

Linux Firewall Exploration

实验报告

|  |  |
| --- | --- |
| 学生姓名 | maybeLocalhost |
| 学 号 |  |
| 专业班级 |  |
| 指导教师 |  |
| 学 院 | 计算机学院 |
| 完成时间 | 2020.10 |

目录

[一、实验简介 1](#_Toc58492135)

[1.1 实验背景 1](#_Toc58492136)

[1.2 实验目的 1](#_Toc58492137)

[1.3 实验内容 1](#_Toc58492138)

[1.4 实验环境 1](#_Toc58492139)

[二、实验原理 2](#_Toc58492140)

[2.1 SSH 2](#_Toc58492141)

[2.2 Netfilter 2](#_Toc58492142)

[2.3 反向SSH隧道 2](#_Toc58492143)

[三、实验过程 3](#_Toc58492144)

[3.1 Task1：使用防火墙 3](#_Toc58492145)

[3.1.1 Task1.1：阻止A对B的telnet连接 3](#_Toc58492146)

[3.1.2 Task1.2：阻止B对A的telnet连接 4](#_Toc58492147)

[3.1.3 Task1.3：阻止A访问外部网站 4](#_Toc58492148)

[3.2 Task2：实现简单的防火墙 5](#_Toc58492149)

[3.3 Task3：绕过出口过滤 7](#_Toc58492150)

[3.3.1 Task3.a：通过防火墙Telnet连接到计算机B 8](#_Toc58492151)

[3.3.2 Task3.b：使用SSH隧道连接到Bilibili 8](#_Toc58492152)

[3.4 Task4：绕过入口过滤 11](#_Toc58492153)

[四、实验结果与结论 12](#_Toc58492154)

[五、参考文献 13](#_Toc58492155)

# 一、实验简介

## 1.1 实验背景

本实验的学习目标是让学生通过使用防火墙软件并实现简化的数据包过滤防火墙来了解防火墙如何工作。防火墙有几种类型。在本实验中，我们专注于包过滤器。数据包过滤器检查数据包，并根据防火墙规则决定是丢弃还是转发数据包。数据包过滤器通常是无状态的; 它们仅基于该数据包中包含的信息过滤每个数据包，而无需注意数据包是否是现有流量的一部分。数据包过滤器通常结合使用数据包的源地址和目标地址，其协议以及端口号（对于TCP和UDP通信）。

## 1.2 实验目的

在本实验中，学生将使用这种类型的防火墙，并且通过实施某些关键功能，他们可以了解防火墙的工作原理。此外，学生还将学习如何使用SSH隧道绕过防火墙。

## 1.3 实验内容

本实验涵盖以下主题：

1. 防火墙功能
2. 网络过滤器
3. 可加载的内核模块
4. 使用SSH隧道绕过防火墙
5. 一个单独的实验室介绍了如何使用VPN绕过防火墙。

## 1.4 实验环境

Ubuntu 16.04

seed1（A）：192.168.120.138

seed2（B）：192.168.120.139

seed3（C）：192.168.120.137

# 二、实验原理

## 2.1 SSH

SSH 为建立在应用层基础上的安全协议。SSH 是较可靠，专为远程登录会话和其他网络服务提供安全性的协议。利用 SSH 协议可以有效防止远程管理过程中的信息泄露问题。SSH最初是UNIX系统上的一个程序，后来又迅速扩展到其他操作平台。SSH在正确使用时可弥补网络中的漏洞。SSH客户端适用于多种平台。几乎所有UNIX平台—包括HP-UX、Linux、AIX、Solaris、Digital UNIX、Irix，以及其他平台，都可运行SSH。netfilter主要采用连线跟踪（Connection Tracking）、包过滤（Packet Filtering）、地址转换、包处理（Packet Mangling)4种关键技术。

