



PHARMING ATTACKS ON DNS



MARCH 12, 2016
DEEPANSHU LULLA
CS6740 NORTHEASTERN UNIVERSITY

[Contents](#)

Introduction	2
Setting up Web Servers.....	3
Setting up DNS servers.....	8
Task 1: Host File Compromise	14
Task 2: DNS ID Poisoning	15
Task 3: DNS Cache poisoning	18

Introduction

The Domain Name System (DNS) is a hierarchical decentralized naming system for computers, services, or any resource connected to the Internet or a private network. The primary purpose is to translate hostnames to IP addresses.

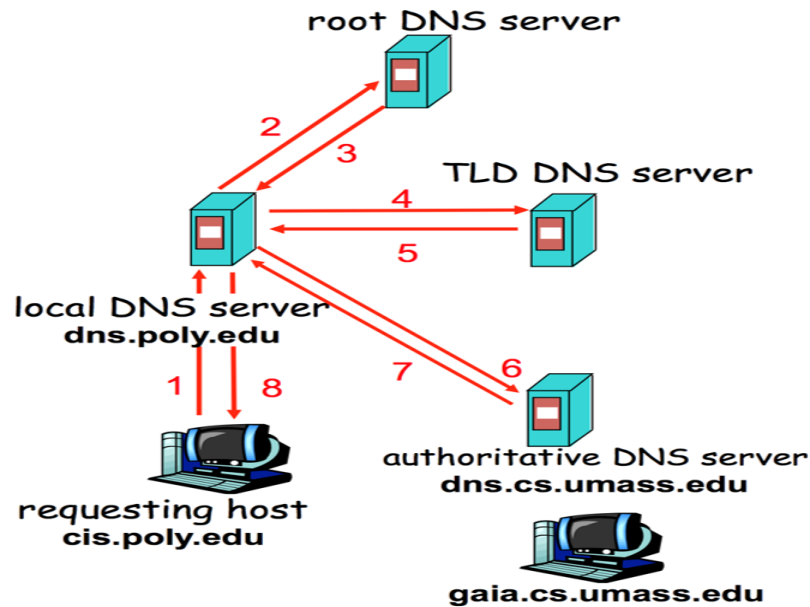
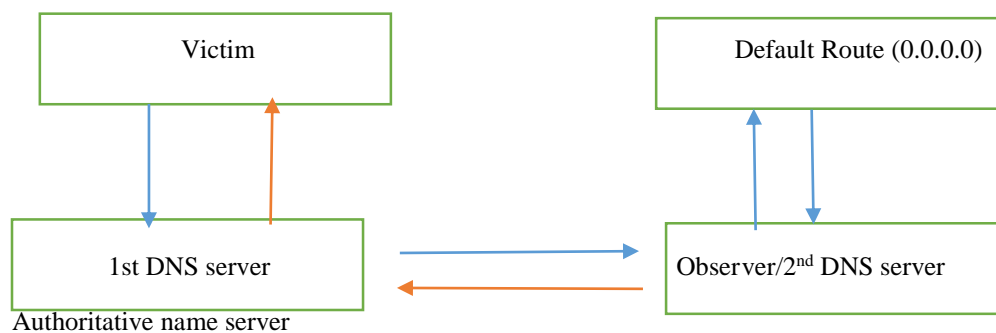


Figure 1: The way DNS servers works

For the project it was needed to setup the servers as well as the lab first. Setting up servers was a challenge for me as I set up

- Two web server: one original web server and one attacker's web server
- Two DNS servers



for www.origwebserver.com

The main objective of Pharming 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.

The attacker VM acted as an attacker. The DNS server for everyone is set as the 1st DNS Server. The DNS server forwards the queries to Observer who then forwards them to 8.8.8.8 or any name resolution server. The first DNS server also acts as an Authoritative Name Server for www.origwebserver.com.

The victim queries for a domain name. The request is first checked in hosts file and then later checked in the cache of victim. If no entry is found the name is sent to 1st DNS server for name resolution. This entry is forwarded to 2nd DNS server which resolves queries for Internet.

I set up 6 virtual machines and the configurations are listed as below:

Host name	IP address	MAC address	Host only Interface
Gateway	192.168.99.1		
Attacker	192.168.99.100	08:00:27:0c:53:b7	Eth13
Victim	192.168.99.101	08:00:27:b1:47:d5	Eth14
Original web server www.origwebserver.com	192.168.99.103	08:00:27:53:da:a8	Eth14
Attacker's web server www.attackerwebserver.com	192.168.99.104	08:00:27:09:1d:2d	Eth14
1 st DNS server (authoritative name server for www.origwebserver.com)	192.168.99.102	08:00:27:b6:56:a3	Eth15
Observer/2 nd DNS server	192.168.99.105	08:00:27:EA:AE:F7	Eth16

Figure 2: Table for mapping IP and Mac addresses in host only interfaces

Please note no zone file was configured for www.attackerwebserver.com, 1st DNS server only had zone information for www.origwebserver.com

Also since all VMs of seed Ubuntu had bind9 and apache2 pre installed. I had to uninstall bind9 in victim and attacker where it was not needed.

Three attacks were done on local DNS servers

Attack	Successful
Host File Compromise	Yes
DNS ID Poisoning	Yes
DNS cache poisoning	Yes

Figure 3: Table for List of attacks and their success

Setting up Web Servers

Step 1:

```
sudo apt-get update
```

```
sudo apt-get install apache2
```

Step 2:

In attacker's web server

```
sudo mkdir -p /var/www/attackerwebserver.com/public_html
```

```
sudo chmod -R 755 /var/www
```

In original web server

```
sudo mkdir -p /var/www/origwebserver.com/public_html
```

Step 3:

Give access of this directory to other hosts

In attacker's web server

```
sudo chown -R $USER:$USER /var/www/attackerwebserver.com/public_html
```

```
sudo chmod -R 755 /var/www
```

In original web server

```
sudo chown -R $USER:$USER /var/www/origwebserver.com/public_html
```

```
sudo chmod -R 755 /var/www
```

```
sudo chmod -R 755 /var/
```

```
sudo chmod -R 755 /var/www/origwebserver.com/public_html/
```

Step 4:

Creating Html file in the folder

In attacker's web server

```
sudo touch /var/www/attackerwebserver.com/public_html/index.html
```

In original web server

```
sudo touch /var/www/origwebserver.com/public_html/index.html
```

Step5:

Now the default virtual host file in apache 2 is default We will copy it over to create a virtual host file for our domain.

For attackerwebserver's web server

```
sudo cp /etc/apache2/sites-available/default /etc/apache2/sites-available/  
attackerwebserver.com.conf
```

Original web server

```
sudo cp /etc/apache2/sites-available/default /etc/apache2/sites-available/  
origwebserver.com.conf
```

Step 6:

Opening file in editor.

A

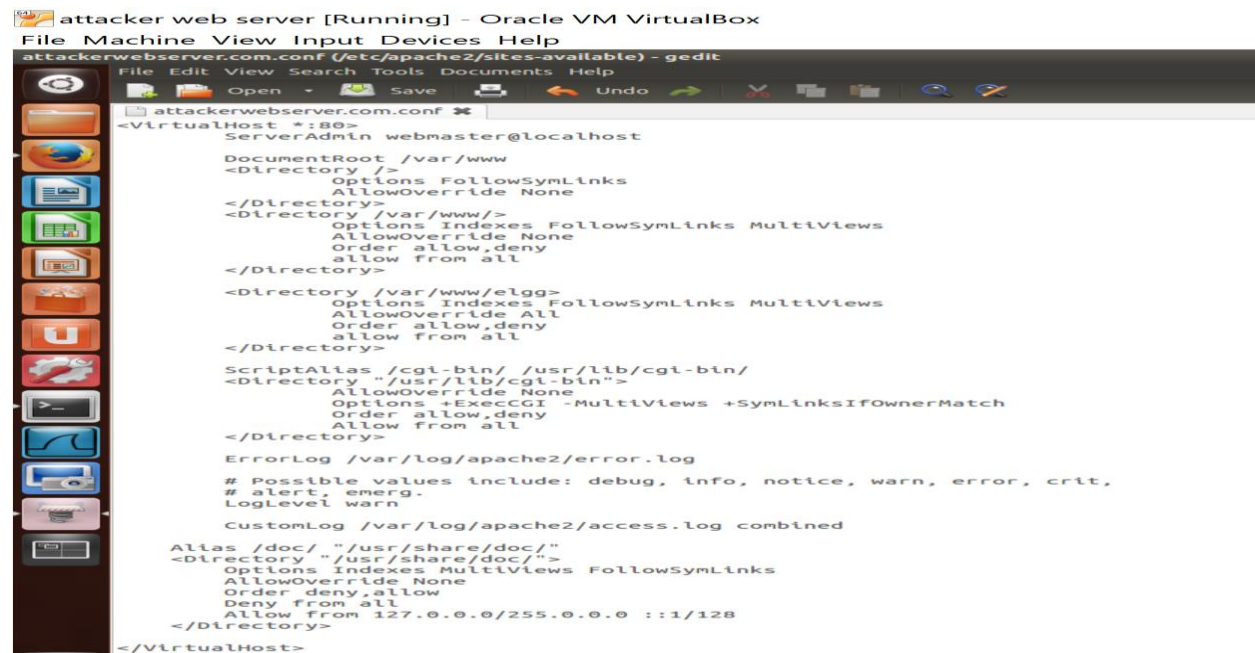


Figure 4: Conf file for web server

In attackerwebserver

```
sudo gedit /etc/apache2/sites-available/attackerwebserver.com.conf
```

