

网络安全实验报告

题 目 Local DNS Attack

学生姓名 龙思怡

指导教师 王伟平

学 院 信息科学与工程学院

专业班级 信安 1401

学 号 0906140104

二〇一六 年 十 一 月 十 九 日

一、问题描述

DNS (Domain Name System, 域名系统), 因特网上作为域名和IP地址相互映射的一个分布式数据库, 能够使用户更方便的访问互联网, 而不用去记住能够被机器直接读取的IP数串。通过主机名, 最终得到该主机名对应的IP地址的过程叫做域名解析(或主机名解析)。DNS协议运行在UDP协议之上, 使用端口号53。主机名到IP地址的映射有两种方式:

- 1) 静态映射, 每台设备上配置主机到 IP 地址的映射, 各设备独立维护自己的映射表, 而且只供本设备使用;
- 2) 动态映射, 建立一套域名解析系统(DNS), 只在专门的 DNS 服务器上配置主机到 IP 地址的映射, 网络上需要使用主机名通信的设备, 首先需要到 DNS 服务器查询主机所对应的 IP 地址。

通过主机名, 最终得到该主机名对应的 IP 地址的过程叫做域名解析(或主机名解析)。在解析域名时, 可以首先采用静态域名解析的方法, 如果静态域名解析不成功, 再采用动态域名解析的方法。可以将一些常用的域名放入静态域名解析表中, 这样可以大大提高域名解析效率。

一般来说, 浏览器都是通过将两种方式结合的办法来提高用户体验的, 所以DNS攻击其实可以分为两种, 一种是针对静态映射的攻击, 即直接修改用户本地的hosts文件, 这样在浏览器解析地址的时候就不会查询DNS服务器而是直接根据文件中存取的对应的IP地址来进行访问, 达到了攻击的目的。而另外一种方式则是直接攻击用户机器或者服务器来达到目的, 主要是通过发送假的response包来进行欺骗。

所以在日常生活中, 如果可以冒充域名服务器, 然后把查询的IP地址设为攻击者的IP地址, 这样的话, 用户上网就只能看到攻击者的主页, 而不是用户想要取得的网站的主页了, 这就是DNS欺骗的基本原理。DNS欺骗其实并不是真的“黑掉”了对方的网站, 而是冒名顶替、招摇撞骗罢了。

现在的Internet上存在的DNS服务器有绝大多数都是用bind来架设的, 使用的bind版本主要为bind 4.9.5+P1以前版本和bind 8.2.2-P5以前版本. 这些bind有个共同的特点, 就是BIND会缓存(Cache)所有已经查询过的结果, 这个问题就引起了下面的问题的存在.: 在DNS的缓存还没有过期之前, 如果

在DNS的缓存中已经存在的记录,一旦有客户查询,DNS服务器将会直接返回缓存中的记录.当然,还有更多可能引发的问题,在后面将会继续描述。

二、实验概述

1. 环境配置

为了简化实验环境我们将服务器,攻击者,受攻击者的计算机都放在一台物理机上,但是用不同的虚拟机。本实验中用的网站可以是任意的。我们的配置是基于Ubuntu的。你可以从图1看出,我们建立的三台虚拟机在同一个局域网中,用户的IP地址为192.168.0.100,DNS服务器的IP地址为192.168.0.10,攻击者的IP为192.168.0.200

a. 配置DNS服务器

DNS服务器使用的是bind9软件,在ubuntu14.04上已经下载好了。

b. 创建配置文件

DNS服务器需要读取/etc/bind/named.conf文件来启动DNS服务,而named.conf.options就是一个被包含的文件,而我们需要将这句话加入到named.conf.options文件中:

```
/var/cache/bind/dump.db
```

这是为了让DNS服务器使用dump.db作为它的缓存。

c. 创建zone文件

把ZONE 文件拿出来简单说明一下。ZONE 文件是DNS 上保存域名配置的文件,对

BIND 来说一个域名对应一个ZONE 文件,现以abc.com 的ZONE 文件为例展开。

(罗嗦一句,ZONE 存在于权威DNS 上)

```
=====+=====+=====+=====
=====
```

```
$TTL 6h //第1 行
```

```
$ORIGIN abc.com. //第2 行
```

```
@ 3600 IN SOA ns1.ddd.com. root.ddd.com. ( //第3 行
```

```

929142851 ; Serial //第4 行
1800 ; Refresh //第5 行
600 ; Retry //第6 行
2w ; Expire //第7 行
300 ; Minimum //第8 行)
@ 2d IN NS ns1.ddd.com. //第9 行
@ 2d IN NS ns2.ddd.com. //第10 行
@ 2d IN NS ns3.ddd.com. //第11 行
@ 3600 IN A 120.172.234.27 //第12 行
a 3600 IN A 120.172.234.27 //第13 行
b 3600 IN CNAME a.abc.com. //第14 行
@ 3600 IN MX a.abc.com. //第15 行
@ 3600 IN TXT "TXT" //第16 行

=====+=====+=====+=====
=====