## 2.2 Netfilter

Netfilter是Linux 2.4.x引入的一个子系统，它作为一个通用的、抽象的框架，提供一整套的hook函数的管理机制，使得诸如数据包过滤、网络地址转换(NAT)和基于协议类型的连接跟踪成为了可能。

netfilter的架构就是在整个网络流程的若干位置放置了一些检测点（HOOK），而在每个检测点上登记了一些处理函数进行处理。

## 2.3 反向SSH隧道

反向SSH隧道允许我们使用已建立的连接来建立从本地计算机到远程计算机的新连接。因为SSH连接是从远程计算机到本地计算机的，所以将其用于另一个方向就是“反向”使用它。反向SSH隧道依赖于使用已建立连接的远程计算机来侦听来自本地计算机的新连接请求。远程计算机在本地计算机上的网络端口上侦听。 如果它检测到对该端口的SSH请求，则会通过已建立的连接将该连接请求中继回自身。 这提供了从本地计算机到远程计算机的新连接。内网的A通过SSH连接公网的B，并且两者建立一个通信隧道，通常防火墙只限制入站不限制出站，所以SSH可以穿透防火墙

# 三、实验过程

## 3.1 Task1：使用防火墙

### 3.1.1 Task1.1：阻止A对B的telnet连接

开始时，A可以对B进行telnet：

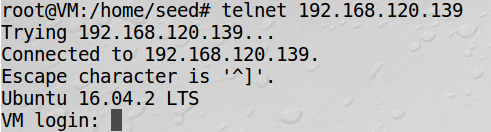


图 1 阻止前A可以对B进行telnet连接

然后我们在B终端运行以下命令：

iptables -A INPUT -s 192.168.120.138 -j DROP



图 2 添加iptables规则

再次进行A对B的telnet连接，发现无法连接：

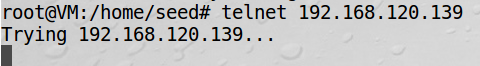


图 3 A无法对B进行telnet连接

### 3.1.2 Task1.2：阻止B对A的telnet连接

开始时，B可以对A进行telnet：

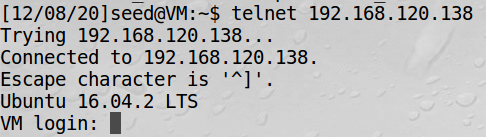


图 4 阻止前B可以对A进行telnet连接

然后我们在A终端运行以下命令：

iptables -A INPUT -s 192.168.120.139 -j DROP



图 5 添加iptables规则

再次进行B对A的telnet连接，发现无法连接：

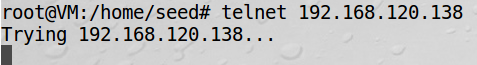


图 6 B无法对A进行telnet连接

### 3.1.3 Task1.3：阻止A访问外部网站

未阻止时，A可以ping通外部网站:

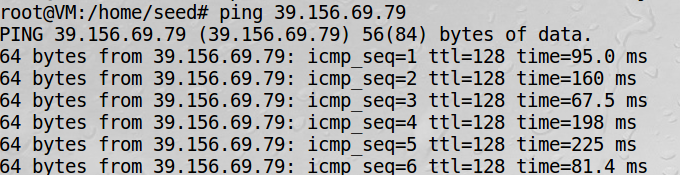


图 7 阻止前A可以ping通外部网站

然后，我们运行以下命令：

iptables -A OUTPUT -d 39.156.69.79 -j DROP



图 8 添加iptables规则

再次尝试去ping同一网站，发现无法ping通，提示操作不允许：

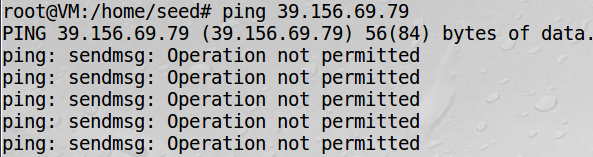


图 9 无法ping通外部网站

## 3.2 Task2：实现简单的防火墙

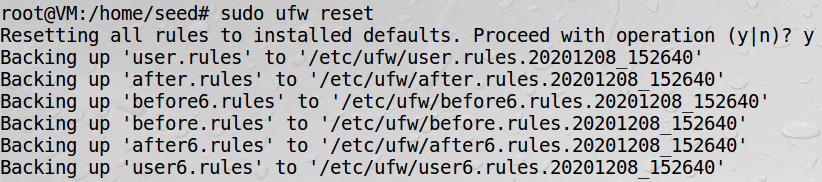
首先，我们对刚刚实验里A主句和B主机中增加到规则进行删除：  


图 10 重置主机A中的防火墙规则

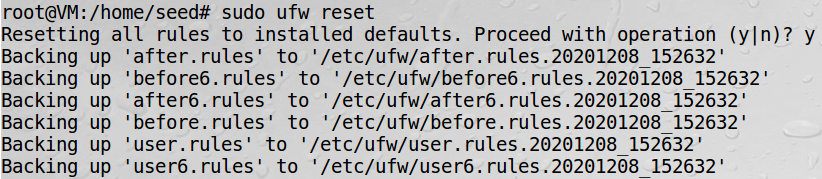


图 11 重置主机B中的防火墙规则

然后根据实验指导书中的内容使LKN和Netfilter来实现包过滤模块，其关键部分代码如下：

1. **int** eq\_daddr(**const** **struct** iphdr \*iph, **const** **char** \*ip\_addr)
2. {
3. //check if the dst ip address equals the given address
4. **char** source[16];
5. snprintf(source, 16, "%pI4", &iph->daddr);
6. **if** (strcmp(source, ip\_addr) == 0)
7. **return** 1;
8. **return** 0;
9. }
11. **int** eq\_saddr(**const** **struct** iphdr \*iph, **const** **char** \*ip\_addr)
12. {
13. //check if the src ip address equals the given address
14. **char** source[16];
15. snprintf(source, 16, "%pI4", &iph->saddr);
16. **if** (strcmp(source, ip\_addr) == 0)
17. **return** 1;
18. **return** 0;
19. }
21. unsigned **int** telnetFilter\_1(**void** \*priv, **struct** sk\_buff \*skb,
22. **const** **struct** nf\_hook\_state \*state)
23. // rule for task 1.2: Prevent B from doing `telnet` to Machine A
24. {
25. **struct** iphdr \*iph;
26. **struct** tcphdr \*tcph;
28. iph = ip\_hdr(skb);
29. tcph = (**void** \*)iph + iph->ihl \* 4;
31. **if** (iph->protocol == IPPROTO\_TCP && tcph->dest == htons(23) && eq\_daddr(iph, "192.168.120.138") && eq\_saddr(iph, "192.168.120.139"))
32. {
33. printk(KERN\_INFO "Dropping telnet from %pI4 packet to %pI4\n", &iph->saddr, &iph->daddr);
34. **return** NF\_DROP;
35. }
36. **else**
37. {
38. **return** NF\_ACCEPT;
39. }
40. }

对该程序进行编译，获取相应模块并安装：

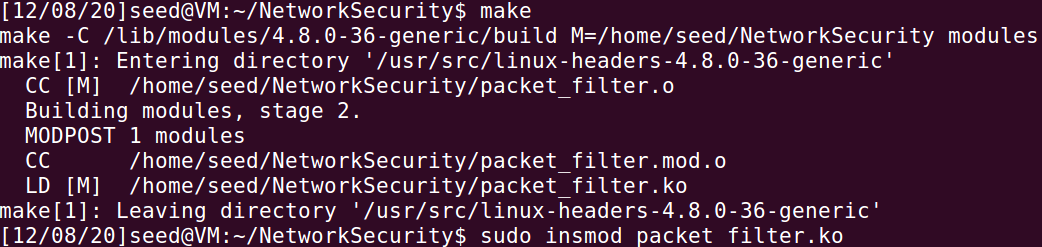


图 12 编译并安装模块

在主机B中尝试使用telnet连接到A，发现无法连接，被防火墙阻止：

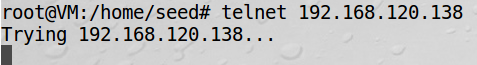


图 13 主机B无法telent连接到A

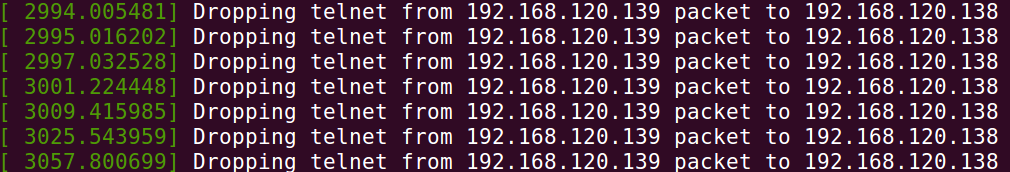


图 14 防火墙日志

## 3.3 Task3：绕过出口过滤

许多公司和学校都实施出口过滤，来阻止了网络内部的用户访问某些网站或Internet服务。在许多情况下，这种类型的防火墙会检查传出数据包中的目标IP地址和端口号。如果数据包符合限制，它将被丢弃。由于性能原因，它们通常不进行深入的数据包检查（即，调查数据包的数据部分）。在此任务中，我们将使用SSH隧道机制绕过此类出口过滤。