Replace ServerName with attackerwebserver.com

Add ServerAlias with value www.attackerwebserver.com

Change server admin to

ServerAdmin admin@attackerwebserver.com

Alter the DocumentRoot directive to reflect the directory we created:

DocumentRoot /var/www/attackerwebserver.com/public_html

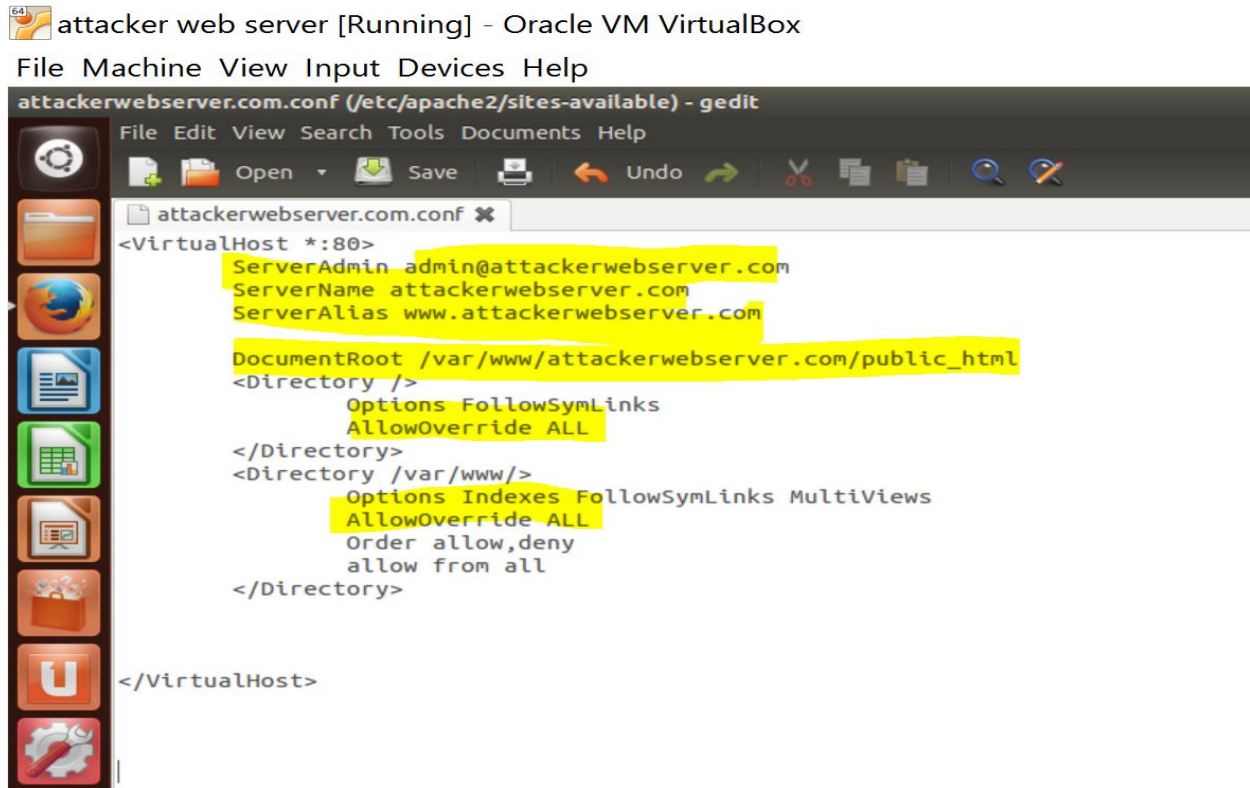


Figure 5: changes made in conf file of web server

Next open default and set AllowOverride ALL in all lines. This step is important.

Do the same changes in origwebserver.

Step 7: Change the html pages of servers.

`sudo gedit /var/www/attackerwebserver.com/public_html/index.html`

```
<html>
<head>
  <title>Welcome to attackerwebserver.com!</title>
</head>
<body>
  <h1> Welcome to attackerwebserver.com!.This site belongs to attackerwebserver</h1>
</body>
</html>
```

Do similar changes in origwebserver

Step 8:

Enable new host files and restart web server

In attackerwebserver

```
sudo a2dissite default
```

```
sudo a2ensite attackerwebserver.com.conf
```

```
sudo service apache2 restart
```

In origwebserver

```
sudo a2ensite origwebserver.com.conf
```

```
sudo service apache2 restart
```

