

实操实验教程

项目 2 : 第 1 部分

数据包过滤防火墙 (*iptables*)

目录

1. 任务 1 实验室环境的准备工作	5
1.1 任务 1.1 测试网络连接性	6
1.2 任务 1.2 测试已安装的软件和服务	7
1.3 任务 1.3 将防火墙重置为白名单	8
2. 任务 2 设置无状态数据包过滤防火墙的要求	9
3. 交付成果	9
4. 第 1 部分 实验室评估 (100 分)	10
5. 相关信息和资源	11

类别：

CS-CNS:计算机网络安全

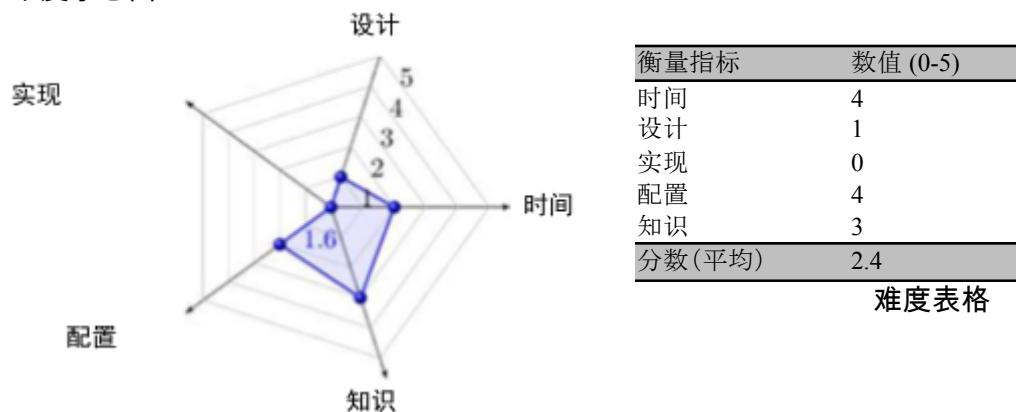
目标：

- 1 设置数据包过滤防火墙 (*iptables*)，以允许和阻止网络流量
- 2 设置网络地址转换 (NAT) 服务
- 3 使用基本的网络和诊断工具，如 ifconfig、ip、route、netstat、ping、traceroute 以及 tcpdump
- 4 根据已提供的流量策略，用 *iptables* 来调节网络流量
并启用 Web、FTP、SSH 等服务

预计时长：

1. 行家：120 分钟
2. 新手：360 分钟

难度示意图：

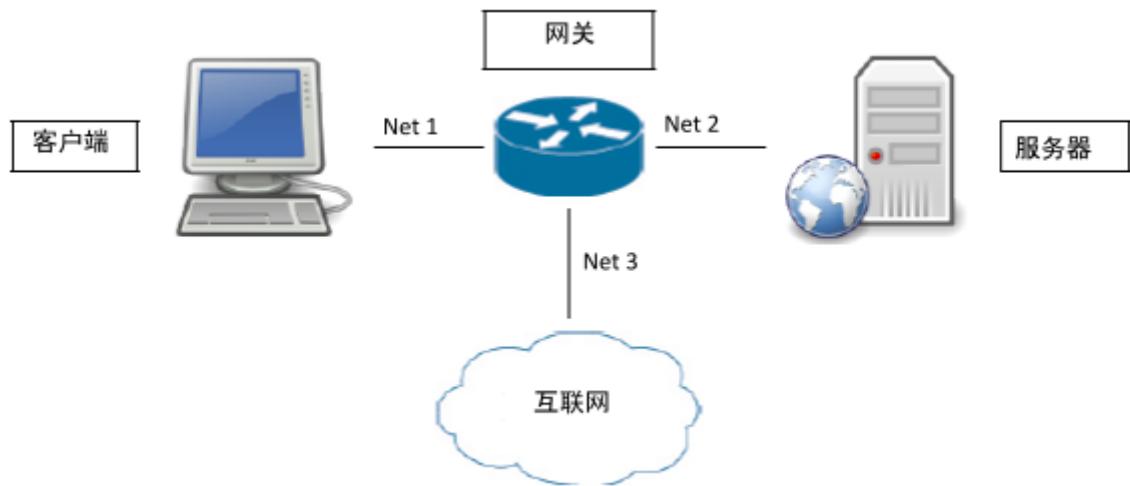


所需的操作系统：

Linux Ubuntu

实验室运行环境：

学生个人计算机上的 VMware 或其他虚拟机解决方案



1. 客户端 : Linux (Ubuntu 18.04 LTS)
2. 服务器 : Linux (Ubuntu 18.04 LTS)
3. 网关 : Linux (Ubuntu 18.04 LTS)
4. 网络设置：
通过 Net 3 连接互联网 : 172.16.0.0/12
客户端 Net 1 : 192.168.0.0/24
服务器端 Net 2 : 10.0.0.0/8

