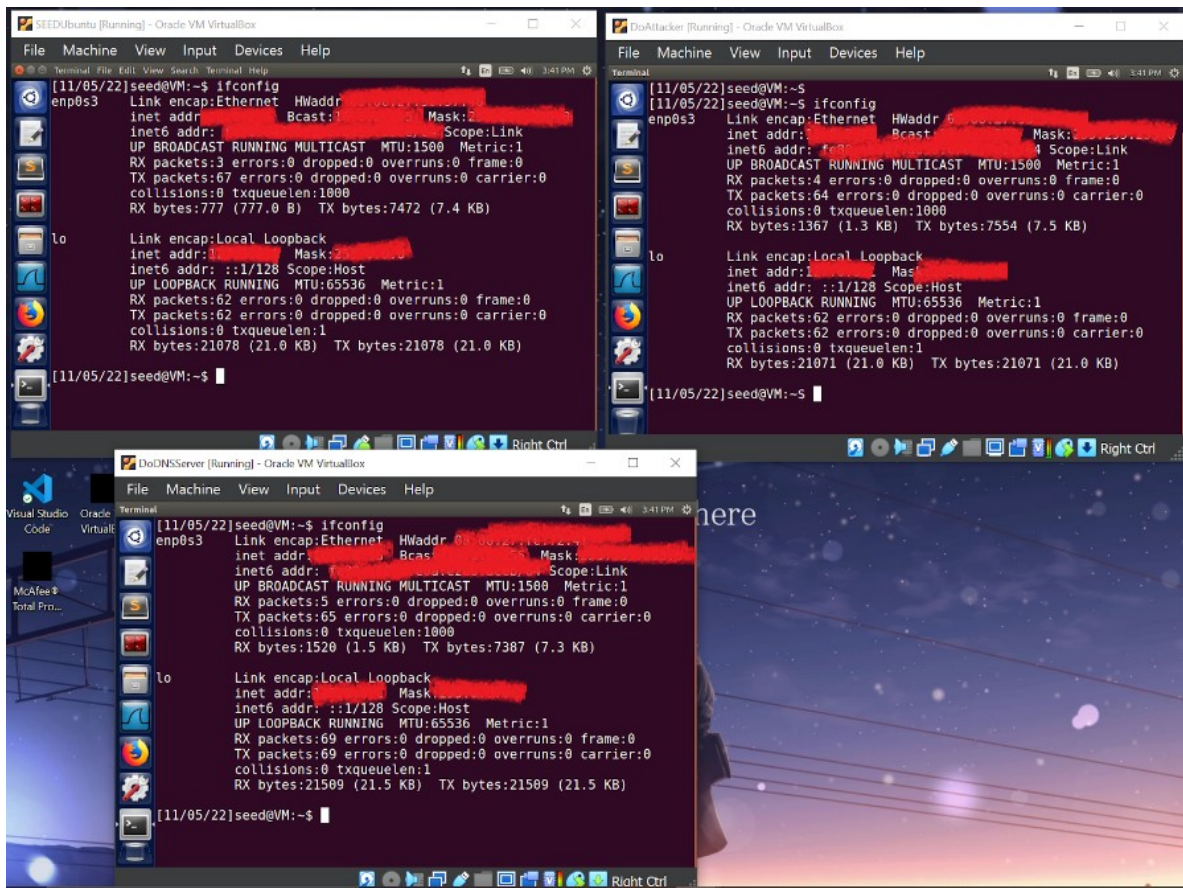
9. Describe the types of rules that would be needed for a rule-based intrusion detection system to detect a smurf attack.

For a rule-based intrusion detection system, signatures are needed to match and detect certain attacks. The rule would encode a signature for every attack in the system. When the IDS manager identifies the smurf attack, it will find a matching pair with its signature and an alarm would immediately sound within the system with its indicated attack type.

Part II: Lab Report

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

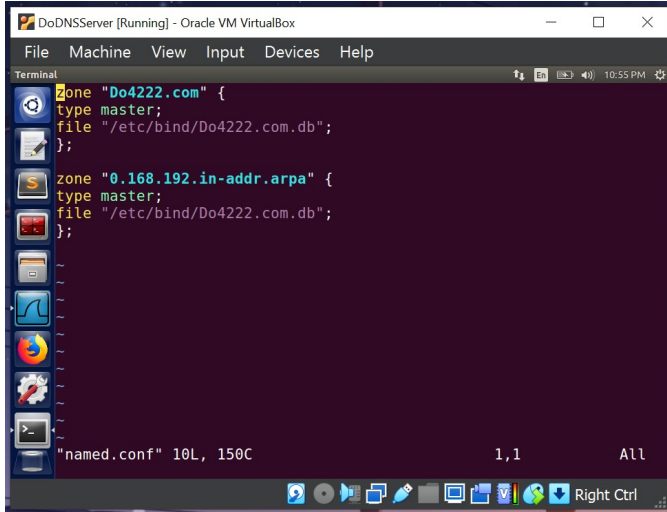
	User VM	Attacker VM	DNS Server VM
IP V4 address	127.X.X.X	127.X.X.X	127.X.X.X
Network Mask	255.XXX.XXX.X	255.XXX.XXX.X	255.XXX.XXX.X
DNS Server	10.X.X.X	10.X.X.X	10.X.X.X