Step 9: Open hosts file

In attacker

```
sudo gedit /etc/hosts
```

Add the line

<IP-address of host only interface> <domain_name>

192.168.99.103 www.attackerwebserver.com

Make similar changes in orig web server

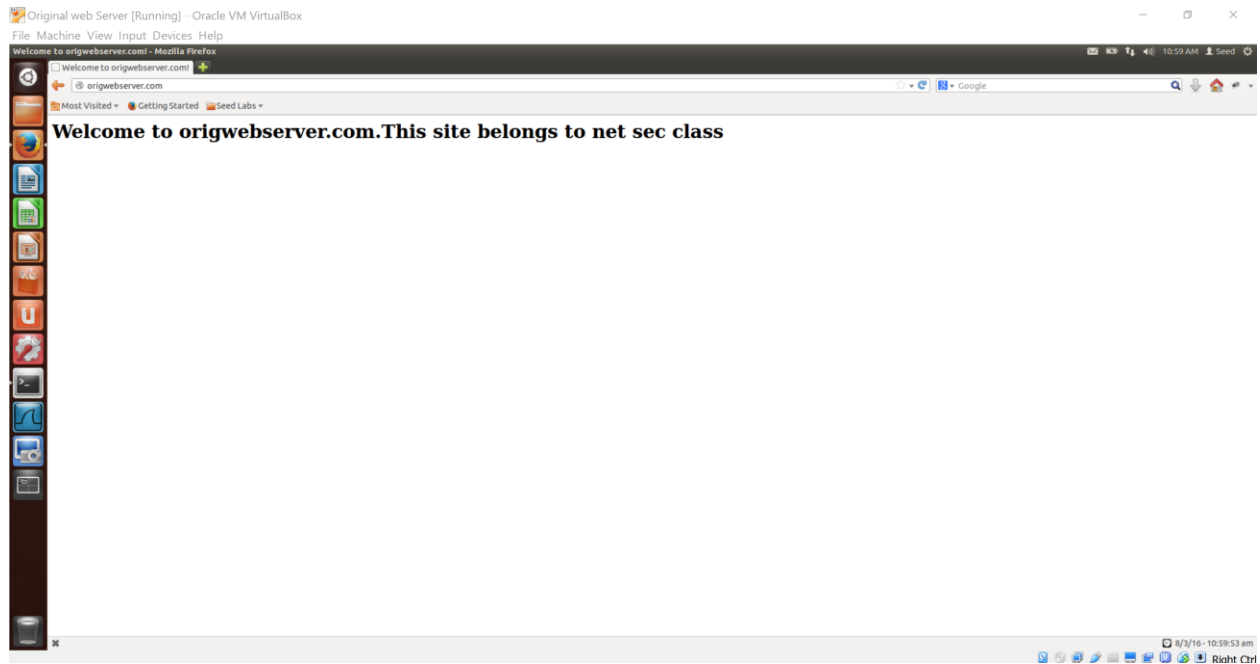


Figure 6: Web Site for www.origwebserver.com

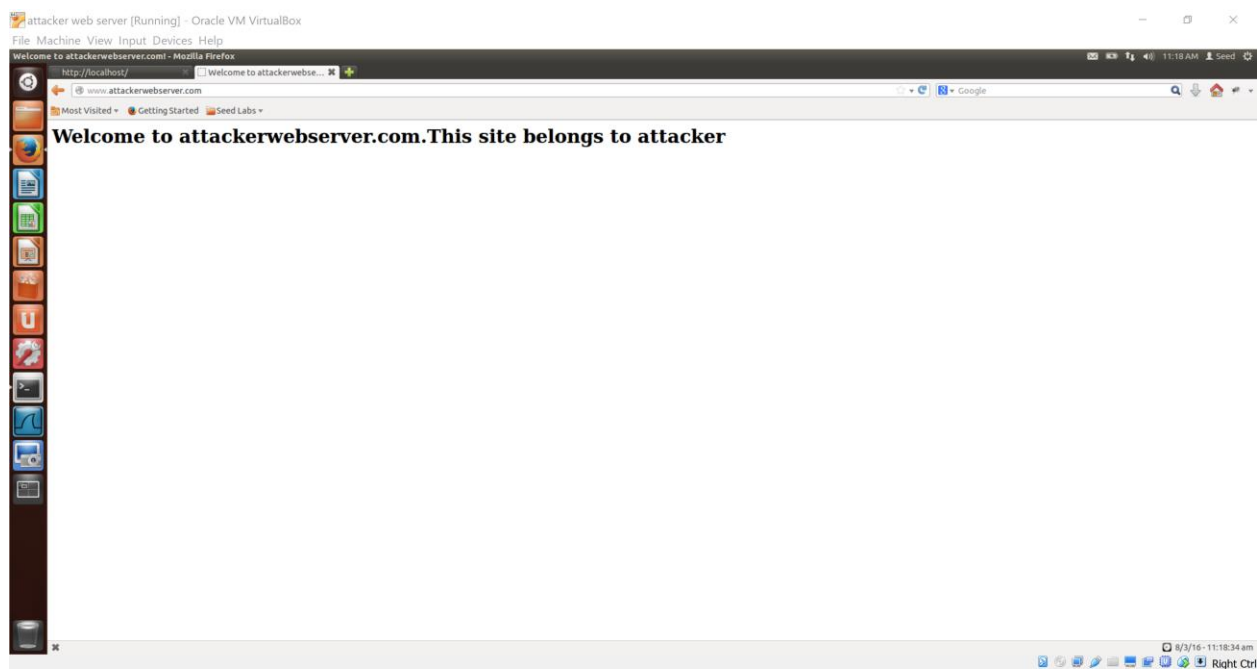


Figure 7: Website for www.attackerwebserver.com

Setting up DNS servers

Step 1: Install bind

\$sudo apt-get install bind9

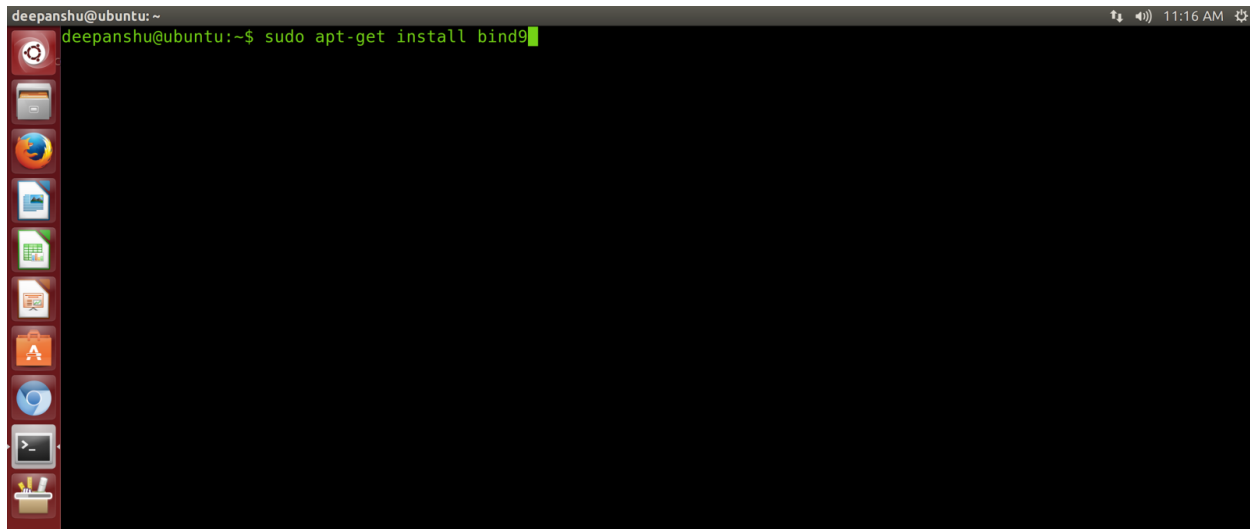


Figure 8: Installing bind9

Step2:

Then open the file named.conf.options in the /etc/bind/ directory

\$sudo gedit /etc/bind/named.conf.options

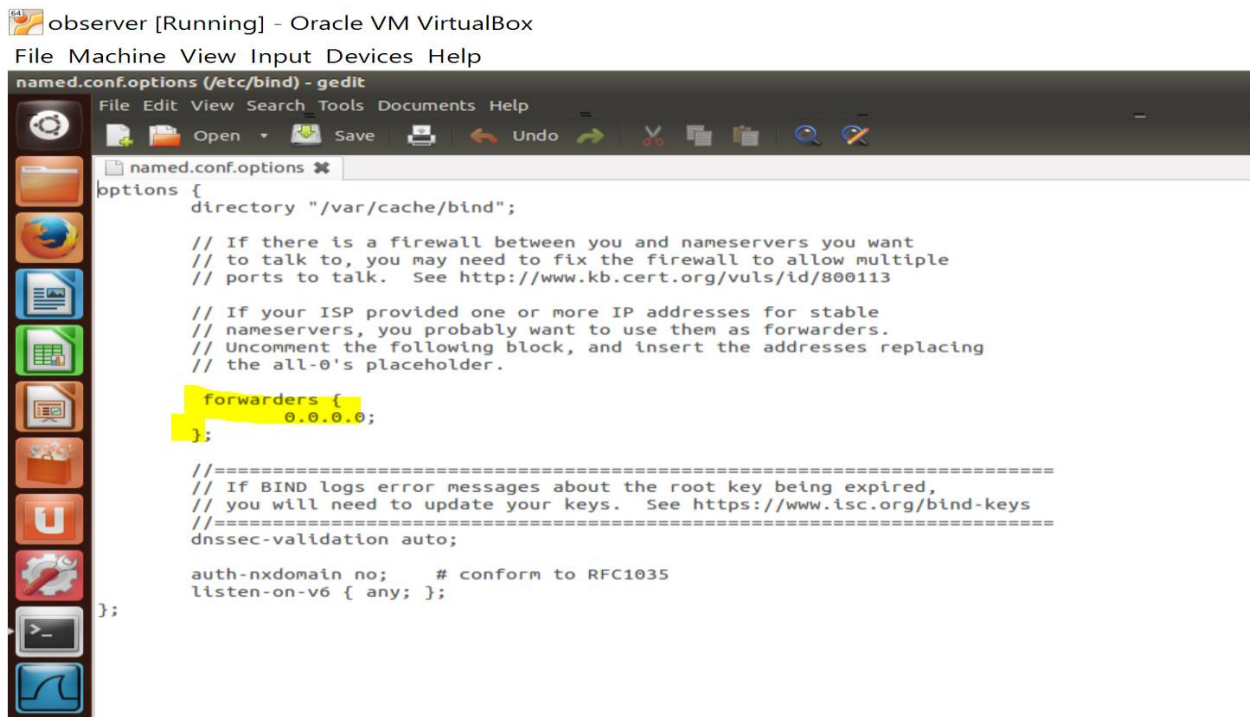
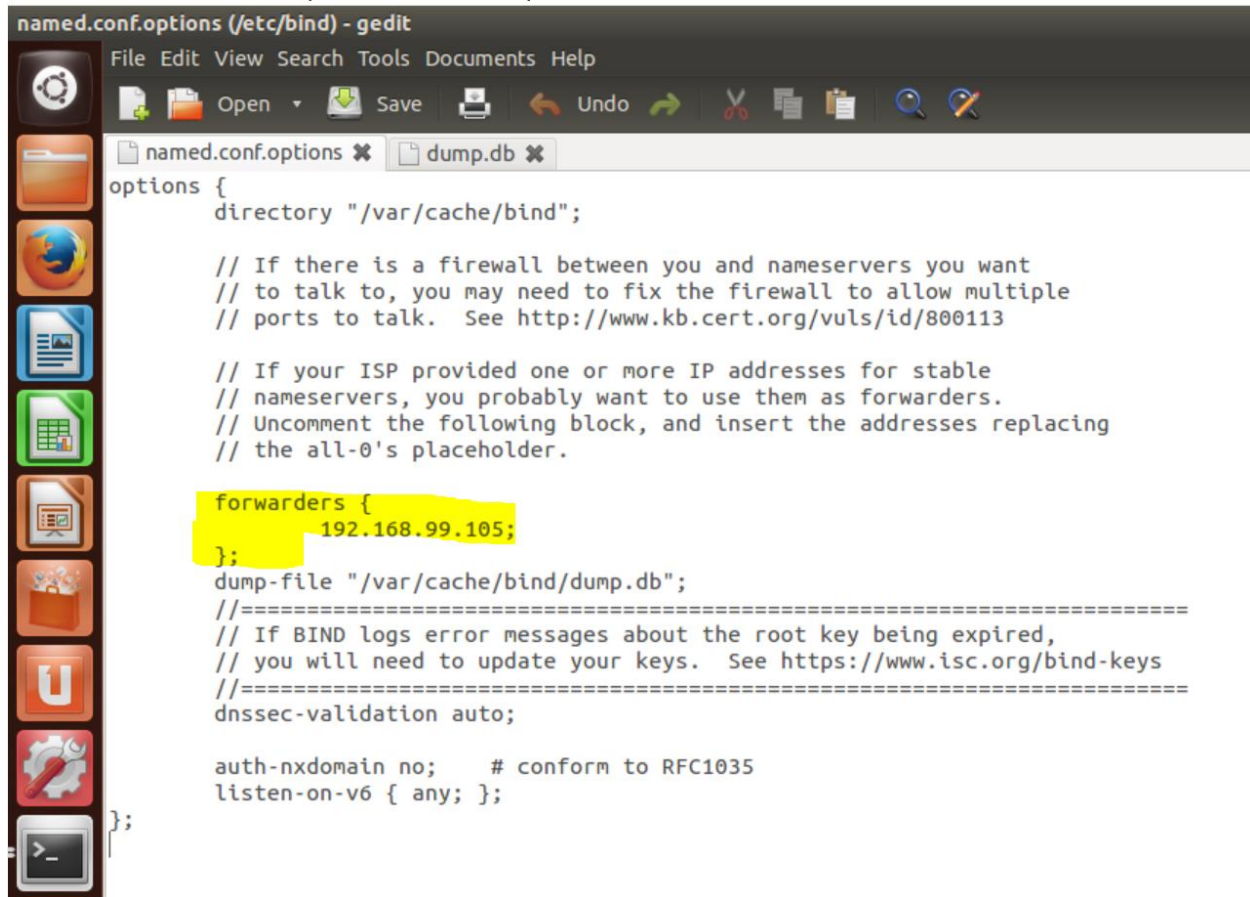


