



中南大学  
CENTRAL SOUTH UNIVERSITY

网络安全  
Seed Project 实验报告  
——本地 DNS 攻击实验

学生姓名：于澜

指导老师：王伟平

专业班级：信安 1401 班

学院：信息科学与工程学院

日期：2016 年 11 月

# 目录:

一、 实验目的.....	3
二、 实验环境.....	3
三、 实验步骤.....	4
1) 安装并配置 DNS 服务器 .....	4
2) 配置用户机器 .....	7
3) 配置攻击机 .....	8
4) 预期产出 .....	8
5) 安装 Wireshark.....	8
四、 实验原理.....	8

# 本地 DNS 攻击实验

## 一、实验目的

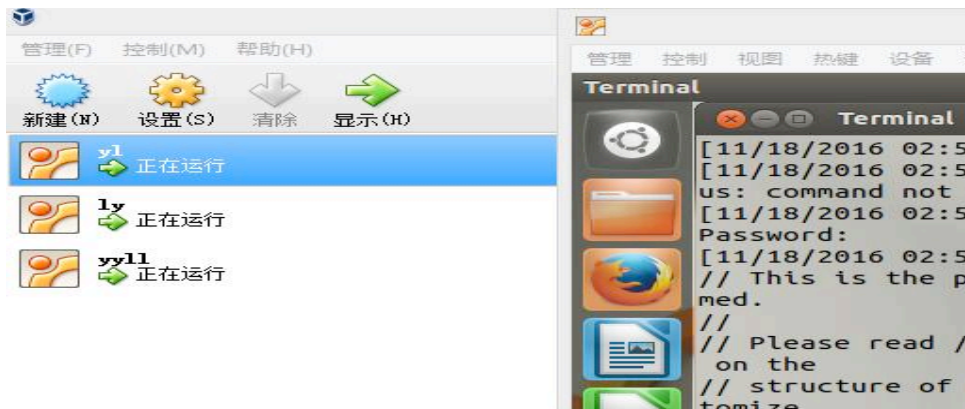
DNS（域名系统）是互联网的电话簿；它将主机名转换为 IP 地址（或 IP 地址到主机名）。这种翻译是通过 DNS 解析，发生在场景后面。DNS Pharming 攻击以各种方式操纵这个解析过程，意图误导用户到其他目的地，这通常是恶意的。本实验的目的是了解这种情况的攻击工作。

## 二、实验环境

需要建立实验室环境像。为了简化实验室环境，我们让用户的计算机，DNS 服务器和攻击者的计算机在一台物理机上，但使用不同的虚拟机。本实验中使用的网站可以是任何网站。配置是基于 Ubuntu，这是在预构建的虚拟机中使用的操作系统。设置了 DNS 服务器，用户机和攻击机同一个局域网。我测试三台虚拟机的 ip 分别为 10.0.2.4，10.0.2.5，10.0.2.6。