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

	User VM	Attacker VM	DNS Server VM
IP V4 address	127.X.X.X	127.X.X.X	127.X.X.X
Network Mask	255.XXX.XXX.X	255.XXX.XXX.X	255.XXX.XXX.X
DNS Server	10.X.X.X	10.X.X.X	10.X.X.X

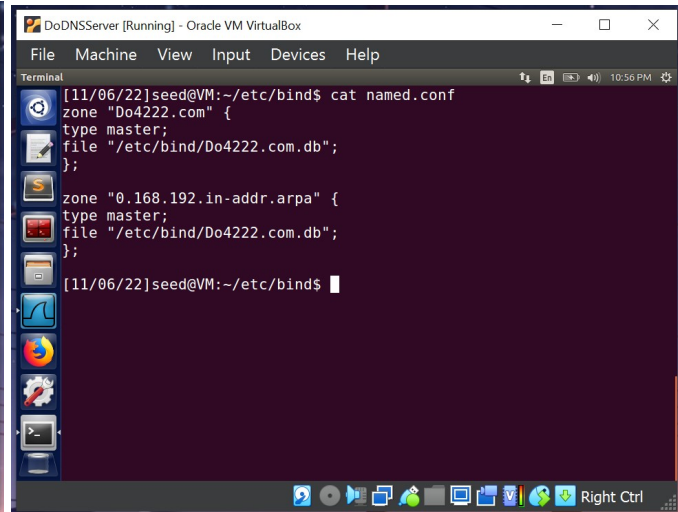
Q3: Show screenshots of your codes in `/etc/bind/named.conf`.



```
DoDNSServer [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
Terminal
zone "Do4222.com" {
type master;
file "/etc/bind/Do4222.com.db";
};

zone "0.168.192.in-addr.arpa" {
type master;
file "/etc/bind/Do4222.com.db";
};

"named.conf" 10L, 150C 1,1 All
```

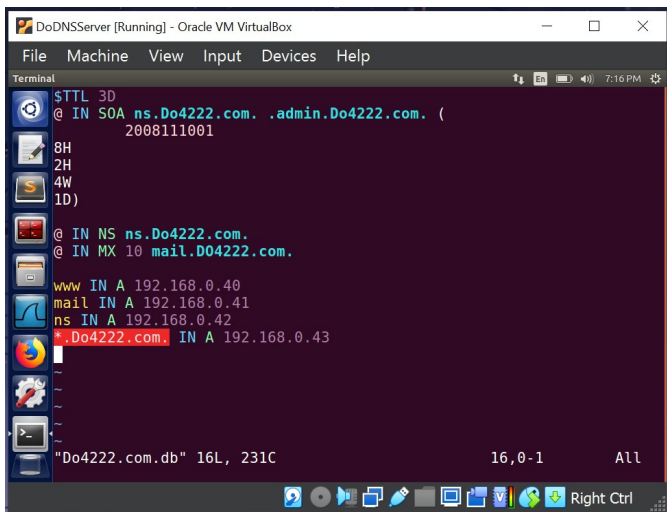


```
DoDNSServer [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
Terminal
[11/06/22]seed@VM:~/etc/bind$ cat named.conf
zone "Do4222.com" {
type master;
file "/etc/bind/Do4222.com.db";
};

zone "0.168.192.in-addr.arpa" {
type master;
file "/etc/bind/Do4222.com.db";
};

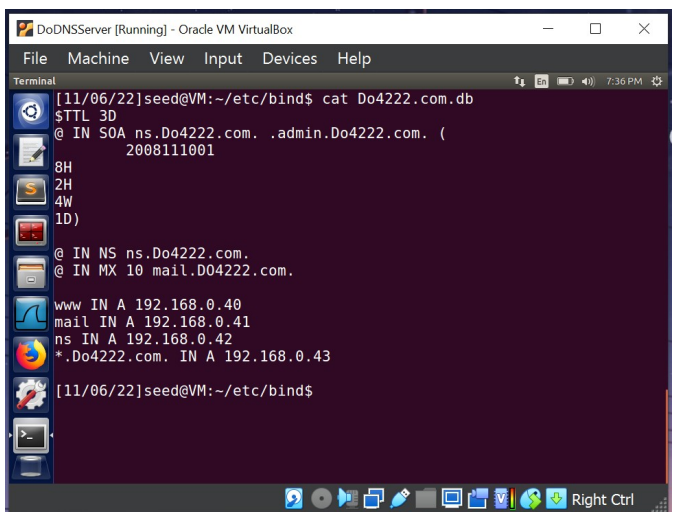
[11/06/22]seed@VM:~/etc/bind$
```