Figure 9: DNS Forwarding configured for 2nd DNS server

64 dnsServer [Running] - Oracle VM VirtualBox

File Machine View Input Devices Help



```
named.conf.options (/etc/bind) - gedit
File Edit View Search Tools Documents Help
named.conf.options x dump.db x
options {
    directory "/var/cache/bind";

    // If there is a firewall between you and nameservers you want
    // to talk to, you may need to fix the firewall to allow multiple
    // ports to talk. See http://www.kb.cert.org/vuls/id/800113

    // If your ISP provided one or more IP addresses for stable
    // nameservers, you probably want to use them as forwarders.
    // Uncomment the following block, and insert the addresses replacing
    // the all-0's placeholder.

    forwarders {
        192.168.99.105;
    };
    dump-file "/var/cache/bind/dump.db";
    //=====
    // If BIND logs error messages about the root key being expired,
    // you will need to update your keys. See https://www.isc.org/bind-keys
    //=====
    dnssec-validation auto;

    auth-nxdomain no;    # conform to RFC1035
    listen-on-v6 { any; };
};
```

Figure 10: DNS forwarding for 1st DNS Server. 192.168.99.105 is IP address of 2nd DNS server

Step3:

Our system must work atleast as a forwarder.

\$dig @127.0.0.1 www.facebook.com

So what the above command does is it forwards the requests from our localhost to google's DNS which acts as a recursive server to resolve facebook's IP address.

```

deepanshu@ubuntu:~$ clear
deepanshu@ubuntu:~$ dig @127.0.0.1 www.facebook.com

;; <<>> DiG 9.9.5-3ubuntu0.5-Ubuntu <<>> @127.0.0.1 www.facebook.com
;; (1 server found)
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 22086
;; flags: qr rd ra; QUERY: 1, ANSWER: 2, AUTHORITY: 13, ADDITIONAL: 1

;; OPT PSEUDOSECTION:
;; EDNS: version: 0, flags:; udp: 4096
;; QUESTION SECTION:
;; www.facebook.com.
;;
;; ANSWER SECTION:
www.facebook.com. 3568 IN CNAME star-mini.c10r.facebook.com.
star-mini.c10r.facebook.com. 21 IN A 31.13.73.36

;; AUTHORITY SECTION:
3621 IN NS d.root-servers.net.
3621 IN NS a.root-servers.net.
3621 IN NS h.root-servers.net.
3621 IN NS l.root-servers.net.
3621 IN NS c.root-servers.net.
3621 IN NS b.root-servers.net.
3621 IN NS k.root-servers.net.
3621 IN NS j.root-servers.net.
3621 IN NS e.root-servers.net.
3621 IN NS m.root-servers.net.
3621 IN NS i.root-servers.net.
3621 IN NS f.root-servers.net.

```

Figure 11: Queries for websites outside LAN are forwarded to be resolved from Internet

Step4:

Next open the file named.conf.local in the /etc/bind/ directory

```
$sudo gedit /etc/bind/named.conf.local
```

So in this step we are specifying the locations where we are going to save the file. Type the configuration as shown in the image for creating a master server.

dnsServer [Running] - Oracle VM VirtualBox

```

named.conf.local (/etc/bind) - gedit
File Edit View Search Tools Documents Help
named.conf.local x
//
// Do any local configuration here
//
// Consider adding the 1918 zones here, if they are not used in your
// organization
//include "/etc/bind/zones.rfc1918";

zone "origwebserver.com" {
    type master;
    file "/etc/bind/db.origwebserver.com";
};

#ipv4 address of orig web server 192.168.99.103
zone "99.168.192.in-addr.arpa" {
    type master;
    file "/etc/bind/db.origwebserver.rev";
};

```

Figure 12: Zone File mapping in named.conf.local

Step 5:

Change your current directory to /etc/bind/

```
$cd /etc/bind
```

Create a file db.origwebserver.com

```
$touch db.origwebserver.com
```

Copy db.local file to db.origwebserver.com

```
$sudo cp db.local db.origwebserver.com
```

Then open db.origwebserver.com in editor.

```
$sudo gedit db.origwebserver.com
```

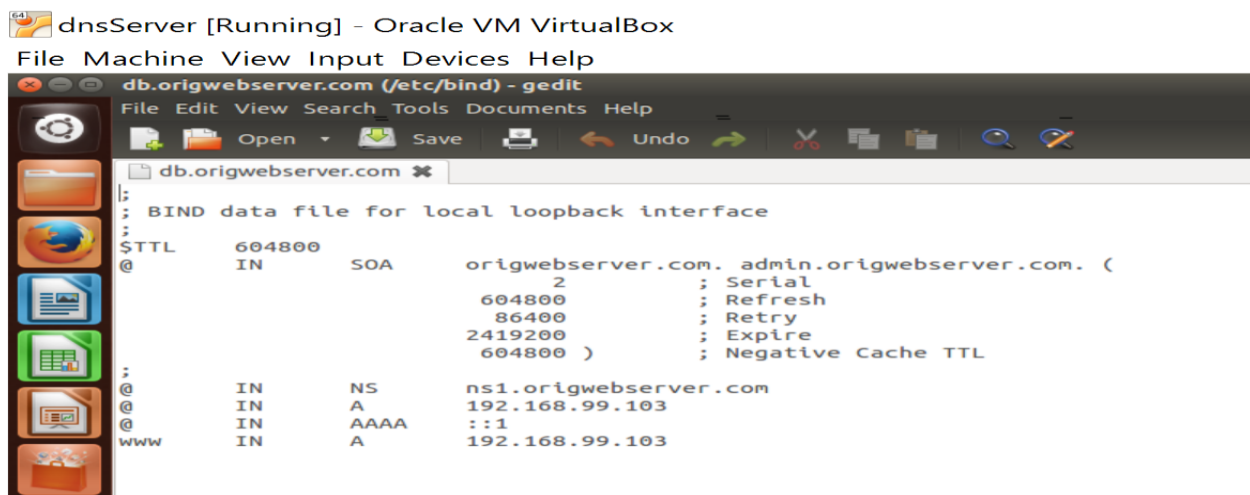


Figure 13: DNS records in first DNS server

Step 6:

Create a file db.origwebserver.rev


```
$touch db.origwebserver.rev
```

Copy db.origwebserver.com file to db.origwebserver.rev

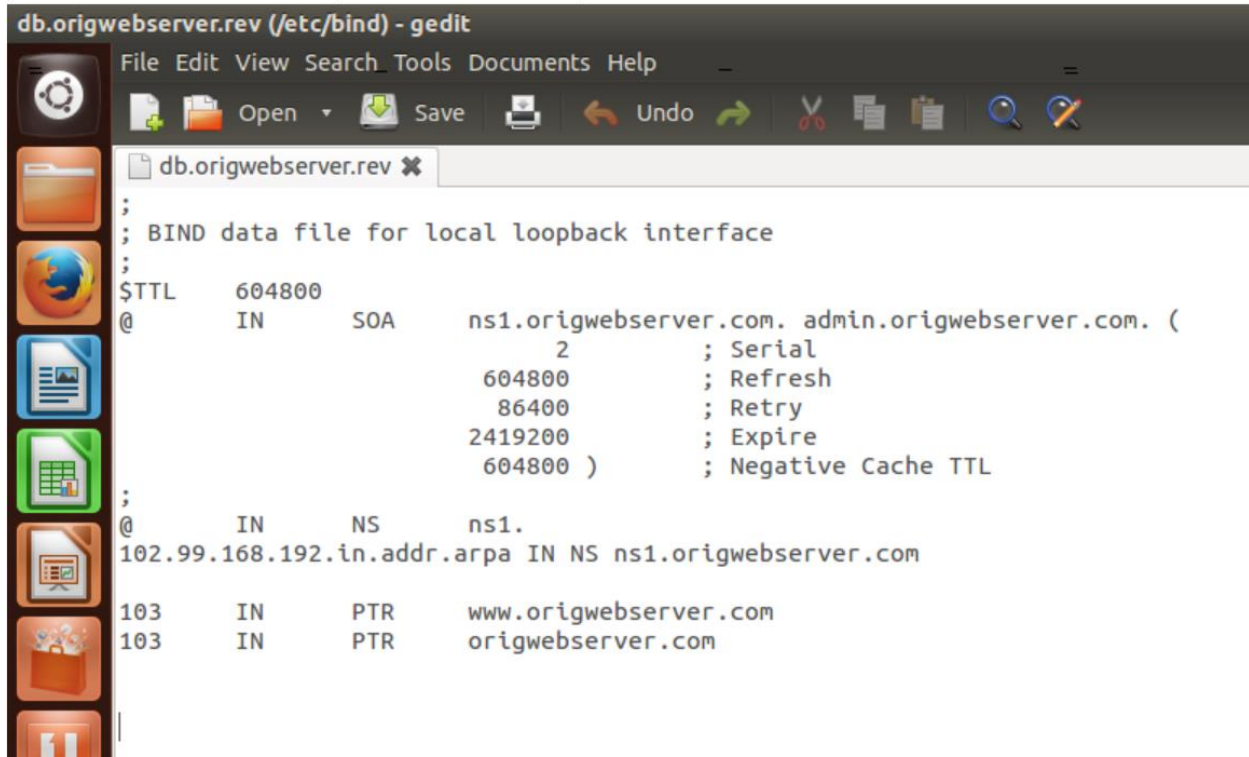
```
$sudo cp db.origwebserver.com db.origwebserver.rev
```

Then open db.origwebserver.rev in editor.

```
$sudo gedit db.origwebserver.rev
```

64  dnsServer [Running] - Oracle VM VirtualBox

File Machine View Input Devices Help



```
db.origwebserver.rev (/etc/bind) - gedit
File Edit View Search Tools Documents Help
Open Save Undo
db.origwebserver.rev x
;
; BIND data file for local loopback interface
;
$TTL      604800
@        IN      SOA      ns1.origwebserver.com. admin.origwebserver.com. (
                                2          ; Serial
                                604800     ; Refresh
                                86400      ; Retry
                                2419200    ; Expire
                                604800 )   ; Negative Cache TTL
;
@        IN      NS       ns1.
102.99.168.192.in.addr.arpa IN NS ns1.origwebserver.com

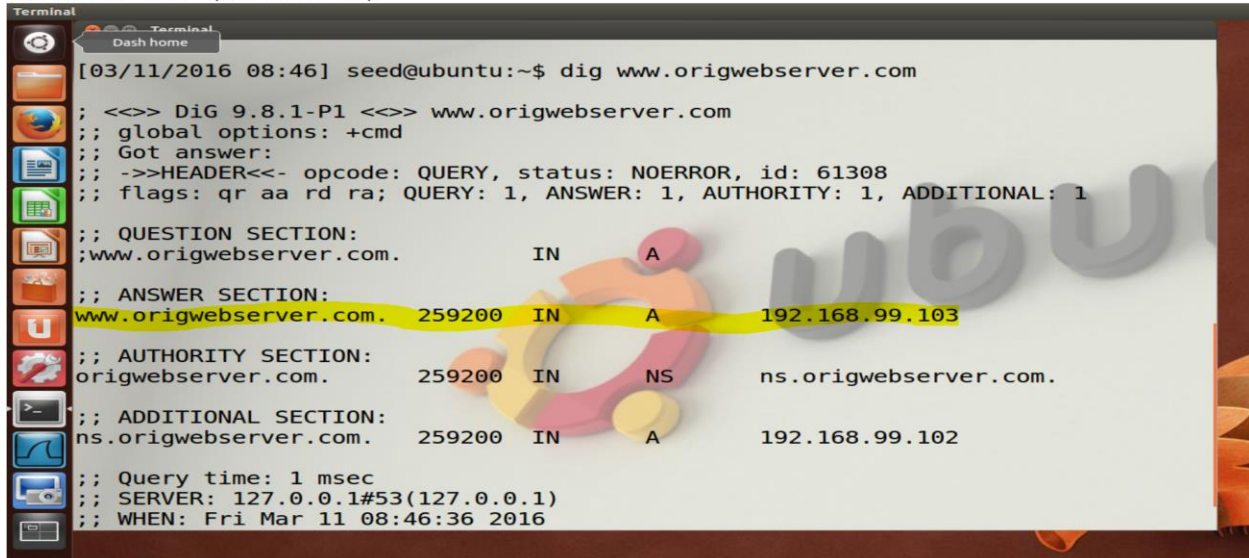
103      IN      PTR      www.origwebserver.com
103      IN      PTR      origwebserver.com
```

Figure 14: PTR records

This should be the final result of the complete setup.

 victim [Running] - Oracle VM VirtualBox

File Machine View Input Devices Help



```
Terminal
Dash home
[03/11/2016 08:46] seed@ubuntu:~$ dig www.origwebserver.com
; <<<> DiG 9.8.1-P1 <<<> www.origwebserver.com
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 61308
;; flags: qr aa rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 1, ADDITIONAL: 1
;; QUESTION SECTION:
;www.origwebserver.com.      IN      A
;; ANSWER SECTION:
www.origwebserver.com.  259200  IN      A      192.168.99.103
;; AUTHORITY SECTION:
origwebserver.com.      259200  IN      NS      ns.origwebserver.com.
;; ADDITIONAL SECTION:
ns.origwebserver.com.   259200  IN      A      192.168.99.102
;; Query time: 1 msec
;; SERVER: 127.0.0.1#53(127.0.0.1)
;; WHEN: Fri Mar 11 08:46:36 2016
```

Figure 15: DNS query should be resolved from victim

Task 1: Host File Compromise

This attack assumes the victim has already been compromised and the attacker can change the host file of the victim.

Before attack

Before the attack, if the victim queries for www.origwebserver.com it is redirected to IP of www.origwebserver.com. The name is resolved using 1st DNS server.

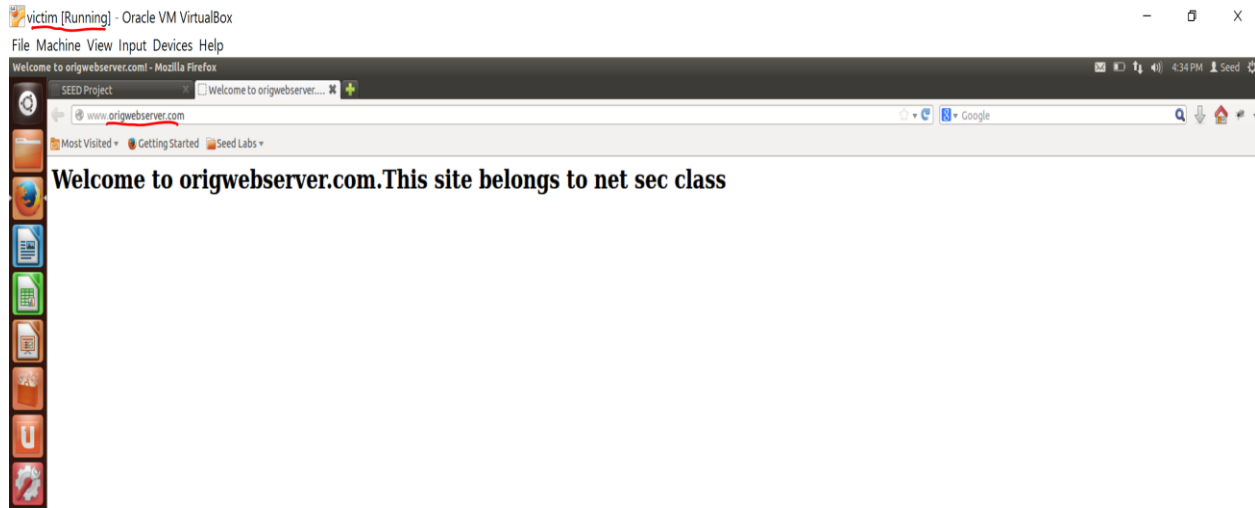


Figure 16: Web site for www.origwebserver.com on being queried for the same

After attack

For the attack I had to make changes in `/etc/hosts` file.