1. 使用虚拟机软件。
2. 使用 Wireshark, Netwag 和 Netwox 工具。
3. 配置 DNS 服务器。

三台虚拟机如图



### 三、实验步骤

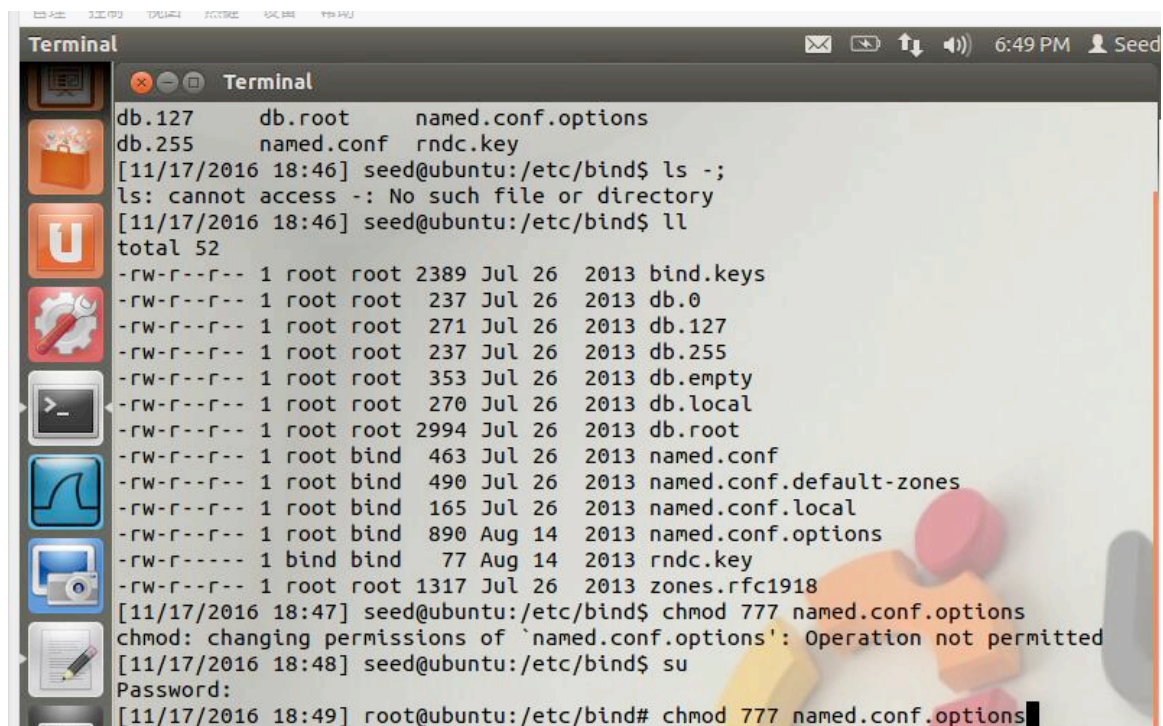
#### 1) 安装并配置 DNS 服务器

##### 1. 装载并启动 bind9

```
[11/17/2016 18:35] root@ubuntu:/home/seed# service bind9 start
* Starting domain name service... bind9 [ OK ]
```

##### 2. 创建 named.conf.options 文件

一开始我一直失败，后来才知道在这之前要获取权限，看一下哪些文件是可以被修改的。

A terminal window titled 'Terminal' showing a user's attempt to list files in /etc/bind. The user runs 'ls -l' and 'll', both of which fail with the message 'ls: cannot access -: No such file or directory'. Then, the user runs 'ls -l' again, which succeeds and shows a list of files and their permissions. The file 'named.conf.options' is highlighted in red. The user then runs 'chmod 777 named.conf.options', which fails with the message 'chmod: changing permissions of 'named.conf.options': Operation not permitted'. Finally, the user runs 'su' to become root, and then runs 'chmod 777 named.conf.options' again, which succeeds.

```
Terminal
db.127 db.root named.conf.options
db.255 named.conf rndc.key
[11/17/2016 18:46] seed@ubuntu:/etc/bind$ ls -l;
ls: cannot access -: No such file or directory
[11/17/2016 18:46] seed@ubuntu:/etc/bind$ ll
total 52
-rw-r--r-- 1 root root 2389 Jul 26 2013 bind.keys
-rw-r--r-- 1 root root 237 Jul 26 2013 db.0
-rw-r--r-- 1 root root 271 Jul 26 2013 db.127
-rw-r--r-- 1 root root 237 Jul 26 2013 db.255
-rw-r--r-- 1 root root 353 Jul 26 2013 db.empty
-rw-r--r-- 1 root root 270 Jul 26 2013 db.local
-rw-r--r-- 1 root root 2994 Jul 26 2013 db.root
-rw-r--r-- 1 root bind 463 Jul 26 2013 named.conf
-rw-r--r-- 1 root bind 490 Jul 26 2013 named.conf.default-zones
-rw-r--r-- 1 root bind 165 Jul 26 2013 named.conf.local
-rw-r--r-- 1 root bind 890 Aug 14 2013 named.conf.options
-rw-r--r-- 1 bind bind 77 Aug 14 2013 rndc.key
-rw-r--r-- 1 root root 1317 Jul 26 2013 zones.rfc1918
[11/17/2016 18:47] seed@ubuntu:/etc/bind$ chmod 777 named.conf.options
chmod: changing permissions of 'named.conf.options': Operation not permitted
[11/17/2016 18:48] seed@ubuntu:/etc/bind$ su
Password:
[11/17/2016 18:49] root@ubuntu:/etc/bind# chmod 777 named.conf.options
```