```

第1 行，这行内容给出了该域名(abc.com)各种记录的默认TTL 值，这里为6 小时。即如

果该域名的记录没有特别定义TTL，则默认TTL 为有效值。

第2 行，这行内容标识出该ZONE 文件是隶属那个域名的，这里为abc.com。

第3 行，从这行开始到第8 行为该域名的SOA 记录部分，这里的@代表域名本身。

ns1.ddd.com 表示该域名的主权威DNS。root.ddd.com 表示该主权威DNS 管理员邮箱，等价于root@ddd.com。

第4 行，Serial 部分，这部分用来标记ZONE 文件更新，如果发生更新则Serial 要单增，否则MASTER 不会通知SLAVE 进行更新。

第5 行，Refresh 部分，这个标记SLAVE 服务器多长时间主动(忽略MASTER 的更新通知)向MASTER 复核Serial 是否有变，如有变则更新之。

第6 行, Retry 部分, 如Refresh 不能完成, 重试的时间间隔。

第7 行, Expire 部分, 如SLAVE 无法与MASTER 取得联系, SLAVE 继续提供DNS 服务的时间, 这里为2W(两周时间)。Expire 时间到期后SLAVE 仍然无法联系MASTER 则停止工作, 拒绝继续提供服务。Expire 的实际意义在于它决定了MASTER 服务器的最长下线时间(如MASTER 迁移, DOWN 机等)。

第8 行, Minimum 部分, 这个部分定义了DNS 对否定回答(NXDOMAIN 即访问的记录在权威DNS 上不存在)的缓存时间。

第9-11 行, 定义了该域名的3 个权威DNS 服务器。通常NS 记录的TTL 大些为宜, 这里为2天。设置过小只会增加服务器无谓的负担, 同时解析稳定性会受影响。

第12-16 行, 比较简单, 是两个A, CNAME, MX 记录, 不再讨论了。

附录FAQ

SOA 记录: 权威记录从这里开始, 它定义了3-8 行这些重要的参数。

A 记录: 域名到IP 之间的关联。

CNAME 记录: 让张三住到李四家里, 这时张三李四是同一个地址。

MX 记录: 定义了发往XXX@ABC.COM 邮箱的邮件服务器地址。

TXT 记录: 这个记录的内容是文本格式如163.COM 的TXT 为“v=spf1 include:spf.163.com -all”, TXT 通常用于邮件服务器来标识自己的身份避免被认为是垃圾邮件服务器

d. 配置zone文件

DNS的核心就存在于zone文件中, zone文件中各个字段的含义在上面都有解释。我们直接从Seed Project上下载对应的zone文件即可。

e. 启动DNS服务器

通过`service start bind9`或者`service restart bind9`即可启动。

f. 配置用户机

用户机的配置主要是将DNS服务器配置成192.168.0.10, 这里的方式是修改/etc/resolv.conf, 将nameserver修改为192.168.0.10.但是这里有个问题就是这个文件有可能被DHCP协议修改成默认的网关。所以我