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



```
DoDNSServer [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
Terminal
$TTL 3D
@ IN SOA ns.Do4222.com. .admin.Do4222.com. (
2008111001
8H
2H
4W
1D)
@ IN NS ns.Do4222.com.
@ IN MX 10 mail.Do4222.com.
www IN A 192.168.0.40
mail IN A 192.168.0.41
ns IN A 192.168.0.42
*.Do4222.com. IN A 192.168.0.43

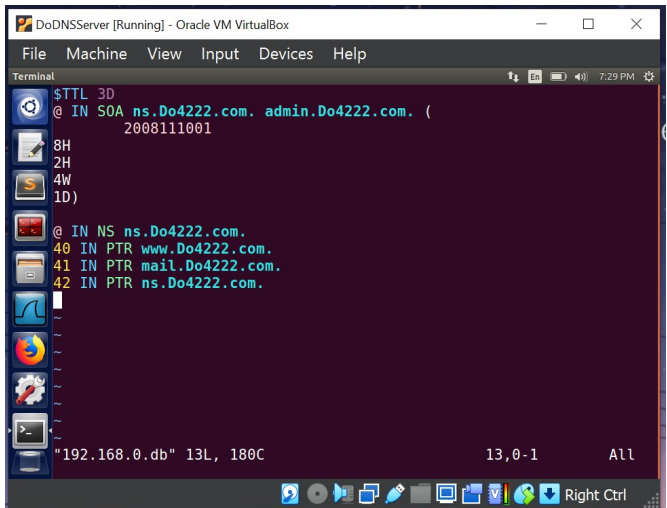
"Do4222.com.db" 16L, 231C 16,0-1 All
```



```
DoDNSServer [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
Terminal
[11/06/22]seed@VM:~/etc/bind$ cat Do4222.com.db
$TTL 3D
@ IN SOA ns.Do4222.com. .admin.Do4222.com. (
2008111001
8H
2H
4W
1D)
@ IN NS ns.Do4222.com.
@ IN MX 10 mail.Do4222.com.
www IN A 192.168.0.40
mail IN A 192.168.0.41
ns IN A 192.168.0.42
*.Do4222.com. IN A 192.168.0.43

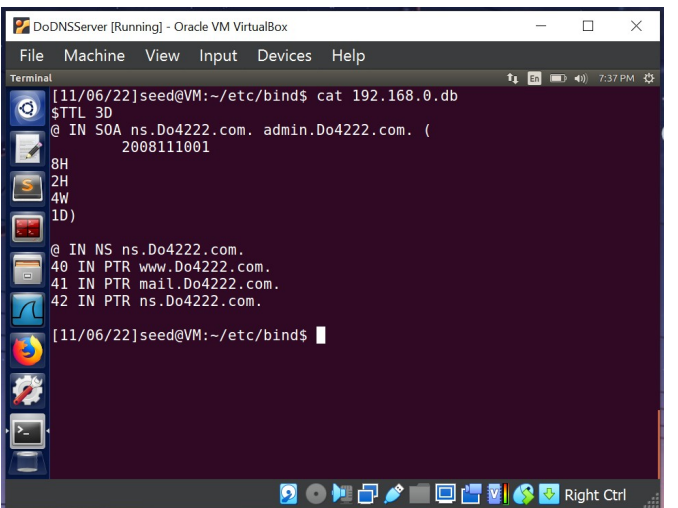
[11/06/22]seed@VM:~/etc/bind$
```

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



```
DoDNSServer [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
Terminal
$TTL 3D
@ IN SOA ns.Do4222.com. admin.Do4222.com. (
2008111001
8H
2H
4W
1D)
@ IN NS ns.Do4222.com.
40 IN PTR www.Do4222.com.
41 IN PTR mail.Do4222.com.
42 IN PTR ns.Do4222.com.

"192.168.0.db" 13L, 180C 13,0-1 All
```