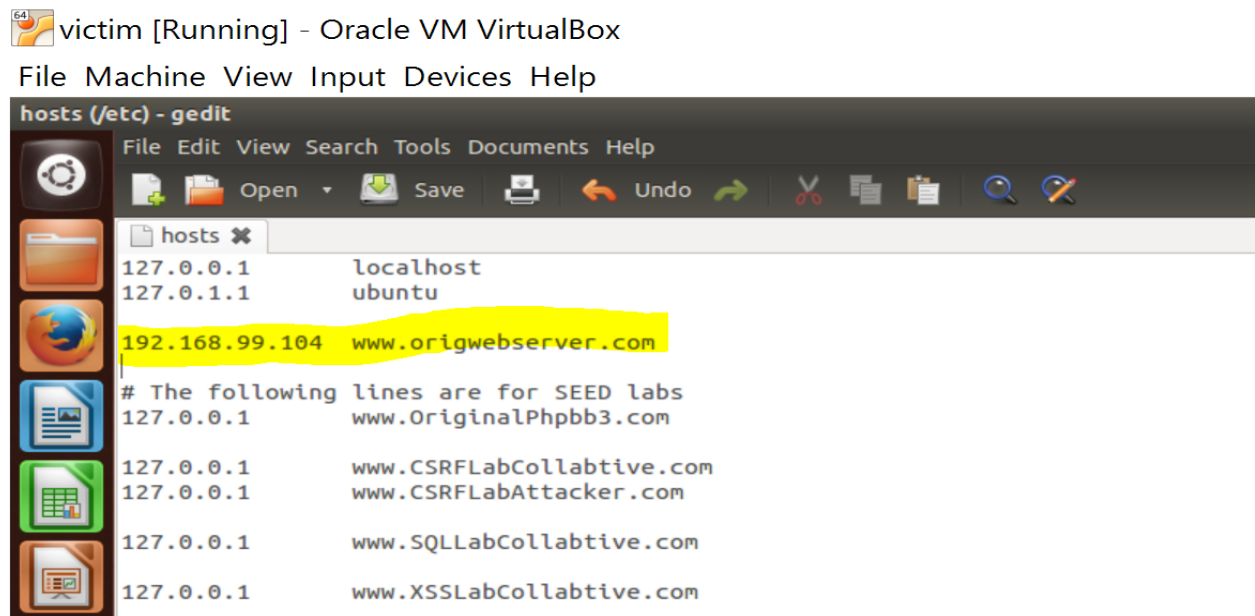


Figure 17: changes in host file for the attack

After adding this incorrect mapping (ip of attackerwebserver.com) to www.origwebserver.com. After clearing the browser as well as the DNS cache, the next request to www.origwebserver.com lead to www.attackerwebserver.com

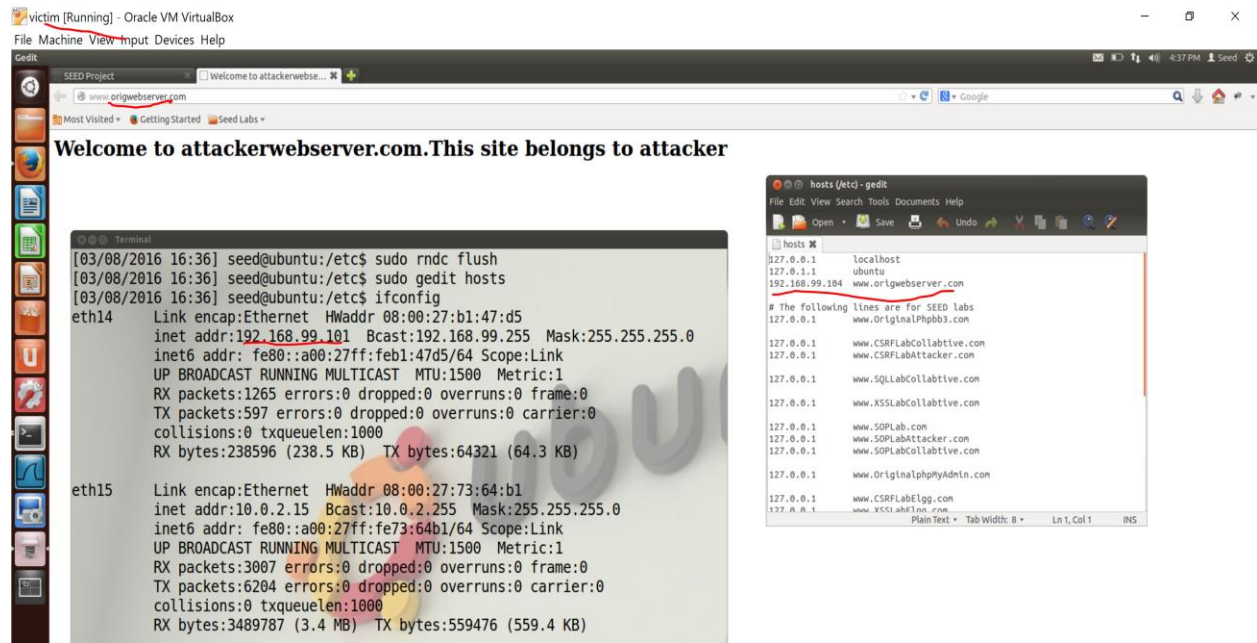
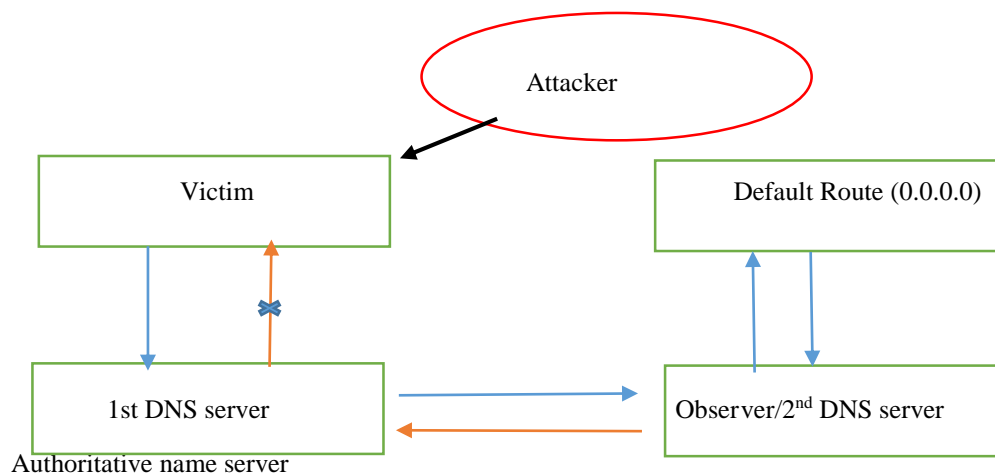


Figure 18: The victim is redirected to attacker's website on querying for www.origwebserver.com

Task 2: DNS ID Poisoning

The next task was DNS ID poisoning. In this attack, the attacker tries to sniff packets between the first DNS server and the victim. Then it tries to send a spoofed response to the victim pretending to be 1st DNS server.



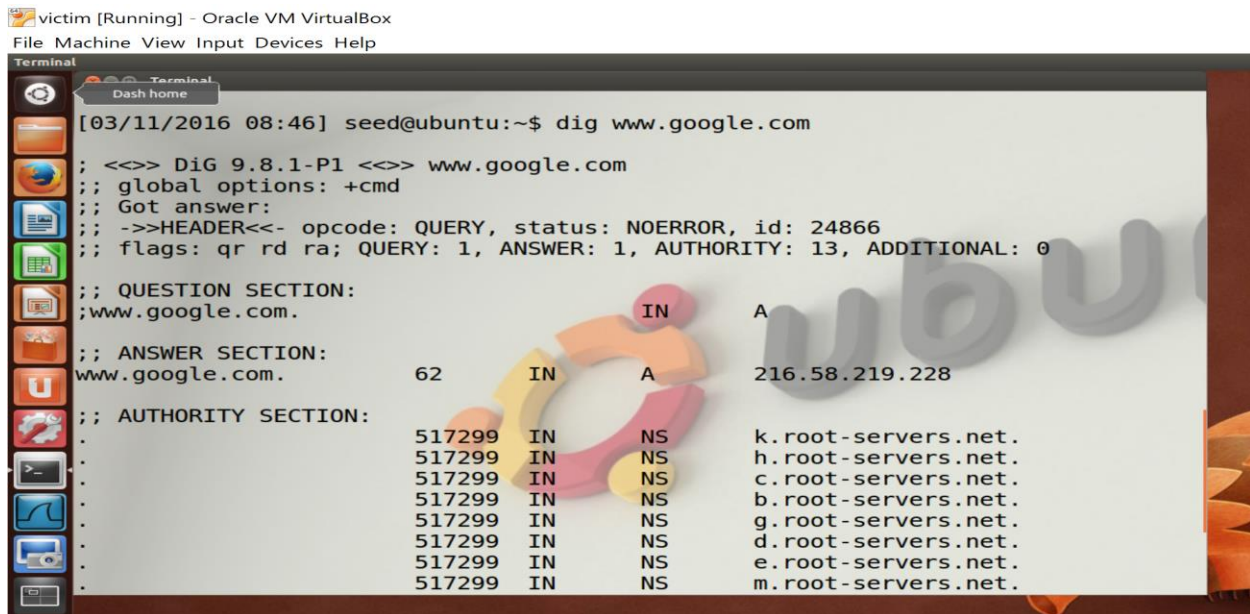
The fake DNS response will be accepted by the user's computer if it meets the following criteria:

1. The source IP address must match the IP address of the DNS server.
2. The destination IP address must match the IP address of the user's machine.
3. The source port number (UDP port) must match the port number that the DNS request was sent to (usually port 53).
4. The destination port number must match the port number that the DNS request was sent from.
5. The UDP checksum must be correctly calculated.
6. The transaction ID must match the transaction ID in the DNS request.
7. The domain name in the question section of the reply must match the domain name in the question section of the request.