实验室的准备工作：

1. 了解如何使用 Linux 操作系统
2. 有关计算机网络的基础知识
3. 了解如何设置 Web、FTP 和 SSH 等服务。

实验室概述

在本实验室中，你将利用 Linux 防火墙 *iptables* 来探索数据包过滤防火墙。实验室的第一部分是建立必要的 *iptables* 运行环境；第二部分会详细说明实现防火墙过滤规则以启用和禁用网络流量的要求。

实验室的评估是根据实现满足所需防火墙安全策略的防火墙过滤规则的完整性做出的。学生需要提交一系列屏幕截图和相应的图例，证明自己是如何满足防火墙的数据包过滤要求的。

总体而言，学生需要完成的任务包括：

- 使用 ifconfig、route、ip、ping、traceroute 和 tcpdump 等工具，设置网络数据包转发并检查流量
- 检查服务设置，如 apache2 网络服务、vsftpd 服务和 ssh 服务
- 使用并测试基于 *iptables* 的数据包过滤防火墙，以启用和禁用对已建立服务的访问

1. 任务 1 实验室环境的准备工作

在本实验室中，会提供一个 *iptables* 防火墙运行脚本模板，该模板让你可以更轻松地管理和运行 *iptables* 规则。你可以从实验室资源库中下载并解压缩该模板。你可以用下面的 wget 和 unzip 命令来下载实验室资源文件。注意，请用以下命令检查 wget 是否已安装：

```
$ wget --version  
$ unzip --version
```

如果未安装 wget 和 unzip，你可以用以下命令安装：

```
$ sudo apt install wget  
$ sudo apt install unzip
```

然后，下载压缩的实验室文件：

```
$ Wget  
https://gitlab.thothlab.org/thoth-group/ThoThLabResource/raw/master/lab-cs-cn  
s-00001a.zip  
$ unzip lab-cs-cns-00001a.zip  
$ cd lab-cs-cns-00001a
```

防火墙脚本模板文件“rc.firewall”位于文件夹“lab-cs-cns-00001a”中。这是一个 shell 脚本。要使其可执行，你可以通过以下方式更改其权限（参阅实验 CS-SYS-00001 中有关 Linux 文件权限的更多详细信息）：

```
$ sudo chmod 755 rc.firewall % 当你显示“ls -l”命令时，这将把文件变成绿色
```

rc.firewall 是一个 shell 脚本，可以帮助你更轻松地管理防火墙规则。该脚本只包含一些基本的设置。为了实现本实验室的总体目标，你需要在其中添加和更新防火墙规则。要编辑并运行脚本文件，你可以：

```
$ vim rc.firewall % 用 vim 来编辑该脚本。脚本中的注释  
足以说明清楚。  
$ sudo ./rc.firewall % 运行该脚本。
```

1.1 任务 1.1 测试网络连接性

第一步是检查虚拟机之间的连接性。

1. 使用 *ping* 检查连接（如果此步骤成功，请跳到任务 1.2）：

```
$ ping ip_address % 你需要在连接至同一本地网络上的任意一对虚拟机之间  
互相执行 ping
```

通常，*ping* 响应不成功会导致以下情况：

Ping 响应的可能原因	可能原因
请求超时	超时，如 windows 的默认超时时间是 4 秒
<终点>无应答	终点无回应，路径上的路由器工作正常。
<终点>不可达	源节点不知道如何到达终点，比如路由有问题
无法从网关访问 ICMP 主机	转发数据包的网关/路由器设置不正确

2. 检查客户端和服务器上是否正确设置了默认网关。默认网关应设置为直接连接到客户端或服务器网络的网关 IP 地址。例如：

```
$ route -n % 检查默认网关设置  
$ sudo route add default gw <default_gw_ip> <interface_to_gw_net> % 将默认网  
关设置为直接连接到接口的默认网关 IP
```