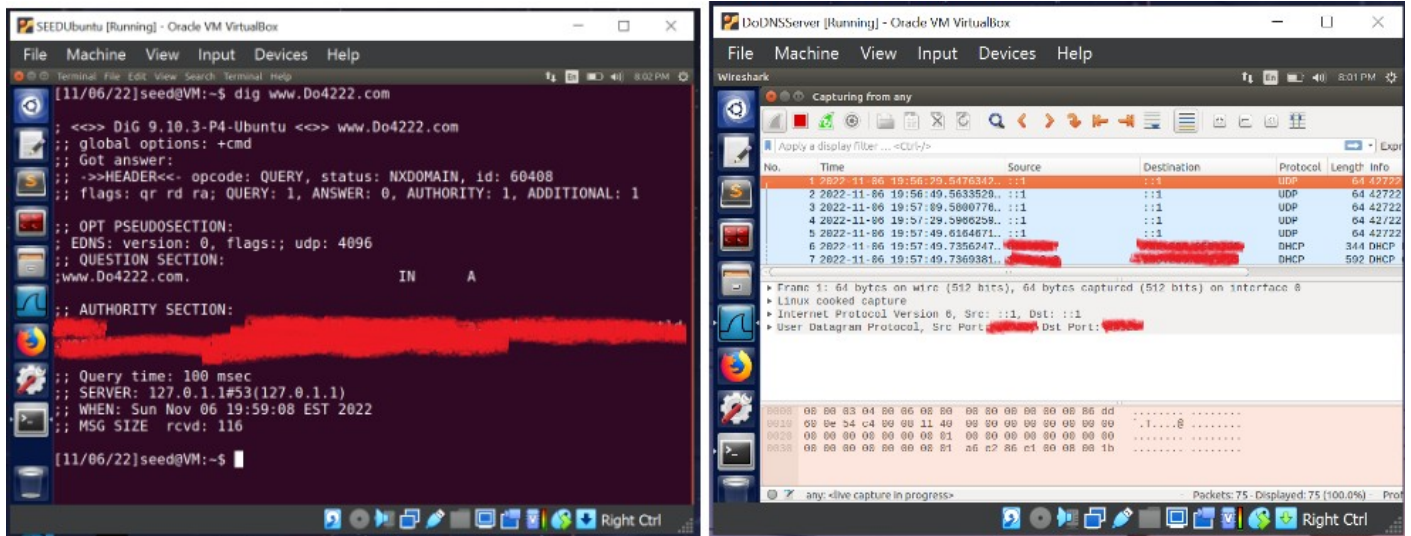


```
DoDNSServer [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
Terminal
[11/06/22]seed@VM:~/etc/bind$ cat 192.168.0.db
$TTL 3D
@ IN SOA ns.Do4222.com. admin.Do4222.com. (
2008111001
8H
2H
4W
1D)
@ IN NS ns.Do4222.com.
40 IN PTR www.Do4222.com.
41 IN PTR mail.Do4222.com.
42 IN PTR ns.Do4222.com.

[11/06/22]seed@VM:~/etc/bind$
```

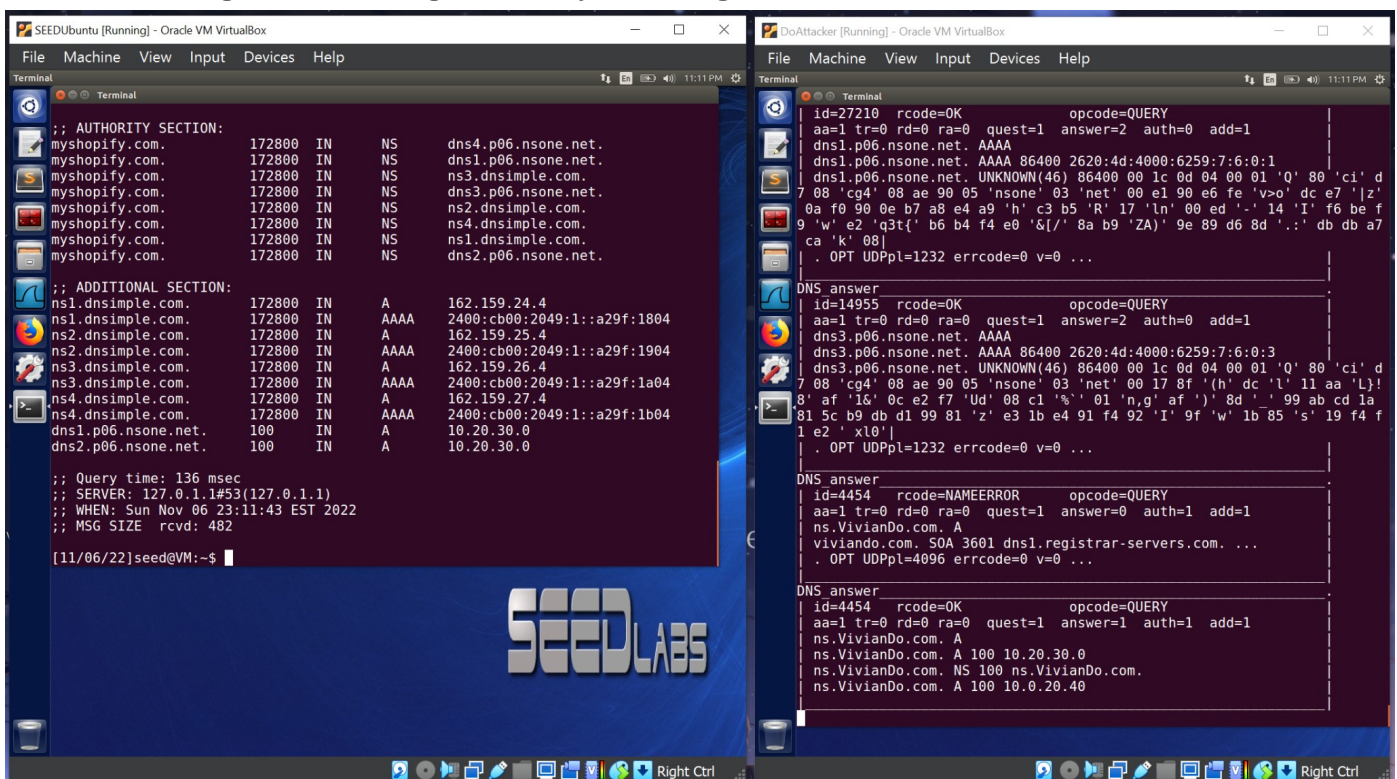
Q6: Show two Screenshots that you successfully set up the server and can dig the www.XXXX4222.com by using your User VM. One is from the User VM terminal after