们还需要在System Setting里的Network中设置IPv4，把DHCP协议改为固定的DNS服务器，但是这里有个问题就是无法联网。

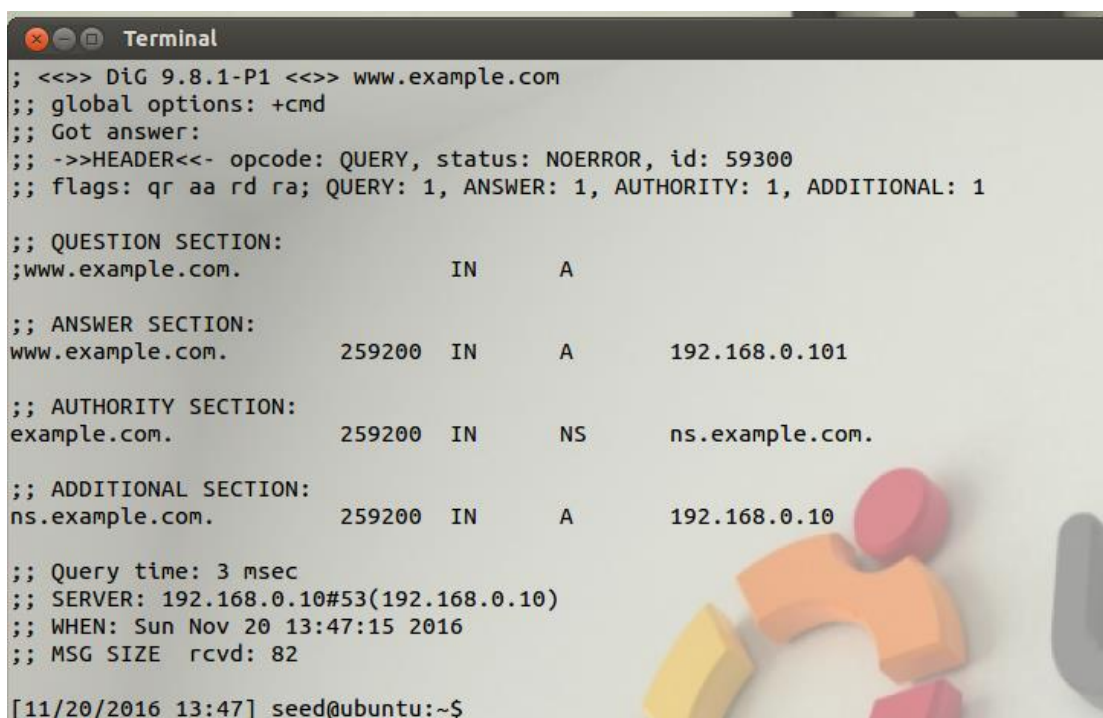
g. 攻击机只需要设置IP地址即可。

2. 任务说明

在经过前面的设置以后，要保证整个DNS查询的过程是畅通的，我们在用户机上输入以下命令进行查询：

```
dig www.example.com
```

这个时候会出现如下的画面：



```
Terminal
; <<>> DiG 9.8.1-P1 <<>> www.example.com
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 59300
;; 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: 192.168.0.10#53(192.168.0.10)
;; WHEN: Sun Nov 20 13:47:15 2016
;; MSG SIZE rcvd: 82

[11/20/2016 13:47] seed@ubuntu:~$
```

你会发现返回的IP地址为你在DNS服务器中设定好了的192.168.0.101.

接下来就要开始我们这个实验的任务了，第一个任务是本地DNS攻击，在这个实验中我们主要是通过远程ssh已经被攻破的用户机器来修改hosts文件达到把用户引导到错误的网址上去的目的。第二个任务是通过向用户机器发送伪造的DNS回应包来达到目的，而第三个实验是通过向DNS服务器发送欺骗包来达到DNS攻击的目的。

对用户实施网域嫁接攻击的主要目的是当用户使用A的域名想要访

问机器A时，将用户重定向到另一台机器B。比如，当用户试图登陆网上银行时，比如www.chase.com，如果攻击者可以把用户定向到一个和正常网站很像的恶意网站，那么用户可能会被误导而提交自己网上银行的账号和密码。

当用户在浏览器中输入www.chase.com时，用户的机器会访问一个DNS队列来找出这个域名对应的IP地址。攻击者的目标就是用一个虚假的DNS回复欺骗用户机器，也就是把www.chase.com重定向到一个恶意IP地址。在实验中，我们会以www.example.com作为用户想要登陆的网站作为实例。

三、实验流程

Task 1: 修改HOSTS文件

Hosts文件是一个用于存储计算机网络中节点信息的文件，它可以将主机名映射到相应的IP地址，实现DNS的功能，它可以由计算机的用户进行控制。Hosts文件的存储位置在不同的操作系统中并不相同，甚至不同Windows版本的位置也不大一样：Windows NT/2000/XP/2003/Vista/win7：默认位置为%SystemRoot%\system32\drivers\etc\，但也可以改变。Linux下一般放在/etc/hosts下。

有很多网站不经过用户同意就将各种各样的插件安装到你的计算机中，其中有些说不定就是木马或病毒。对于这些网站我们可以利用Hosts把该网站的域名映射到错误的IP或本地计算机的IP，这样就不能访问了。在大多数系统中，约定127.0.0.1为本地计算机的IP地址，0.0.0.0是错误的IP地址。