有关如何检查默认网关设置的详细信息，可参阅课程项目 1。检查/设置默认网关配置后，再次相互执行 *ping*。

3. 如果你仍然无法对客户端或服务器的网关执行 *ping*，则可能需要检查网关上的防火墙设置是否阻止了 *ping*。你可能需要禁用网关上的防火墙，然后再次尝试 *ping*：

```
$ sudo ufw disable % 禁用防火墙
```

4. 检查连通性后，你应该检查网关上的数据包转发设置：

(a) 启用网关上的数据包转发

```
$ sudo echo "1" > /proc/sys/net/ipv4/ip_forward
```

(b) 要检查当前的 *iptables* 规则设置，你可以发出以下命令：

```
$ sudo iptables -L % 显示过滤表策略。对于白名单，输入、输出和转发链的默认  
策略应为 DROP。
```

(c) 在网关上，清除所有现有的 *iptables* 规则（如果你已建立 *iptables* 规则，请谨慎使用）：

```
$ sudo iptables -F % 清除所有现有的链  
$ sudo iptables -X % 删除所有用户定义的链
```

- (d) 将 *iptables* 默认策略设置为 *blacklist*¹ (黑名单)。在完成以下 *iptables* 设置后，你应该能够在任意两个虚拟机之间执行 ping。

```
$ sudo iptables -P INPUT ACCEPT % option - P 表示默认策略  
$ sudo iptables -P OUTPUT ACCEPT  
$ sudo iptables -P FORWARD ACCEPT
```

完成上述步骤后，防火墙规则将被清除，虚拟机之间的数据包发送不再受限制。因此，你应该可以在任意两个虚拟机之间执行 ping。

1.2 任务 1.2 测试已安装的软件和服务

第二步是确保在给定的虚拟机上正确安装项目所需的软件包。

注意，你可能需要调整 bind9、apache2、ssh 和 vsftp 的配置，使其适应下文中的要求。

1. 在服务器上测试 web 服务器，确保其正常工作：

- (a) 测试运行以下命令的 Apache 服务器：

```
$ service apache2 status
```

- (b) 通过编辑文件 */var/www/html/index.html* 建立一个演示网站，并添加一条语句，如“欢迎观看演示视频并完成测试！”。

2. 在服务器上，测试 ftp 服务器，确保 vsFTP 服务器以被动模式运行，即数据通道将由客户端启用。因此，你需要在网关上建立端口转发，以便将 ftp 数据通道请求转发到服务器端的指定数据接收端口。有关如何将 vsFTP 设置为被动模式的更多详细信息，可以参考 <https://www.myfreak.com/how-to-setup-ftp-server-with-vsftpd-on-ubuntu-18-04/>。你需要在 ftp 服务器上执行以下配置：

- 启用匿名访问（即无需提供用户帐户和密码）。
- 启用被动模式，即客户端可以通过端口范围[30000，30099]启动数据通道。

你可以在本地测试 ftp 服务。对于本项目，你可能不需要设置 ftp 身份验证和安全性。但你需要确保被动模式已启用。

3. 最后，在服务器上测试 SSH 服务器是否正在运行。

```
$ sshd -v % 显示 ssh 服务器版本  
$ ssh ubuntu@localhost % 设置与服务器本身的 ssh 连接。SSH 服务器不允许 root 用户远程访问服务器，因此你应该使用用户帐户“ubuntu”来访问 SSH 服务。
```

¹ 防火墙黑名单策略意味着只阻止已知的非法流量，并允许所有未指定的网络流量通过。

1.3 任务 1.3 将防火墙重置为白名单

在确保网络连接良好且客户端访问任务 1.2 中描述的所有服务后，你需要强制执行白名单 (*whitelist*) 防火墙策略，作为下一个任务的实验室设置起点，并清除所有现有的防火墙规则和链。防火墙白名单策略意味着防火墙只允许已知的合法流量通过，并阻止所有未指定/未知的网络流量。设置白名单策略后，你应该不能在任何给定的虚拟机之间成功执行 ping，也应该不能从客户端访问服务器上建立的任何服务。

首先，清除 *iptables* 链并删除所有用户定义的链：

```
$ sudo iptables -F          % 清除 ipables 规则  
$ sudo iptables -F          % 删除用户定义链
```

其次，将默认 *iptables* 策略设置为白名单：

```
$ sudo iptables -P INPUT DROP % option - P 表示默认策略  
$ sudo iptables -P OUTPUT DROP  
$ sudo iptables -P FORWARD DROP
```

