

# 中南大学

## 本地 DNS 攻击

### 实验报告

学生姓名 \_\_\_\_\_ 刘晓悦 \_\_\_\_\_

专业班级 \_\_\_\_\_ 信安 1401 \_\_\_\_\_

学 号 \_\_\_\_\_ 0906140118 \_\_\_\_\_

学 院 \_\_\_\_\_ 信息科学与工程学院 \_\_\_\_\_

指导教师 \_\_\_\_\_ 王伟平 \_\_\_\_\_

实验时间 \_\_\_\_\_ 2016 年 12 月 \_\_\_\_\_

# 本地 DNS 攻击实验

## 1 实验室概述

DNS 是互联网的电话簿;它将主机名转换为 IP 地址。这种翻译是通过 DNS 解析,发生在场景后面。DNS 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,这通常恶意的。本实验的目的是了解这种情况攻击工作。

## 2 实验室环境

我们设置了 DNS 服务器,用户机器和攻击者机器同一个局域网。我们假设用户计算机的 IP 地址是 192.168.226.129,DNS 服务器的 IP 是 192.168.226.130,攻击者的 IP 为 192.168.226.128。

### 2.1 安装并配置 DNS 服务器

步骤 1: 安装 DNS 服务器。

在 192.168.226.130, 我们使用安装 BIND9 [3] DNS 服务器  
以下命令:

```
# sudo apt-get install bind9
```

步骤 2: 创建 named.conf.options 文件。

DNS 服务器需要读取/etc/bind/named.conf 配置文件启动。此配置文件通常包括一个选项文件称为/etc/bind/named.conf.options。

请将以下内容添加到选项文件:

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

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

步骤 3: 创建区域。

假设我们拥有一个域: example.com, 这意味着我们负责用于提供关于 example.com 的最终答案。因此,我们需要在中创建一个区域 DNS 服务器通过添加以下内容到/etc/bind/named.conf。 应该注意的是 example.com 域名保留供文档使用,不属于任何人,因此是安全使用它。

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

```
file "/var/cache/bind/192.168.0";
};
```

注意，我们使用 192.168.226.x 作为示例。 如果使用不同的 IP 地址，则需要更改 /etc/bind/named.conf 和 DNS 查找文件（如下所述）。

步骤 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。 在 / var / cache / bind / 目录中，撰写下面的 example.com.db 区域文件

```
$TTL 3D
@      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        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
                                   ;example.com. domain
```

我们还需要设置 DNS 反向查找文件。 在目录 / var / cache / bind / 中，编写 a 反向 DNS 查找文件名为 192.168.0 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.
```

步骤 5：启动 DNS 服务器。

现在我们准备好启动 DNS 服务器。 运行以下命令：

```
%sudo /etc/init.d/bind9 restart
```

```
%sudo sever bind9 restart
```

## 2.2 配置用户机器

在用户计算机 192.168.226.129 上,我们需要让机器 192.168.226.130 成为默认 DNS 服务器。我们通过更改用户计算机的 DNS 设置文件/etc/resolv.conf 实现这一点:

```
nameserver 192.168.226.130 # 刚刚设置的 DNS 服务器的 IP
```

做以下 (在 Ubuntu 12.04):

单击“系统设置” -> “网络”,

单击“有线”选项卡中的“选项”

选择“IPv4 设置” -> “方法” -> “自动 (DHCP) 地址”

并仅更新具有 BIND DNS 服务器的 IP 地址的“DNS 服务器”条目。

现在单击右上角的“网络图标”,然后选择

“Auto eth0”。这将刷新有线网络连接和

更新更改。

您应该重新启动您的 Ubuntu 计算机以使修改的设置生效。

## 2.3 配置攻击机

在攻击者机器上,没有太多配置。

## 2.4 预期产出

在你按照 Th 设置实验室环境后

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

You should be able to see something like this:

```
Terminal
; <<> DiG 9.8.1-P1 <<> www.example.com
;; global options: +cmd
;; Got answer:
;; ->HEADER<- opcode: QUERY, status: NOERROR, id: 47643
;; 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.                1000    IN      A      1.2.3.4

;; AUTHORITY SECTION:
ns.example.com.                 1000    IN      NS      ns.example.com.