如果，我们在Hosts中，写入以下内容：

127.0.0.1 # 要屏蔽的网站 A

0.0.0.0 # 要屏蔽的网站 B

这样，计算机解析域名 A和 B时，就解析到本机IP或错误的IP，达到了屏蔽网站A 和B的目的。

我们进行本地攻击时也是利用相同的原理，主机名和IP地址以成对的

方式存放在HOSTS文件中，它用于本地查找，而不是远程的DNS查找。比如，如果用户机器里有一个这样的HOSTS文件，www.example.com将会被解释成IP地址为1.2.3.4的主机，而不会询问任何的DNS服务器。

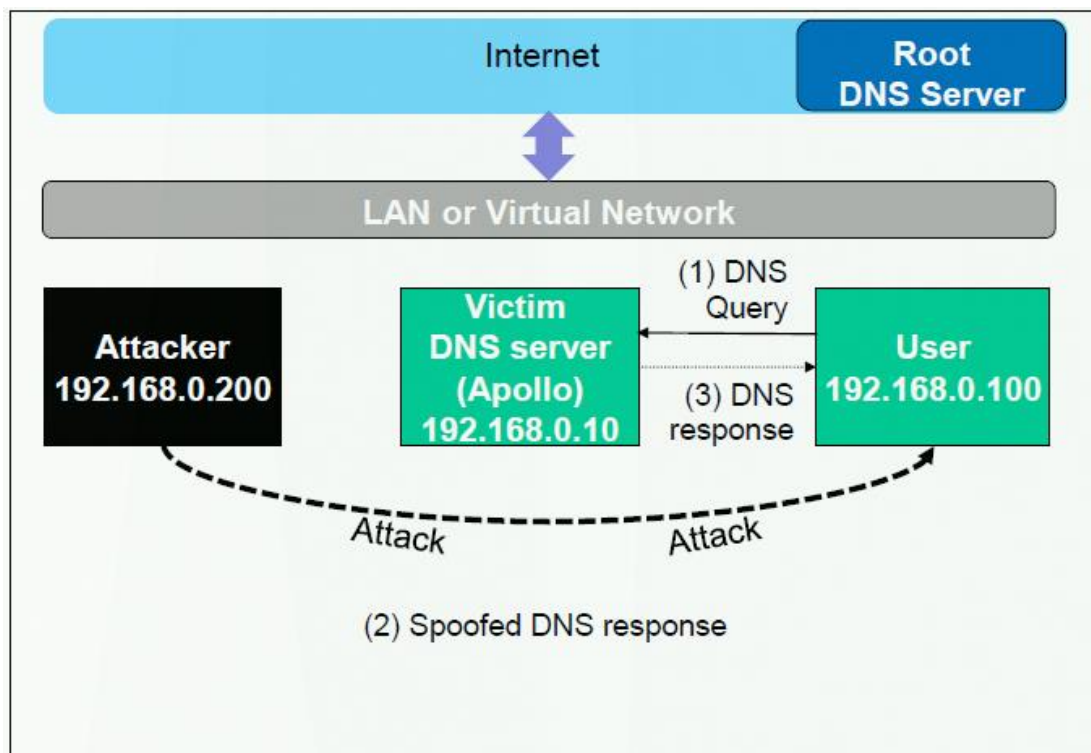
当然了，ping命令和dig命令是有所区别的，这个我们在结果分析中在进行讲述。

Task 2: 本机DNS劫持-对用户的DNS查询欺骗

在这种攻击中，受害者机器没有直接被攻破，所以攻击者不能直接改变DNS列表进程。然而，如果攻击者和受害者在同一网段，他们仍然可以造成很大的危害。当一个用户输入网站的域名时，用户机器将会询问DNS服务器来获取一个IP地址。在侦听到DNS服务器的回应后，攻击者可以创建一个假的DNS回应。假的DNS回应如果具有与真的DNS回应相同的格式将会被用户机器接受。