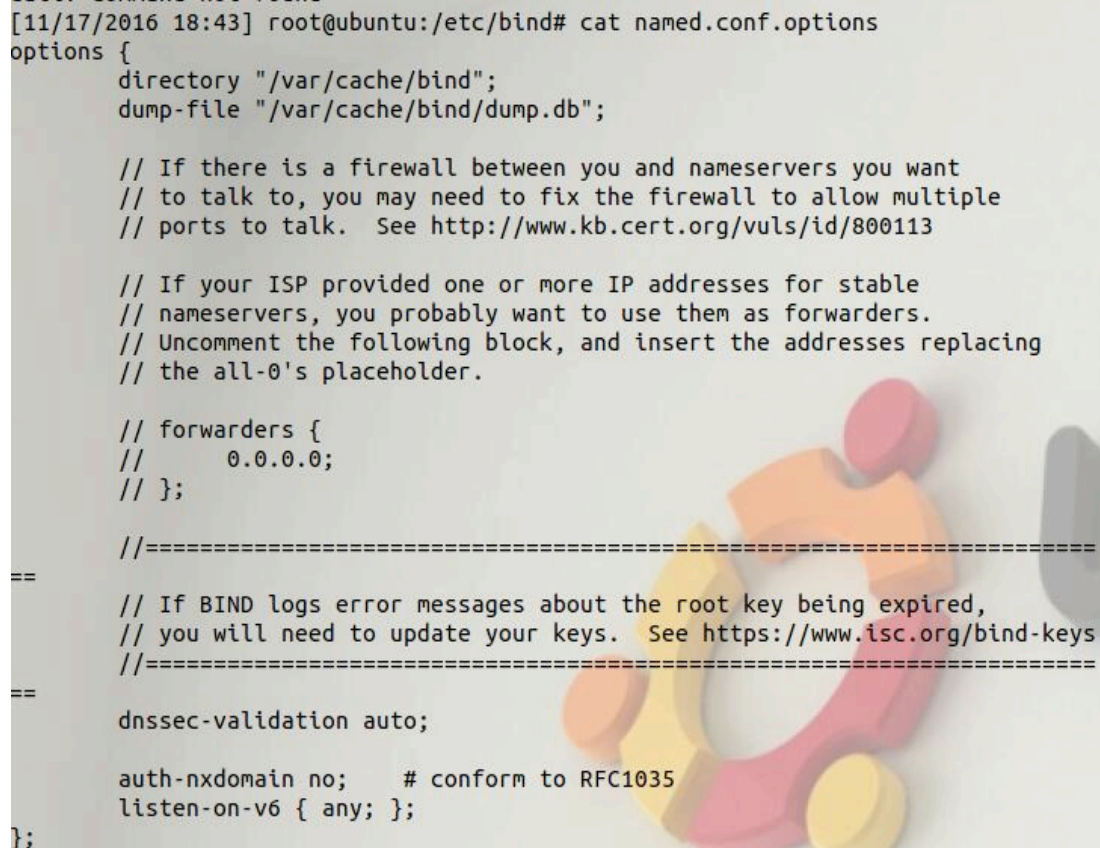
然后对 named.conf.options 文件编辑

```
[11/17/2016 18:35] root@ubuntu:/home/seed# service bind9 start
* Starting domain name service... bind9 [ OK ]
[11/17/2016 18:40] root@ubuntu:/home/seed# vi /etc/bind
[11/17/2016 18:41] root@ubuntu:/home/seed# cd /etc/bind
[11/17/2016 18:41] root@ubuntu:/etc/bind# ls
bind.keys db.empty named.conf.default-zones zones.rfc1918
db.0 db.local named.conf.local
db.127 db.root named.conf.options
db.255 named.conf rndc.key
[11/17/2016 18:41] root@ubuntu:/etc/bind# vi named.conf.options
[11/17/2016 18:43] root@ubuntu:/etc/bind# cat named.conf.options
```