running the dig command. The other screenshot should show the packets captured by the Wireshark on the DNSServer.



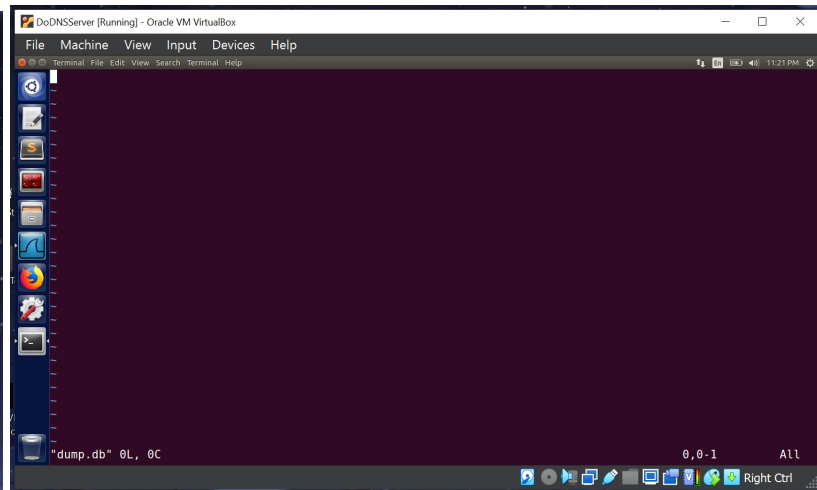
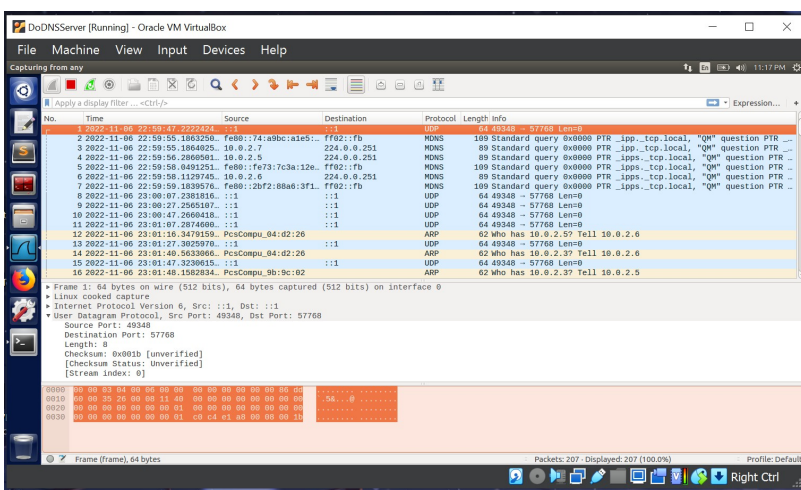
Q7: Explain what happens and how the Netwox 105 tool works to send a response to a user with the wrong IP address.

On the User VM, I was able to get flags for Query, Answer, Authority, and Additional Sections when using \$dig www.VivianDo.com, showing information on the IP addresses of the web and more. On the Attacker VM, it showed boxes of the DNS questions, answers, and information on the flags and IPs. When there were unknown answers, there would be random combination pairs of letters and numbers. The Netwox 105 tool sniffs and replies to DNS requests, always returning the same information.



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

I don't think my cache is saved in my dump.db file, but here is what I have.