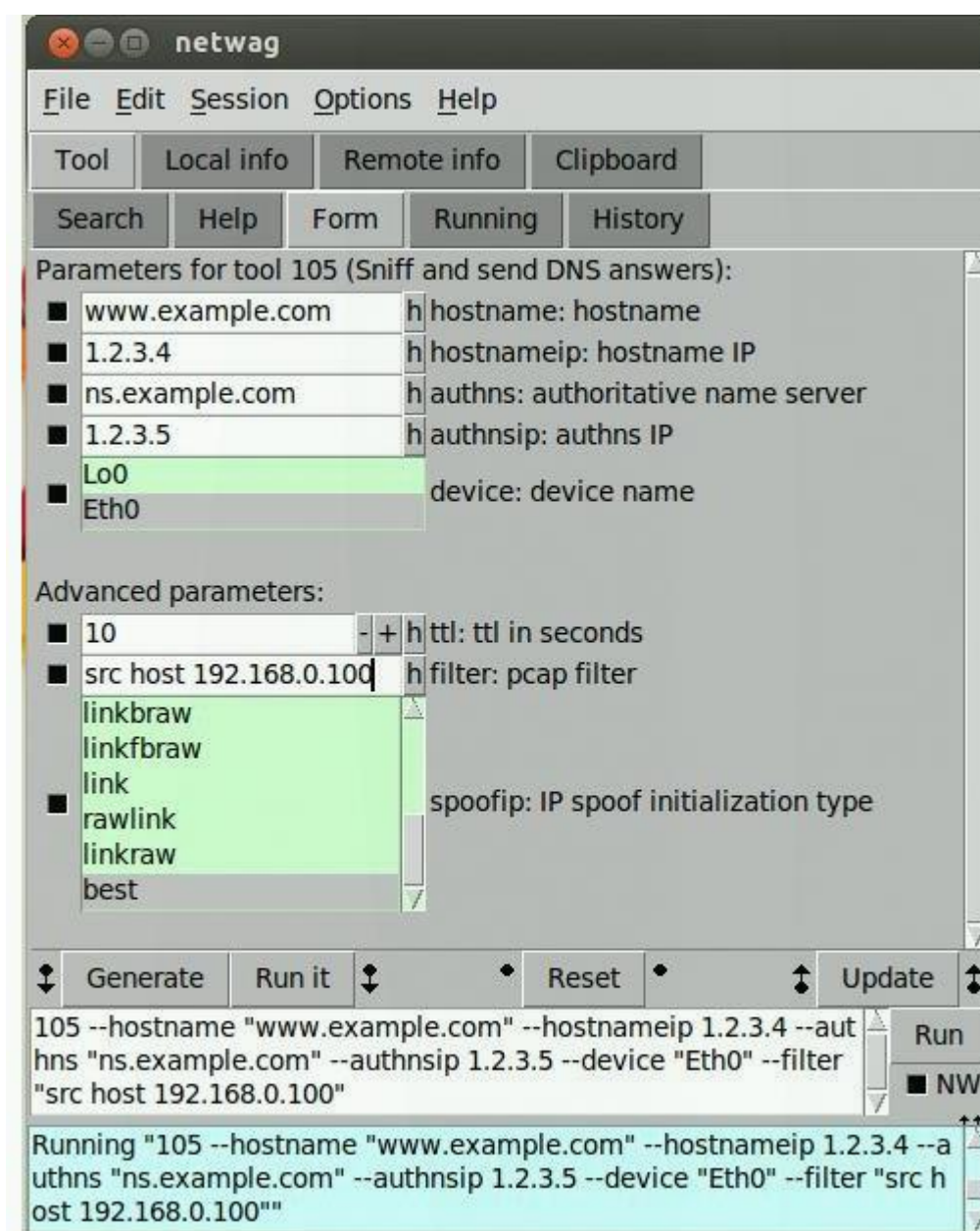
为了满足标准1-8，攻击者需要截取用户的DNS请求，并且在DNS发出回应之前给用户发送DNS请求回应。可以利用Netwox/Netwag来监听DNS服务器的请求与回应。

整个过程如下图所示：



当User向DNS Server发送DNS查询的时候，Attacker监听了这个DNS查询请求，然后在DNS Server回复正确的DNS Response之前，先回复一个伪造欺骗的DNS Response给用户，从而达到了DNS欺骗的效果。