添加以下内容到文件

```
options {  
    dump-file  
        “/var/cache/bind/dump.db” ;  
};
```

添加后用 cat 指令查看一下：

A terminal window screenshot showing the command 'cat /etc/bind/named.conf.options' being executed. The output displays the configuration for the BIND options file, including directory paths, firewall instructions, forwarders, and DNSSEC settings. The background of the terminal window features a colorful, abstract graphic of interlocking puzzle pieces in shades of orange, yellow, and pink.

```
[11/17/2016 18:43] root@ubuntu:/etc/bind# cat named.conf.options  
options {  
    directory "/var/cache/bind";  
    dump-file "/var/cache/bind/dump.db";  
  
    // 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 auto;  
  
    auth-nxdomain no;    # conform to RFC1035  
    listen-on-v6 { any; };  
};
```

注意：文件/var/cache/bind/dump.db 用于转储 DNS 服务器的缓存。

3、DNS 服务器通过添加以下内容到/etc/bind/named.conf

```
zone “example.com” {  
    type master;  
    file “/var/cache/bind/example.com.db” ;  
};  
zone “0.0.10.in-addr.arpa” {  
    type master;  
    file “/var/cache/bind/10.0.2” }
```

注意 ip 段不同，添加的内容也不同

4、设置区域文件。



上述区域中的 file 关键字后面的文件名称为区域文件。实际的 DNS 解析被放在区域文件中。

(1) 在 /var/cache/bind/ 目录，撰写下面的 example.com.db 区域文件  
注意：

配置文件可以从实验室的网页下载，输入这些文件可能会引入错误。

```
[11/18/2016 04:07] root@ubuntu:/var/cache/bind# touch example.com.db
[11/18/2016 04:07] root@ubuntu:/var/cache/bind# vi example.com.db
[11/18/2016 04:08] root@ubuntu:/var/cache/bind# cat example.com.db
$TTL 86400
@ IN SOA ns.example.com. admin.example.com. (
    2008111001 ;serial, today's date + today's serial number
    8H         ;refresh, seconds
    2H         ;retry, seconds
    4W         ;expire, seconds
    1D)        ;minimum, seconds

@ IN NS ns.example.com. ;Address of name server
@ IN MX 10 mail.example.com. ;Primary Mail Exchanger

www IN A 10.0.2.7 ;Address of www.example.com
mail IN A 10.0.2.8 ;Address of mail.example.com
ns IN A 10.0.2.4 ;Address of ns.example.com
*.example.com. IN A 10.0.3.6 ;Address for other URL in
                           ;example.com. domain

[11/18/2016 04:08] root@ubuntu:/var/cache/bind# cd /etc/bind
[11/18/2016 04:10] root@ubuntu:/etc/bind# rm example.com.db
```

符号 “@” 是一个特殊符号，表示来自 named.conf 的源。因此，’@’ 在这里代表 example.com。“IN” 是指互联网。“SOA” 是开始权限的缩写。此区域文件包含 7 个资源记录 (RR)：SOA（开始权限）RR，NS（名称服务器）RR，MX（邮件 eXchanger）RR 和 4 A（主机地址）RR。

(2) 我们还需要设置 DNS 反向查找文件。 在目录 / var / cache / bind / 中，反向 DNS 查找文件名为 10.0.2 的 example.com 域。

```

$TTL 3D
@      IN      SOA      ns.example.com. admin.example.com. (
                        2008111001
                        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.