8. The domain name in the answer section must match the domain name in the question section of the DNS request.
9. The User's computer must receive the attacker's DNS reply before it receives the legitimate DNS response.

Before Attack

Before the attack, when the victim queries the DNS server the DNS records of google.com are returned.



```

victim [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
Terminal
Dash home
[03/11/2016 08:46] seed@ubuntu:~$ dig www.google.com
; <<>> DiG 9.8.1-P1 <<>> www.google.com
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 24866
;; flags: qr rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 13, ADDITIONAL: 0

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

;; ANSWER SECTION:
www.google.com.      62      IN      A      216.58.219.228

;; AUTHORITY SECTION:
.          517299  IN      NS      k.root-servers.net.
.          517299  IN      NS      h.root-servers.net.
.          517299  IN      NS      c.root-servers.net.
.          517299  IN      NS      b.root-servers.net.
.          517299  IN      NS      g.root-servers.net.
.          517299  IN      NS      d.root-servers.net.
.          517299  IN      NS      e.root-servers.net.
.          517299  IN      NS      m.root-servers.net.

```

Figure 19: Query for google.com before attack

After attack

There is an important step which needs to be done in both DNS servers for the attack to be seen with immediate effect. It is to flush the DNS cache of both DNS servers.

```
$ sudo rndc flush
```

Now when the victim queries for google.com. The attacker's response is accepted first and the victim is redirected to attacker's web site when it requests google.com.

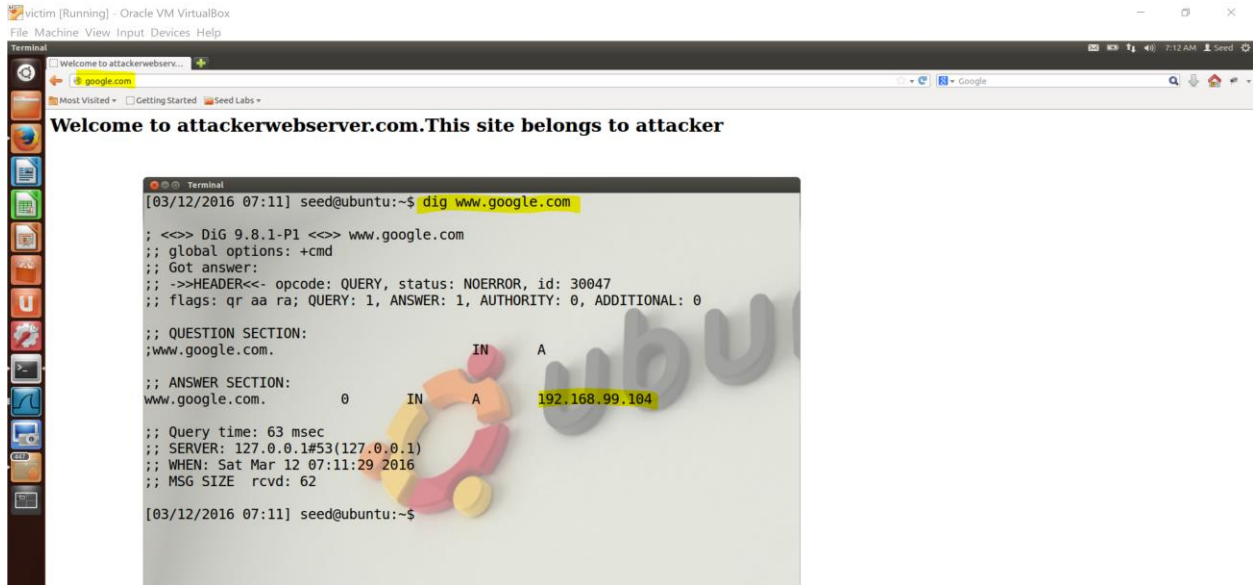


Figure 20: Victim is redirected to attacker's web site and dig for google.com returns poisoned result

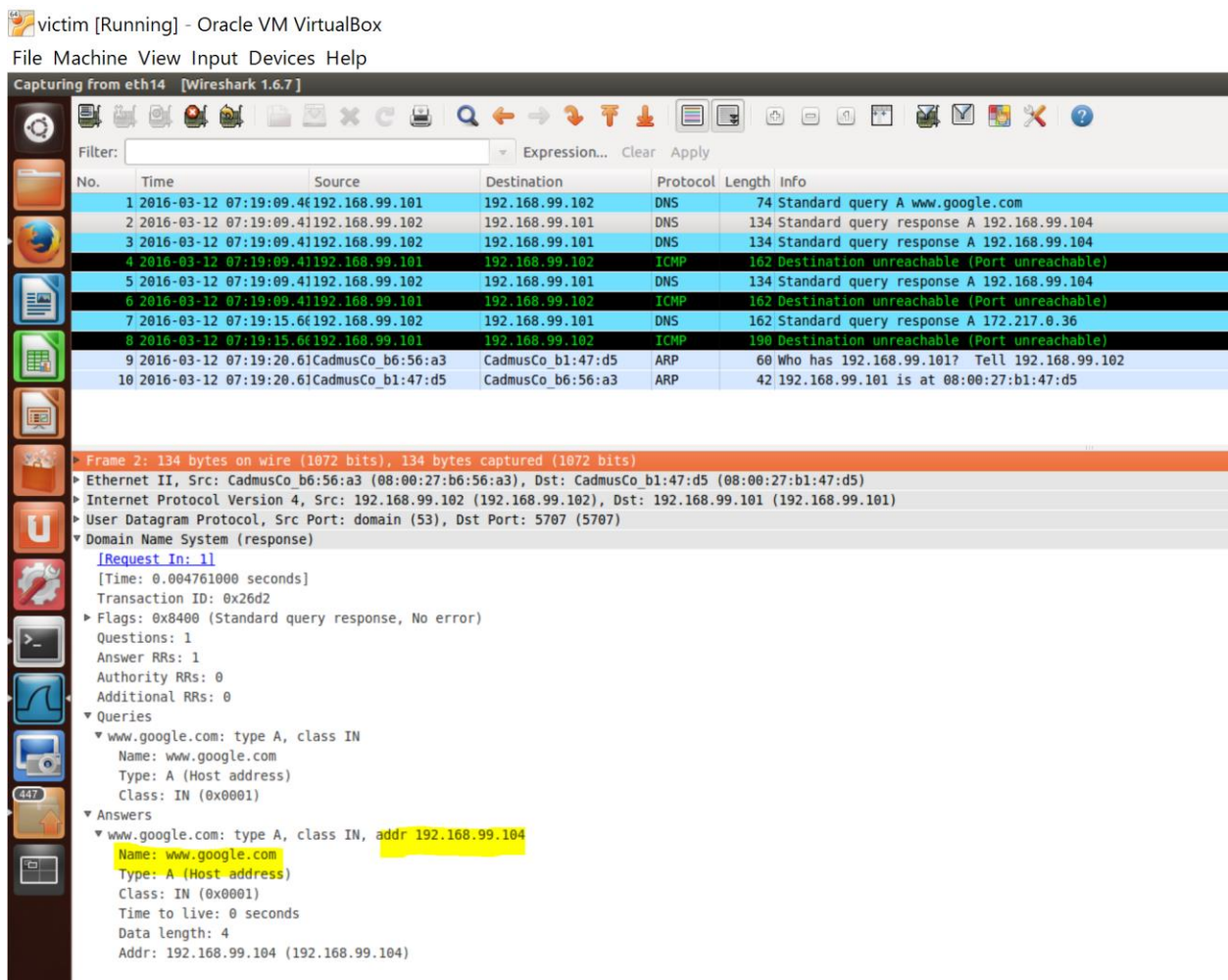
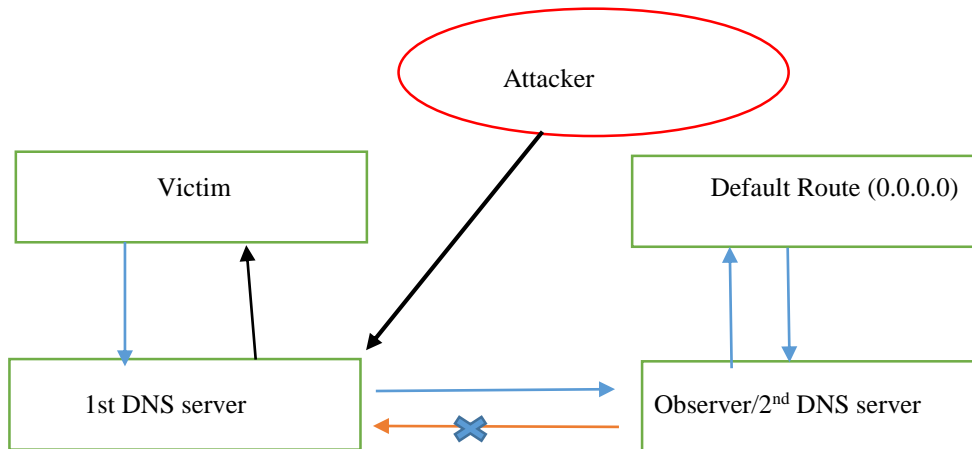


Figure 21: Wireshark capture of spoofed packet

Task 3: DNS Cache poisoning

DNS cache poisoning involving poisoning response from one DNS server to another. It tries to send a spoofed response to the 1st DNS server pretending to be 2nd DNS server.