Part II Lab: Local DNS Attack

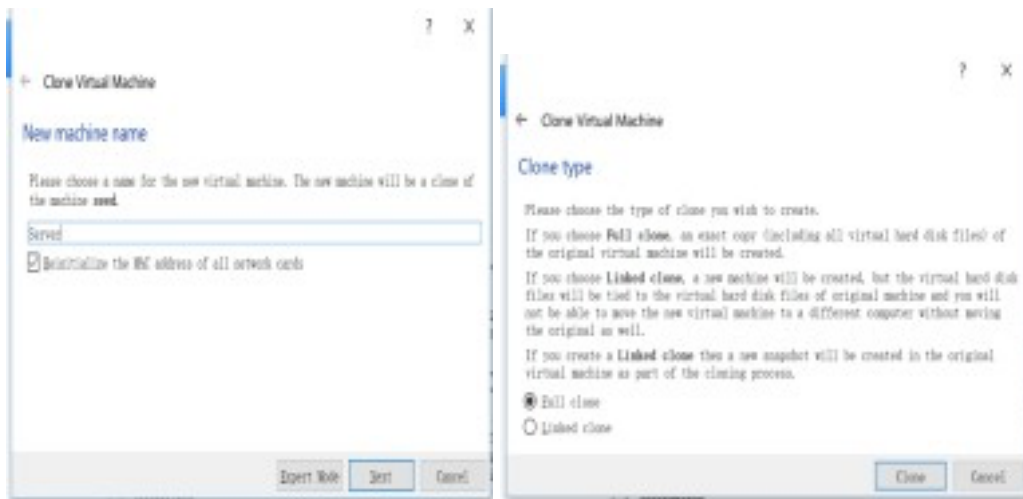
1. Local DNS setup

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, intending 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 try various DNS attacks on the target 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. We will use three virtual machines (VM) to simplify the lab environment: User VM, Attacker VM, and DNS Server VM. The VM configuration is based on Ubuntu, the pre-built operating system we used in the assignment. Make sure to install VirtualBox; go to https://seedsecuritylabs.org/lab_env.html to download the pre-built VMs, and follow this document (https://seedsecuritylabs.org/Labs_16.04/Documents/SEEDVM_VirtualBoxManual.pdf) to run and configure the VM on VirtualBox.

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



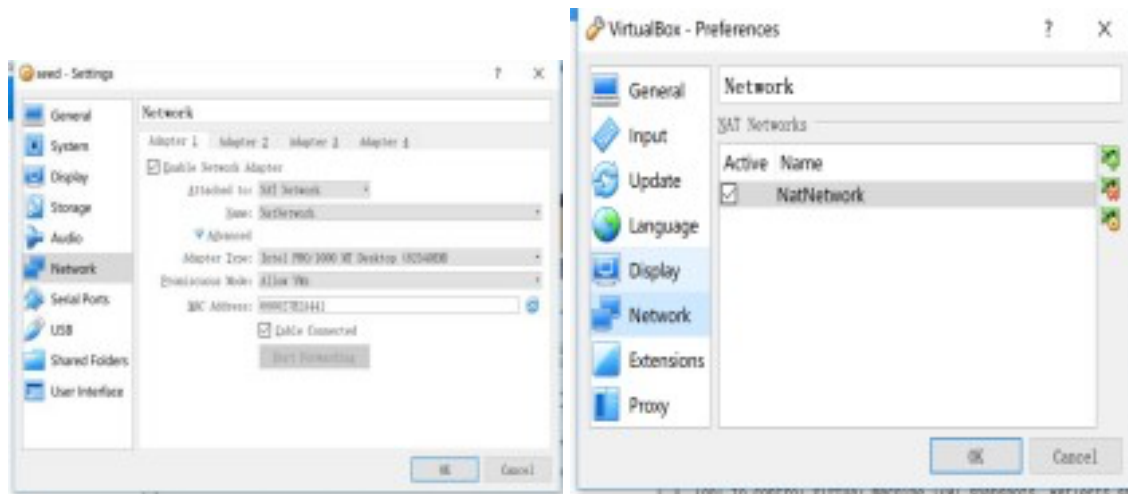
After you correctly perform the clone function, you should have 3 different VMs in your VirtualBox.

Q1: Fill in the information in the following table I and show a screenshot of how you obtained 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 this information.)