```

找了半天虚拟机的编辑截图，找不到了……哭泣

5、启动 DNS 服务器。

```

[11/18/2016 04:18] root@ubuntu:~# sudo /etc/init.d/bind9 restart
* Stopping domain name service... bind9
waiting for pid 855 to die
[ OK ]
* Starting domain name service... bind9
[ OK ]

```

2) 配置用户机器

1、在用户计算机上，需要让机器 10.0.2.4 成为默认 DNS 服务器。通过更改用户计算机的 DNS 设置文件/etc/resolv.conf 来实现这一点：

```

[11/18/2016 04:26] seed@ubuntu:/etc$ su
Password:
[11/18/2016 04:26] root@ubuntu:/etc# cat resolv.conf
# Dynamic resolv.conf(5) file for glibc resolver(3) generated by resolv
conf(8)
#      DO NOT EDIT THIS FILE BY HAND -- YOUR CHANGES WILL BE OVERWRITTEN
nameserver 127.0.0.1
[11/18/2016 04:27] root@ubuntu:/etc# vi resolv.conf
[11/18/2016 04:28] root@ubuntu:/etc# cat resolv.conf
# Dynamic resolv.conf(5) file for glibc resolver(3) generated by resolv
conf(8)
#      DO NOT EDIT THIS FILE BY HAND -- YOUR CHANGES WILL BE OVERWRITTEN
nameserver 10.0.2.4

```

注意：确保这是/etc/resolv.conf 中唯一的名称服务器条目。还要注意，在 Ubuntu，/etc/resolv.conf 可能被 DHCP 客户端覆盖。为了避免这种情况，请禁用 DHCP

Click "System Settings" -> "Network",  
Click "Options" in "Wired" Tab,  
Select "IPv4 Settings" -> "Method" -> "Automatic(DHCP) Addresses Only"  
and update only "DNS Servers" entry with IP address of BIND DNS Server.

Now Click the "Network Icon" on the top right corner and Select  
"Auto eth0". This will refresh the wired network connection and  
updates the changes.