For seeing immediate effects of this attack, the cache needs to be cleared as in task2. Now any website, let's say facebook.com is queried by the victim, the attacker sends a spoofed response to first DNS server pretending to be second DNS server. The First DNS server accepts this response and forwards the spoofed response to victim after replacing it's layer 2, 3 and 4 headers.

Before Attack

```

[03/12/2016 07:59] seed@ubuntu:~$ dig facebook.com
; <<>> DiG 9.8.1-P1 <<>> facebook.com
;; global options: +cmd
;; Got answer:
;; ->HEADER<- opcode: QUERY, status: NOERROR, id: 5413
;; flags: qr rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 2, ADDITIONAL: 0
;; QUESTION SECTION:
;facebook.com.                IN      A
;; ANSWER SECTION:
facebook.com.                 300     IN      A      66.220.158.68
;; AUTHORITY SECTION:
facebook.com.                 172782  IN      NS      b.ns.facebook.com.
facebook.com.                 172782  IN      NS      a.ns.facebook.com.
;; Query time: 3320 msec
;; SERVER: 127.0.0.1#53(127.0.0.1)
;; WHEN: Sat Mar 12 07:59:18 2016
;; MSG SIZE rcvd: 81
[03/12/2016 07:59] seed@ubuntu:~$
  
```

Figure 22: Query for facebook.com before attack

Before attack or when there is no attack, when the query is done by victim, it returns the original web servers for www.google.com.

After Attack

After attack any query to any site leads to attackerwebserver.com.



Figure 23: After attack, the victim is redirected to attacker's web site

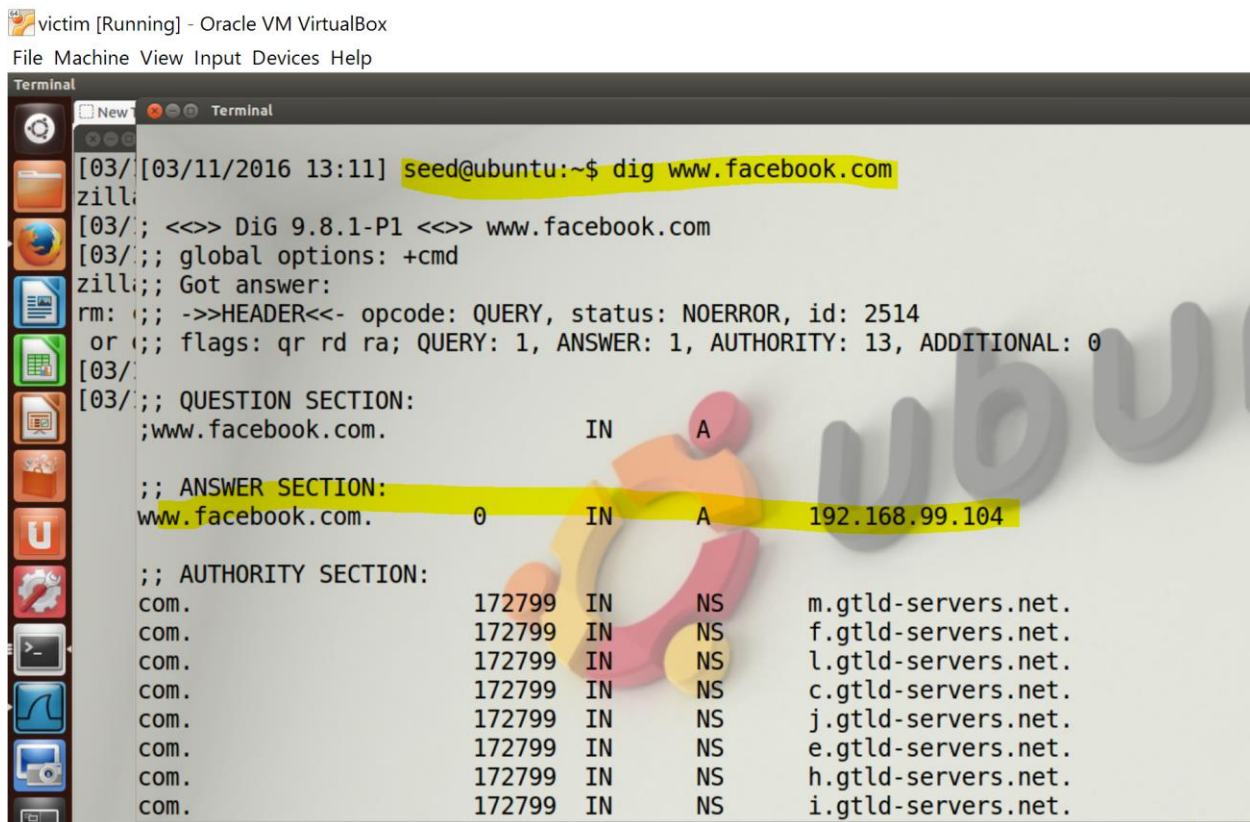


Figure 24: Dig Query returns poisoned result

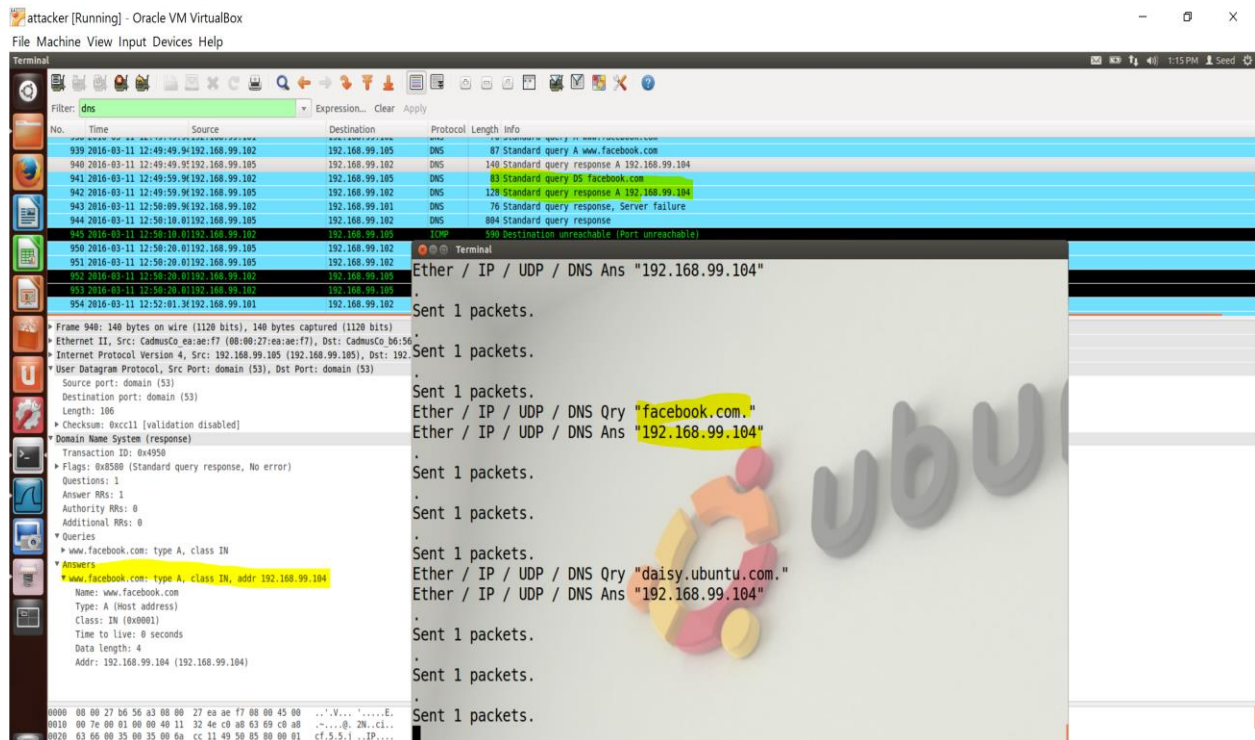


Figure 25: Attacker sending spoofed packet for facebook.com