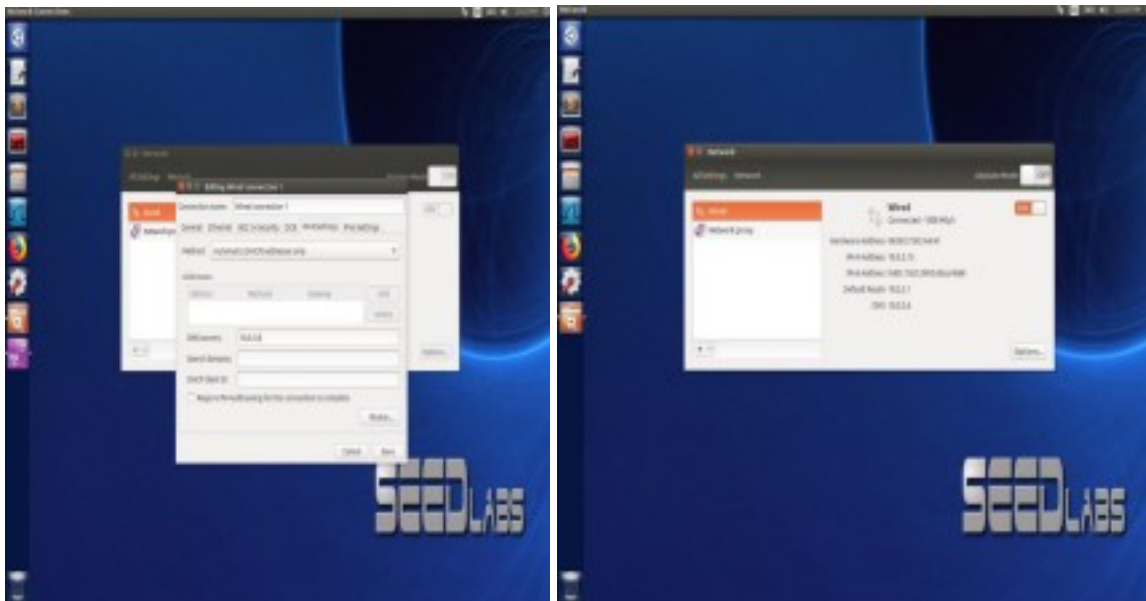
- (2) Create an NAT Adaptor: File=>Preference=>Network=>Adds New NAT Network
- (3) For each VM, we need to modify the Network Adaptor as: “Attached to NAT Network” and change the “Promiscuous Mode: Allow VMs.”



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.



Q2: Restart the VMs and fill in 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 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 the 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 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 the 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, and 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, where 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, where 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 interested in the syntax of the zone file can refer to RFC 1035 for details.

Create a file named `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 an SOA (Start Of Authority) RR, an NS (Name Server) RR, an MX (Mail eXchanger) RR, and 4 A (host Address) RRs.

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

(5) Set up 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 the Bind server

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

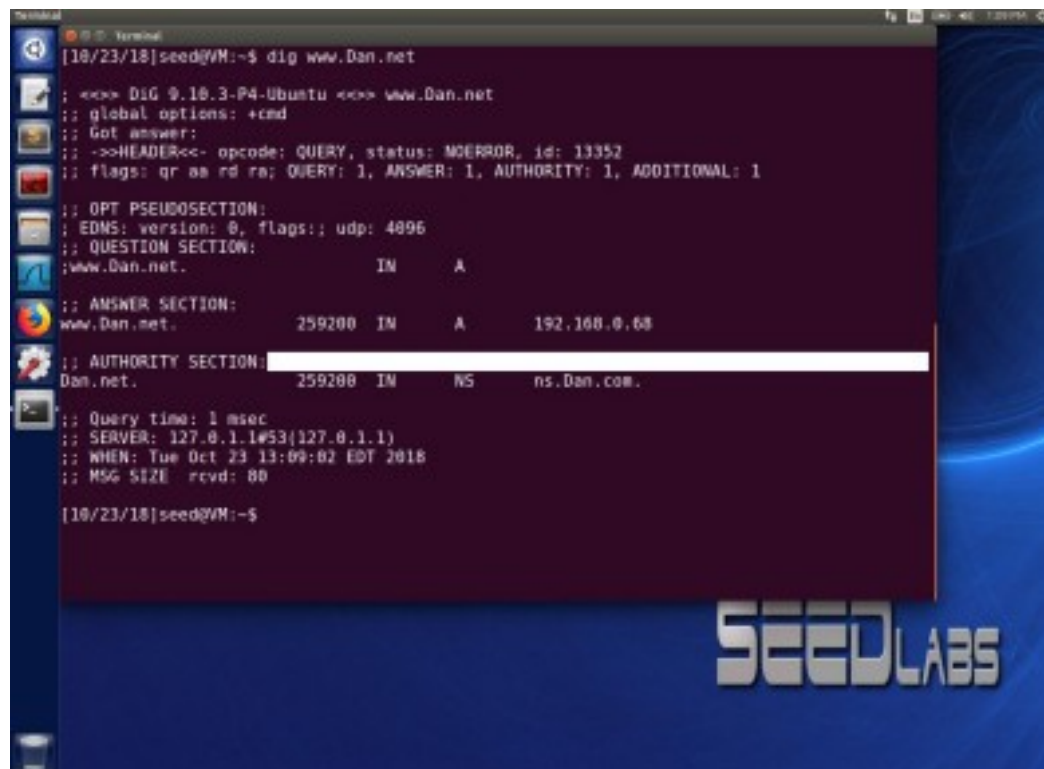
```
$ sudo service bind9 restart
```

1.4 Test Local DNS Server

- (1) Start the Wireshark on your DNS server to capture the 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 are 68.



```
[18/23/18]seed@VM:~$ dig www.Dan.net
;; <<<< Dig 0.18.3-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:
;; 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

[18/23/18]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 the 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 hostname. For example, when the user attempts to access online banking, if the adversaries can redirect the user to a malicious website that looks very much like the main website of bank, the user might be fooled and give away the password of their online banking account. When a user types the name of a website (a hostname, 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 hostname.

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 hostname 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 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 a certain period. Next time, when a user's machine wants to resolve the same hostname, 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 provides 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
```