;; ADDITIONAL SECTION:
ns.example.com.                 1000    IN      A      1.2.3.5

;; Query time: 2 msec
;; SERVER: 192.168.226.130#53(192.168.226.130)
;; WHEN: Fri Nov 18 05:54:51 2016
;; MSG SIZE rcvd: 88

[11/18/2016 05:54] seed@ubuntu:~$
```

## 2.5 安装 Wireshark

## 3 实验室任务

### 3.1 任务 1: 攻击者已经攻击了受害者的机器

修改 HOSTS 文件。使用 HOSTS 文件（/etc/hosts）中的主机名和 IP 地址对用于本地查找;它们优先于远程 DNS 查找。例如，如果有以下内容在用户计算机的 HOSTS 文件中输入，www.example.com 将解析为 1.2.3.4 用户的计算机而不要求任何 DNS 服务器：

1.2.3.4 www.example.com

### 3.2 任务 2: 对用户的直接欺骗响应

在这次攻击中，受害者的机器没有受到攻击，所以攻击者不能直接更改 DNS 查询进程在受害者的机器上。但是，如果攻击者处于同一个局域网上受害者，他们仍然可以实现巨大的伤害。当用户在 web 浏览器中键入网站的名称（主机名，例如 www.example.com）时，用户的计算机将向 DNS 服务器发出 DNS 请求以解析主机名的 IP 地址。在听到这个 DNS 请求后，攻击者可以欺骗假的 DNS 响应。假 DNS 答复将被用户的计算机接受，如果它符合以下标准：

- 1.源 IP 地址必须与 DNS 服务器的 IP 地址匹配。
- 2.目标 IP 地址必须与用户机器的 IP 地址匹配。
- 3.源端口号（UDP 端口）必须与 DNS 请求发送到的端口号匹配（通常为端口 53）。
- 4.目标端口号必须与发送 DNS 请求的端口号相匹配。

- 5.必须正确计算 UDP 校验和。
- 6.事务 ID 必须与 DNS 请求中的事务 ID 匹配。
- 7.答复问题部分中的域名必须与问题中的域名匹配部分。
- 8.答案部分中的域名必须与问题部分中的域名匹配

DNS 请求。

- 9.用户的计算机在接收合法 DNS 之前必须接收攻击者的 DNS 回复响应。

为了满足标准 1 到 8，攻击者可以窥探受害者发送的 DNS 请求消息;他们可以然后创建一个假的 DNS 响应，并发送回受害者，在真正的 DNS 服务器之前。

### Netwox

工具 105 提供进行这种嗅探和响应的实用程序。

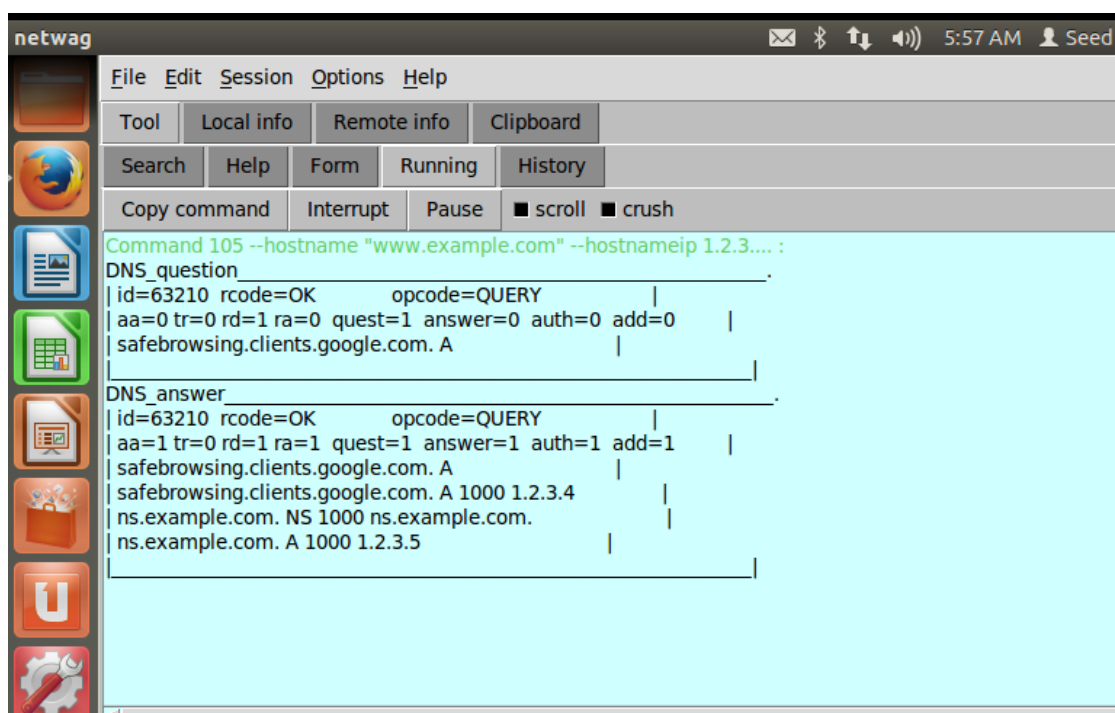
提示：在 Netwox / Netwag 工具 105 中，您可以使用“过滤器”字段指示您的 IP 地址

目标。例如，在下面显示的场景中，您可以使用“src host 192.168.226.129”。



### 3.3 任务 3: DNS 服务器缓存中毒

上述攻击针对的是用户的机器。为了达到持久的效果，每次用户的机器发出一个 DNS 查询 `www.example.com`，攻击者的机器必须发出一个欺骗 DNS 响应。这可能不是那么高效;有一个更好的方式来进行攻击的目的 DNS 服务器，而不是用户的机器。当 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。请注意，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，在阿波罗询问其他 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 服务器阿波罗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。