然后重新启动 Ubuntu 计算机以使修改的设置生效。

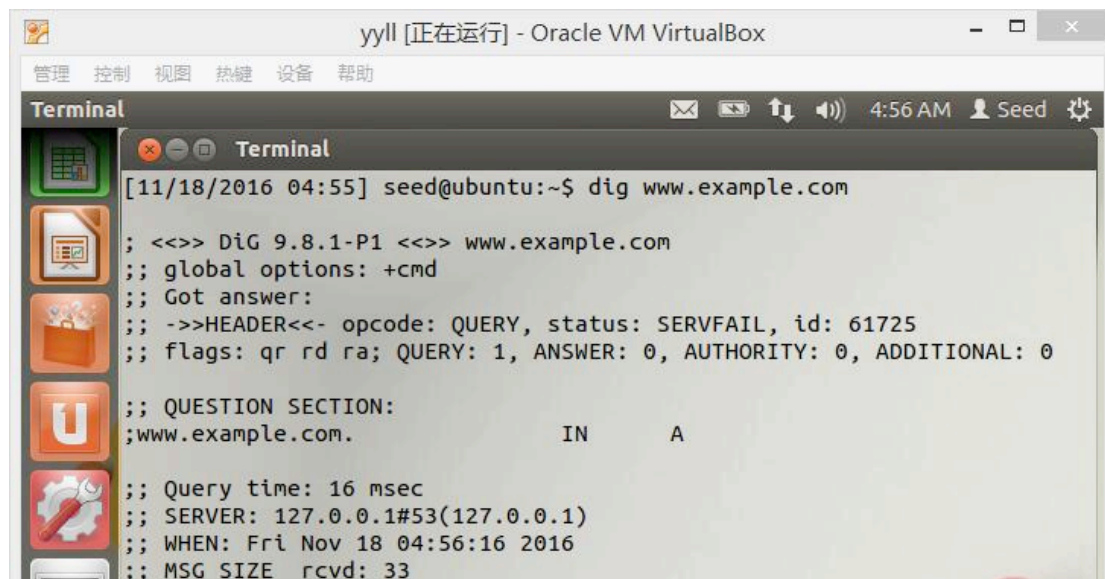
### 3) 配置攻击机

在攻击者机器上，没有太多配置。

### 4) 预期产出

根据上述步骤设置实验室环境后，DNS 服务器已准备就绪。在用户计算机上，发出以下命令：`%dig www.example.com`

运行结果



```
yyll [正在运行] - Oracle VM VirtualBox
管理 控制 视图 热键 设备 帮助
Terminal
[11/18/2016 04:55] seed@ubuntu:~$ dig www.example.com
; <<>> DiG 9.8.1-P1 <<>> www.example.com
;; global options: +cmd
;; Got answer:
;; ->HEADER<- opcode: QUERY, status: SERVFAIL, id: 61725
;; flags: qr rd ra; QUERY: 1, ANSWER: 0, AUTHORITY: 0, ADDITIONAL: 0
;; QUESTION SECTION:
;www.example.com.                IN      A
;; Query time: 16 msec
;; SERVER: 127.0.0.1#53(127.0.0.1)
;; WHEN: Fri Nov 18 04:56:16 2016
;; MSG SIZE rcvd: 33
```

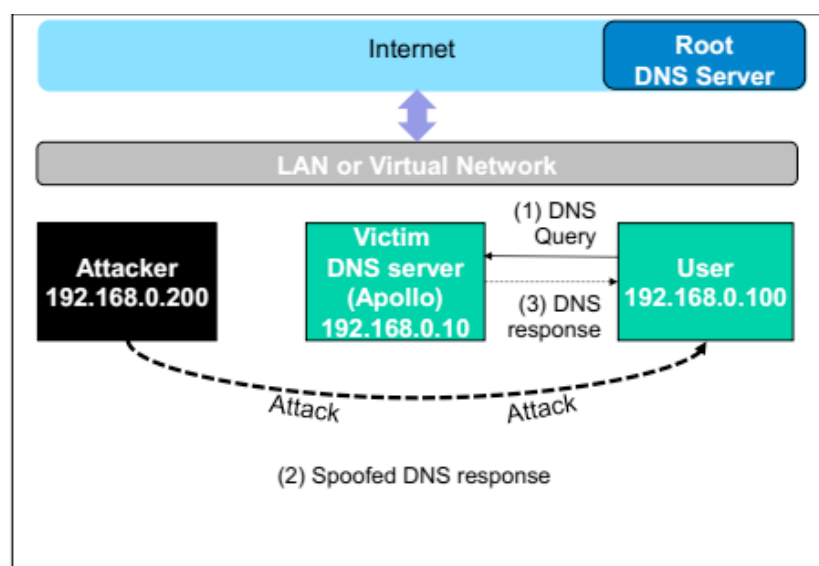
### 5) 安装 Wireshark

Wireshark 是这个实验室非常重要的工具，可以嗅探每一个经历的包裹 LAN。

## 四、实验原理



当一个 DNS 服务器 Apollo 收到一个查询时，如果主机名不在 Apollo 的域内，它将会请求其他 DNS 服务器获取主机名解析。注意，在我们的实验室设置中，我们的 DNS 域服务器是 example.com； 因此，对于其他域（例如 www.google.com）的 DNS 查询，DNS 服务器 Apollo 将询问其他 DNS 服务器。然而，在询问其他 DNS 服务器之前，它首先从自己的缓存中寻找答案；如果答案是肯定的，DNS 服务器会简单地回复与来自其缓存的信息。如果答案不在缓存中，DNS 服务器将尝试获取答案从其他 DNS 服务器。当 Apollo 得到答案时，它会将答案存储在缓存中，所以接下来没有必要问其他 DNS 服务器。因此，如果攻击者可以欺骗来自其他 DNS 服务器的响应，Apollo 将保留欺骗响应在其缓存中一段时间。下一次，当用户的机器想要解决相同的主机名，Apollo 将使用在缓存中的欺骗响应来回复。这样，攻击者只需要一次欺骗，并且影响将持续直到缓存的信息过期。这种攻击称为 DNS 缓存中毒。



上下是两种不同方式