实验中我们借用了Netwox/Netwag tool 105来进行DNS欺骗，具体的设置如下：



攻击机根据上图进行设置，然后在用户机器上再次运行dig www.example.com的命令，就可以看到如下的结果了：

```
Terminal
[11/20/2016 14:36] root@ubuntu:/home/seed# dig www.example.com

; <<> DiG 9.8.1-P1 <<> www.example.com
;; global options: +cmd
;; Got answer:
;; ->HEADER<- opcode: QUERY, status: NOERROR, id: 26742
;; 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.                600     IN      A      1.2.3.4

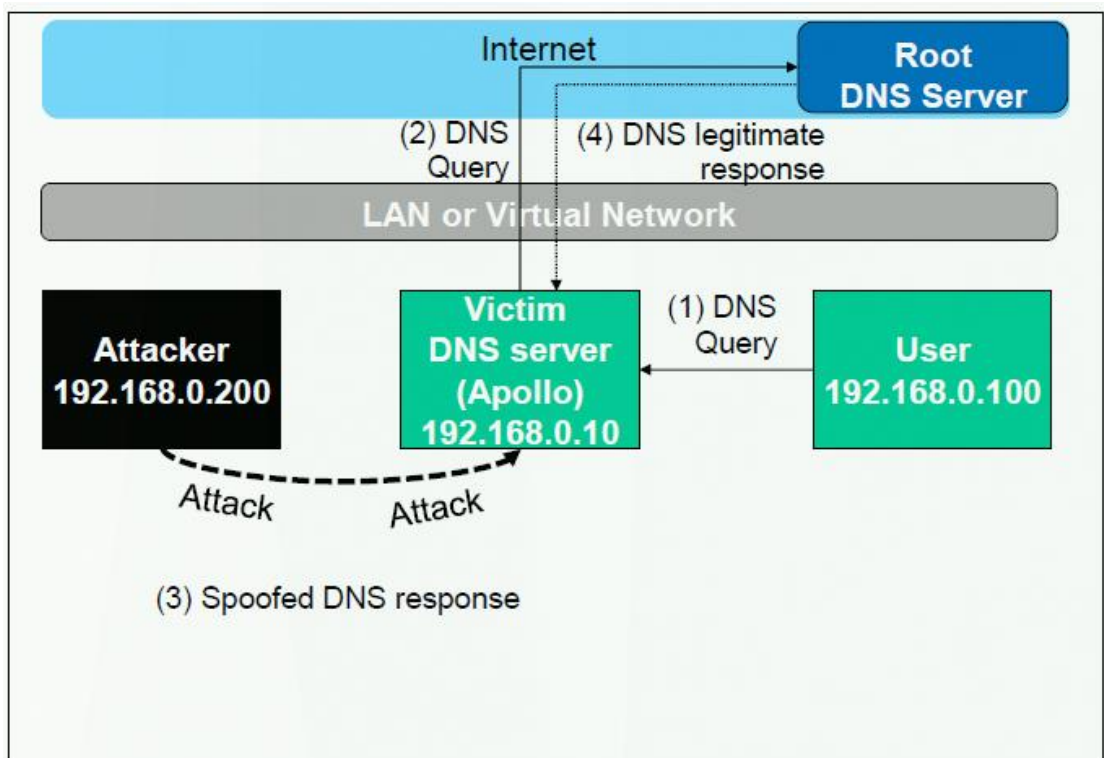
;; AUTHORITY SECTION:
ns.example.com.                 600     IN      NS      ns.example.com.

;; ADDITIONAL SECTION:
ns.example.com.                 600     IN      A      1.2.3.5

;; Query time: 6 msec
;; SERVER: 192.168.0.10#53(192.168.0.10)
;; WHEN: Sun Nov 20 14:36:22 2016
;; MSG SIZE rcvd: 88
```

Task 3: 本机DNS劫持-对服务器的DNS欺骗

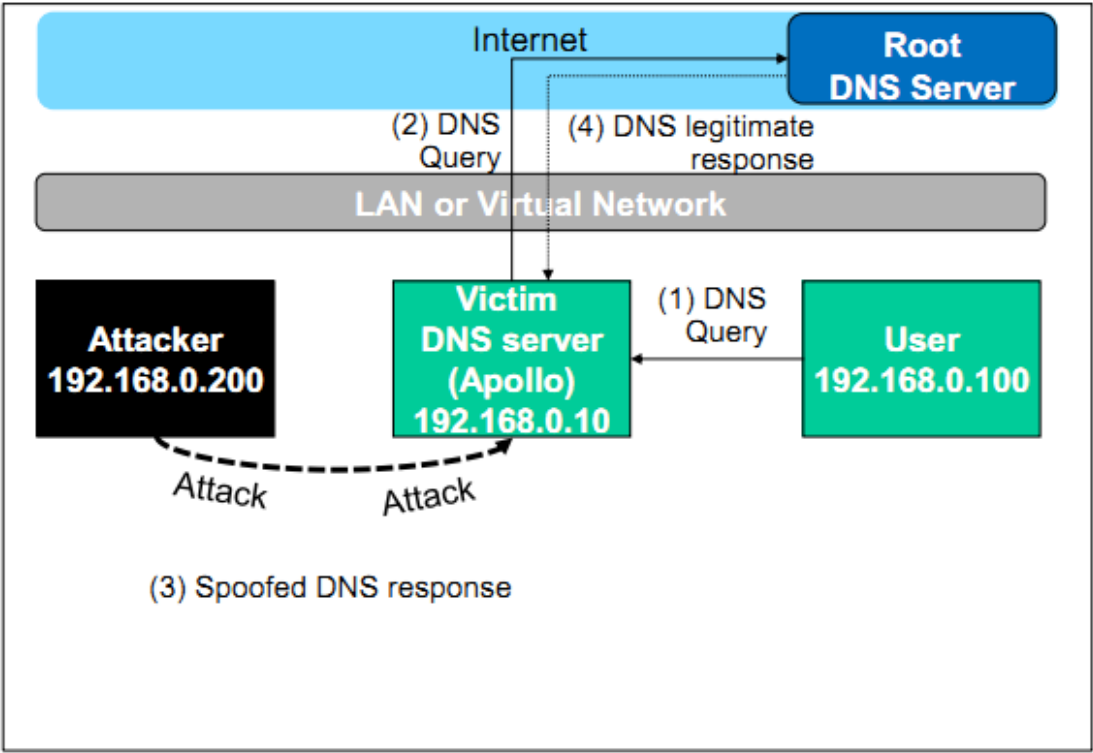
对于服务器的欺骗过程如下图所示:



在DNS的缓存还没有过期之前, 如果在DNS的缓存中已经存在的记录, 一

一旦有客户查询, DNS服务器将会直接返回缓存中的记录。

上述的攻击都是针对于用户机器, 为了达到保存时间长的目的, 每一次用户机器发送DNS问询时, 攻击者都需要发送一个捏造的DNS回应, 这效率会很低。现在有一个更好的方法, 即攻击DNS服务器, 而不是用户机器。



四、结果分析

Task 1: 修改HOSTS文件结果分析

用户本来的hosts文件中, www.example.com对应的是127.0.0.1, 结果如下图所示:

```
;; 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
```

ping命令的结果:

```
[11/20/2016 14:14] root@ubuntu:/home/seed# ping www.example.com
PING www.example.com (127.0.0.1) 56(84) bytes of data.
64 bytes from localhost (127.0.0.1): icmp_req=1 ttl=64 time=0.020 ms
```

修改hosts文件后的Ping结果如下所示:

```
[11/20/2016 15:00] root@ubuntu:/home/seed# ping www.example.com
PING www.example.com (1.2.3.4) 56(84) bytes of data.
```

可以看出, 修改hosts文件会影响ping命令但是不会影响dig命令, 也就是说, dig查询命令依然会问询DNS服务器, 但是Ping命令为了用户体验, 需要快速的查找到域名所对应的IP, 所以会先遍历hosts文件来进行一次查询。

Task 2: 本机DNS劫持-对用户的DNS查询欺骗结果分析

在尝试几次之后, 我们会得到如下的结果:

```
Terminal
[11/20/2016 14:36] root@ubuntu:/home/seed# dig www.example.com

; <<>> DiG 9.8.1-P1 <<>> www.example.com
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 26742
;; 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.                600     IN      A      1.2.3.4

;; AUTHORITY SECTION:
ns.example.com.                 600     IN      NS      ns.example.com.

;; ADDITIONAL SECTION:
ns.example.com.                 600     IN      A      1.2.3.5

;; Query time: 6 msec
;; SERVER: 192.168.0.10#53(192.168.0.10)
;; WHEN: Sun Nov 20 14:36:22 2016
;; MSG SIZE rcvd: 88
```

但是这其中有一个问题, 因为在局域网内, 互相之间的通信速度是非常快的, 所以虽然把TTL设置的很长, 但是有的时候用户机仍然会接受到正确的DNS包, 这里还需要再改进。

Task 3: 本机DNS劫持-对服务器的DNS欺骗结果分析

```
Terminal
[11/20/2016 14:38] root@ubuntu:/home/seed# dig www.example.cn

; <<>> DiG 9.8.1-P1 <<>> www.example.cn
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 65050
;; flags: qr aa rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 1, ADDITIONAL: 1

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

;; ANSWER SECTION:
www.example.cn.                 600     IN      A      1.2.3.4

;; AUTHORITY SECTION:
ns.example.com.                 600     IN      NS      ns.example.com.
```