The screenshot shows the netwag application window. The title bar includes the name 'netwag' and system icons for network, volume, and time (5:57 AM). The menu bar contains 'File', 'Edit', 'Session', 'Options', and 'Help'. Below the menu bar are several tabs: 'Tool', 'Local info', 'Remote info', 'Clipboard', 'Search', 'Help', 'Form', 'Running', 'History', 'Copy command', 'Interrupt', 'Pause', 'scroll', and 'crush'. The main display area shows the output of a command: 'Command 105 --hostname "www.example.com" --hostnameip 1.2.3.... :'. The output is divided into two sections: 'DNS\_question' and 'DNS\_answer'. The 'DNS\_question' section shows a query for 'safebrowsing.clients.google.com. A' with various flags. The 'DNS\_answer' section shows a response with multiple records, including 'safebrowsing.clients.google.com. A 1000 1.2.3.4' and 'ns.example.com. NS 1000 ns.example.com.'.

```
netwag
File Edit Session Options Help
Tool Local info Remote info Clipboard
Search Help Form Running History
Copy command Interrupt Pause scroll crush
Command 105 --hostname "www.example.com" --hostnameip 1.2.3.... :
DNS_question
|id=63210 rcode=OK      opcode=QUERY      |
|aa=0 tr=0 rd=1 ra=0  quest=1  answer=0  auth=0  add=0  |
|safebrowsing.clients.google.com. A      |
DNS_answer
|id=63210 rcode=OK      opcode=QUERY      |
|aa=1 tr=0 rd=1 ra=1  quest=1  answer=1  auth=1  add=1  |
|safebrowsing.clients.google.com. A      |
|safebrowsing.clients.google.com. A 1000 1.2.3.4      |
|ns.example.com. NS 1000 ns.example.com.      |
|ns.example.com. A 1000 1.2.3.5      |
```