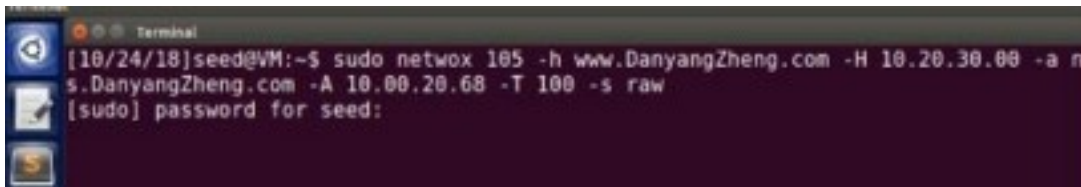
On the DNSServer VM

- (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, the dump file is in the directory of

```
/var/cache/bind  
$sudo rndc dumpdb -cache
```

On the Attacker's VM

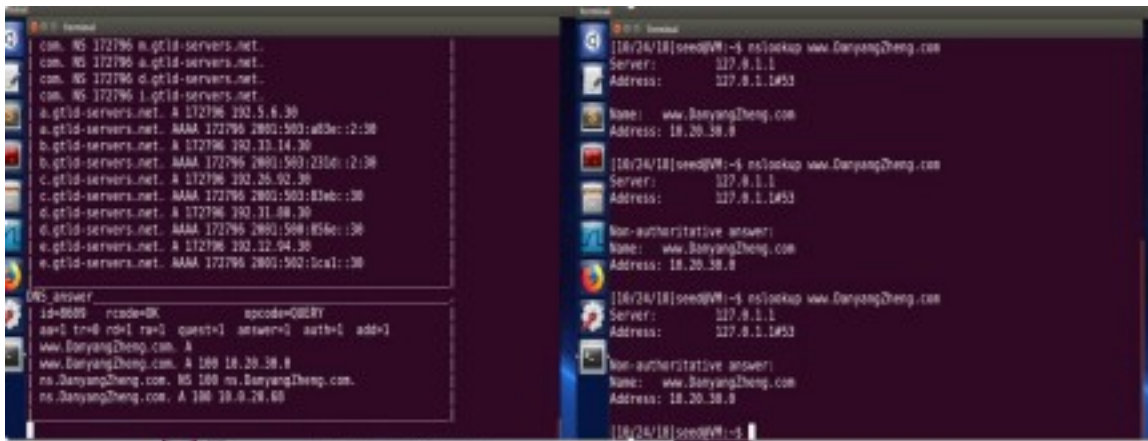
- (4) In the Attacker VM, announce a nonexistent web www.YourFullName.com with a nonexistent Authoritative Name Server ns.YourFullName.com. The IP address of the website is 10.20.30.first-two-digits-of-PantherID and the IP address of your ANS is 10.first-two-digits-of-PantherID.20.last-two-digits-of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



On the User VM

- (5) Using the 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 a user with the wrong IP address.



- (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 dump 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 the dump.dp.

```

; additional
86393 DS 30909 8 2 {
E2D3C916F6DEEAC73294E8268FB5885044A8
33FC5459588F4A91B4CFC41A5766 }

; additional
86393 RRSIG DS 8 1 86400 {
20181106050000 20181024040000 2134 .
akFKaLMwqnfng1bgJT5kLXBHlnN3B110Tgm
lCbppXPQ2TKoRQtE/e0tvrd6Zo2qIoLNKRwY
xpapFMMRJM/yf5/kZNL/ab4Rh+q8aA/9IKCE
cSP/aWJvZp+WMfoSh22IdfKph68vI3nadxYb
g1sLb4ZQTyuEx9cx9Rt0PvyfLIjVbk2Krayf
tZE6513KAi.pw2j/Vz18V4edCd+LDavJRCTJR
QRxd1hUH23Hy6Pth2keMteY4SGz3K7ZnUp8K
bRC18FvgIax/Dk0o/P/onJj8pew72W2La2NC
HL4gA17NMHCCGjee+EjWPELnIHkyAP6PhTsV
a3HivE3lCDYk1fDy6g== }

; authauthority
ns.DanyangZheng.com. 93 NS ns.DanyangZheng.com.
; additional
93 A 10.0.20.60
; 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 2001:503:2316::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

```