在对Netwag进行了相应的设置以后，我们达到了攻击服务器的目的。

五、调试报告

1. 原理概述

一、DNS 篡改（DNS Hijacking or DNS Redirection）

DNS 篡改通常是指直接修改根服务器中的 DNS 条目，当然这种篡改更有可能发生在域名注册商的存放客户注册配置信息的服务器中，从而导致相应的域名在全球范围内都解析错误。（有些情况下，也有可能只是在中层的 DNS 服务器作篡改，这样仅仅劫持某部份的域名解析，只会影响部分地区。）

这种攻击方式的特点如下：

全球性。因为根服务器条目被更改，在下游 DNS 缓存时间过后，被篡改条目必然会更新到所有下游服务器，从而影响到所有用户。

通常所说的“技术性”并非必需。在多数情况下，这种攻击方式在DNS攻击阶段甚至不需要涉及技术性，可以借助社会工程学等手段，攻击者通常已经拥有了足够的修改DNS条目的权限。虽然不排除黑客能够通过技术手段入侵根DNS服务器或者域名服务商，在这种情况下，黑客就可以指哪儿打哪儿无法无天了。但相信这种天下大乱的情况我们不一定能有幸见到。更可能的情况是，某个网站运营商的域名的账号密码被窃取了，黑客

拿到账号密码之后，就能像网站管理员一样冠冕堂皇地登陆域名提供商网站，修改相应的IP地址。

二、DNS 欺骗 (DNS Spoofing)

一种 DNS 欺骗方式是利用漏洞，去年上半年，DNS 协议被发现存在严重漏洞，攻击者可以欺骗下游服务器，使其相信一台假冒的服务器是它的权威服务器。这样攻击者就可以通过在假冒服务器上添加虚假的 DNS 信息来欺骗下游服务器，最终欺骗客户端。在 Windows 系统中，这个漏洞已经在稍早的安全补丁发布获得中解决：

<http://www.microsoft.com/china/technet/security/bulletin/ms08-037.msp>

其他操作系统中也可以查询到相应的漏洞。还有一种 DNS欺骗思路，借助其他欺骗达到DNS欺骗的目的，例如先在目标客户端或DNS服务器的同一局域网网段作ARP欺骗，使受攻击的计算机向ARP攻击的发起者作DNS查询。但是这种攻击方式必须先攻破与目标主机同网段的另一台计算机，而互联网中的DNS服务器通常在ISP的控制之下，渗透进ISP网络的难度不小。攻击根服务器的难度就更别提了。

DNS 欺骗攻击方式的特点如下：

区域性。因为这种攻击只能攻击下游服务器，因此攻击的有效范围必然有限，只会影响到此台服务器下层的服务器和客户端。

漏洞利用。要实现 MS08-037 攻击必然需要利用 DNS 协议的漏洞（MS08-037 或者其他可能存在的漏洞）。如果 DNS 服务器上的漏洞已经修复，那么这种攻击就无法成功。

技巧性。不论利用漏洞还是利用 ARP 的设计缺陷，这类攻击方式在 DNS 攻击阶段都需要一定的技术手段。

2. 经验和收获

DNS欺骗攻击是很难防御的，因为这种攻击大多数本质都是被动的。通常情况下，除非发生欺骗攻击，否则你不可能知道你的DNS已经被欺骗，只是你打开的网页与你想要看到的网页有所不同。在很多针对性的攻击中，用户都无法知道自己已经将网上银行帐号信息输入到错误的网址，直到接到银行的电话告知其帐

号已购买某某高价商品时用户才会知道。这就是说，在抵御这种类型攻击方面还是有迹可循。

使用最新版本的DNS服务器软件，并及时安装补丁

关闭DNS服务器的递归功能。DNS服务器利用缓存中的记录信息回答查询请求或是DNS服务器通过查询其他服务获得查询信息并将它发送给客户机，这两种查询成为递归查询，这种查询方式容易导致DNS欺骗。

保护内部设备：像这样的攻击大多数都是从网络内部执行攻击的，如果你的网络设备很安全，那么那些感染的主机就很难向你的设备发动欺骗攻击。

不要依赖DNS：在高度敏感和安全的系统，你通常不会在这些系统上浏览网页，最后不要使用DNS。如果你有软件依赖于主机名来运行，那么可以在设备主机文件里手动指定。

使用入侵检测系统：只要正确部署和配置，使用入侵检测系统就可以检测出大部分形式的ARP缓存中毒攻击和DNS欺骗攻击。

通过这次试验，我不仅明白了各种DNS攻击的原理，更是学会了如何辨别网站何时会出现异常的情况以及如何去进行修补，获益匪浅。