现在，你应该不能在虚拟机之间执行 ping，也不能从客户端访问服务器托管的服务。

2. 任务 2 设置无状态数据包过滤防火墙的要求

在网关上，请设置以下数据包过滤规则。对于每个必需的规则，演示如何满足每个规则。

1. 检查并将输入、输出和转发链的默认 *iptables* 策略设置为 DROP。这种设置基本上是实现白名单策略，即只允许特定网络流量作为“好”的流量通过，并且禁用所有其他非指定流量。
注意，只有以下所述的所需流量和连接被允许，同时阻止所有其他网络流量和访问。
2. 考虑到服务器端网络是一个私有和受保护的网络，并在网关上正确配置 NAT 服务，以更改访问服务器端私有网络的 IP 地址。(提示：你可以使用 *tcpdump* 来捕获客户端网络上的流量，以验证服务器发送的数据包是否已更改为网关的 IP 地址)。
3. 允许客户端用网关的 IP 访问服务器上的网页 (http)。演示网页应包含关键字“欢迎”，例如“欢迎观看演示视频并测试网页！”
 - http://192.168.0.100
4. 允许客户端用被动模式访问服务器托管的 ftp 服务器，并允许匿名访问，即用以下命令访问 FTP 服务器：

```
$ ftp -p 192.168.0.100 % 用“anonymous”作为用户 ID 和密码来访问服务器
```

设置从服务器打开的被动数据端口的范围为 [30000, 30099]。

5. 允许客户端对网关的客户端一侧的 IP 地址执行 ping。
6. 允许服务器对网关的服务器端一侧的 IP 地址和客户端 IP 地址执行 ping。

3. 交付成果

学生需要提交一系列屏幕截图并附解释说明，证明自己可以达到实验室评估部分所述的要求

4. 第 1 部分 实验室评估 (50 分)

完成任务 1 和任务 2 的实验室评估取决于以下完成情况：

1. (20 分) 客户端

- 可对客户端网络上的网关 IP 地址执行 ping；
- 可对 `ubuntu@gateway` 执行 ssh 并获得对服务器的访问权限；
- 可对网关 IP 执行 `ftp -p` 并访问服务器虚拟机上运行的 FTP 服务器(使用匿名和被动模式，注意，客户端无法在主动模式下发出“ls”等命令，否则系统会死机)
- 可使用网关 IP 访问服务器虚拟机上运行的 Apache 服务器(返回页面必须包含“欢迎……”，你也可以使用网络浏览器)

2. (15 分) 网关

- 应设置 www、ssh 和 ftp 到服务器 IP 地址 (NATed IP 地址) 的端口转发。
- 应启用 POSTROUTING，以允许服务器访问外部网络并更改其源 IP 地址。

3. (10 分) 服务器

- 可对网关和客户端 IP 地址执行 ping。
- 可响应客户端节点对其 www、ssh 和 ftp 服务的请求。

4. (5 分) 附加要求

- 对于输入、输出和转发链，你应该将默认防火墙策略设置为 DROP。
- 除上述允许的网络访问外，任何其他网络访问都不应准许。提供以下结果的屏幕截图：

在客户端虚拟机上：

```
$ sudo nmap -sT -p- 192.168.0.x %
x 是客户端网络上网关 IP 地址的值
$ sudo nmap -sU -p- 192.168.0.x %
x 是客户端网络上网关 IP 地址的值
$ ping 8.8.8.8
$ ping 192.168.0.x %
x 是客户端网络上网关 IP 地址的值
$ ping 10.0.0.w %
w 是服务器 IP 地址的值
```

在服务器虚拟机上：

```
$ sudo nmap -sT -p- 10.0.0.y %
y 是服务器端网络上网关 IP 地址的值
$ sudo nmap -sU -p- 10.0.0.y %
y 是服务器端网络上网关 IP 地址的值
$ ping 8.8.8.8
$ ping 10.0.0.y %
y 是服务器端网络上网关 IP 地址的值
$ ping 192.168.0.z %
z 是客户端 IP 地址的值
```

在网关虚拟机上：

```
$ ping 8.8.8.8  
$ ping 192.168.0.z % z 是客户端 IP 地址的值  
$ ping 10.0.0.w % w 是服务器 IP 地址的值
```

5. 相关信息和资源

IptablesHowTo <https://help.ubuntu.com/community/IptablesHowTo>
《Iptables 教程》1.2.2 版本
<https://www.frozenthux.net/iptables-tutorial/iptables-tutorial.html>