在该实验中，我们将使用3个虚拟机。主机A在防火墙后面运行（即，在公司或学校的网络内部），而主机B和主机C在防火墙外部。

首先，我们需要在主机A上使用iptables设置以下两个防火墙规则：

* 阻止所有传出到外部telnet服务器的流量：

sudo ufw deny out from 192.168.120.138 to any port 23

* 阻止所有流向[www.bilibili.com](http://www.bilibili.com)的出站流量

sudo ufw deny out from 192.168.120.138 to 139.159.241.37

### 3.3.1 Task3.a：通过防火墙Telnet连接到计算机B

在该任务中我们使用以下指令，令主机A通过主机C将主机B的23端口转发到自己的2333端口上：

ssh -fNL 2333:192.168.120.139:23 [seed@192.168.120.137](mailto:seed@192.168.120.137)

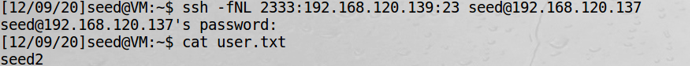


图 15 通过SSH进行端口转发

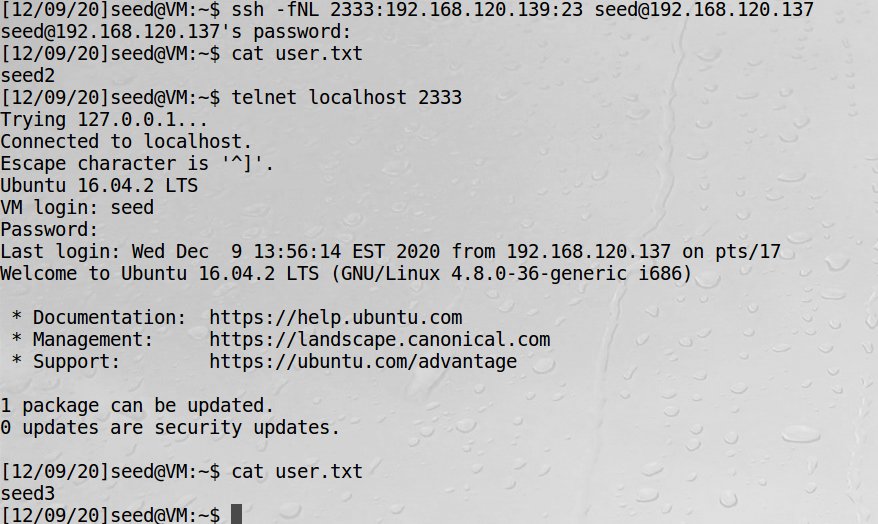


图 16 telnet连接成功

可以看到主机A通过telnet成功地连接到主机B。

### 3.3.2 Task3.b：使用SSH隧道连接到Bilibili

首先我们在浏览器中访问bilibili.com可以发现无法访问：

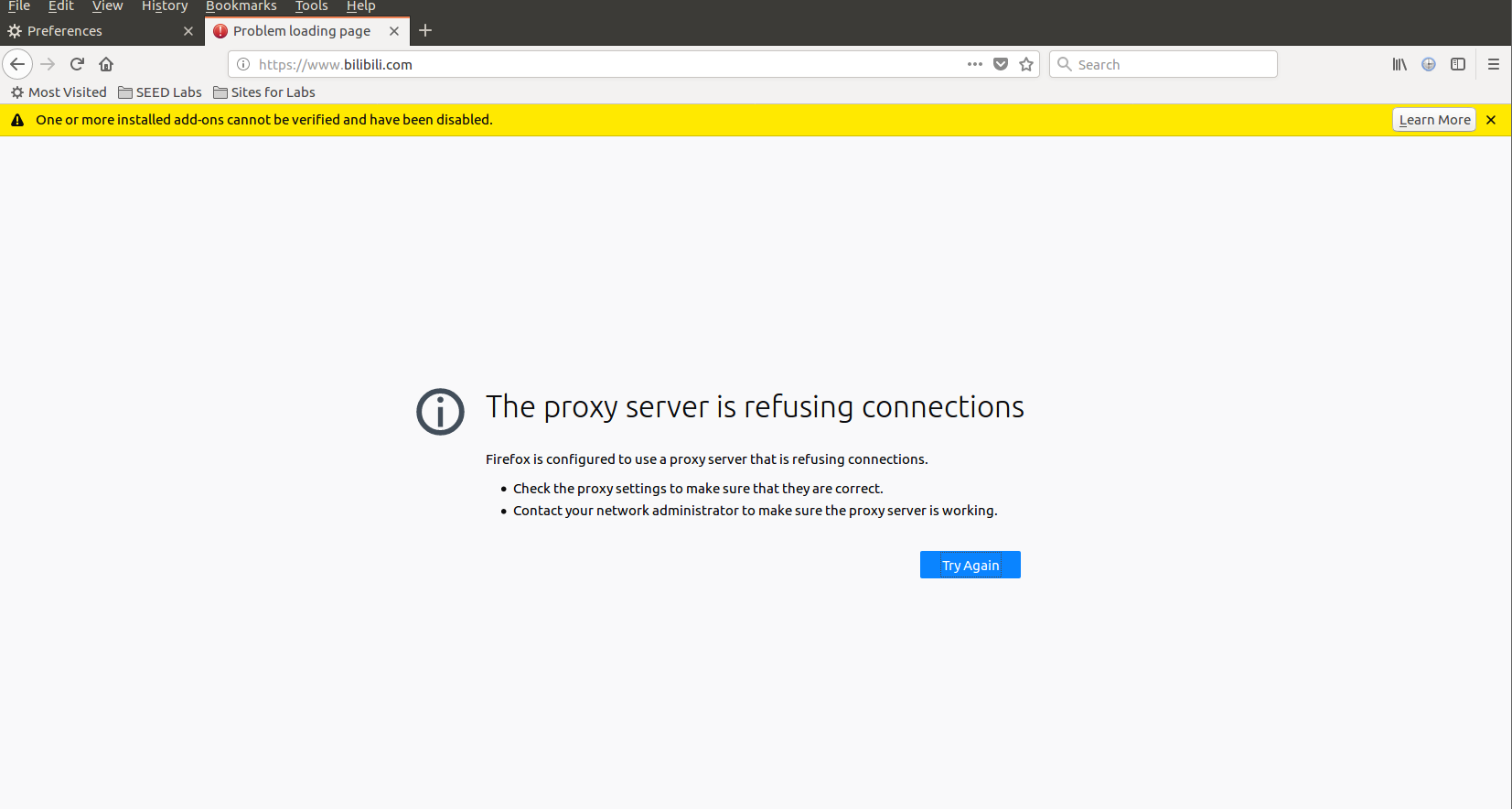


图 17 无法访问bilibiili

然后我们在主机A运行以下命令，与主机C建立SSH隧道：

ssh -D 8848 -C seed@192.168.120.137

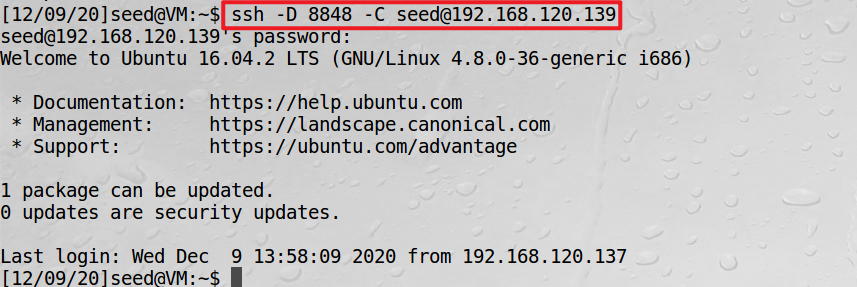


图 18 A与C建立SSH隧道

配置主机A浏览器的代理端口：

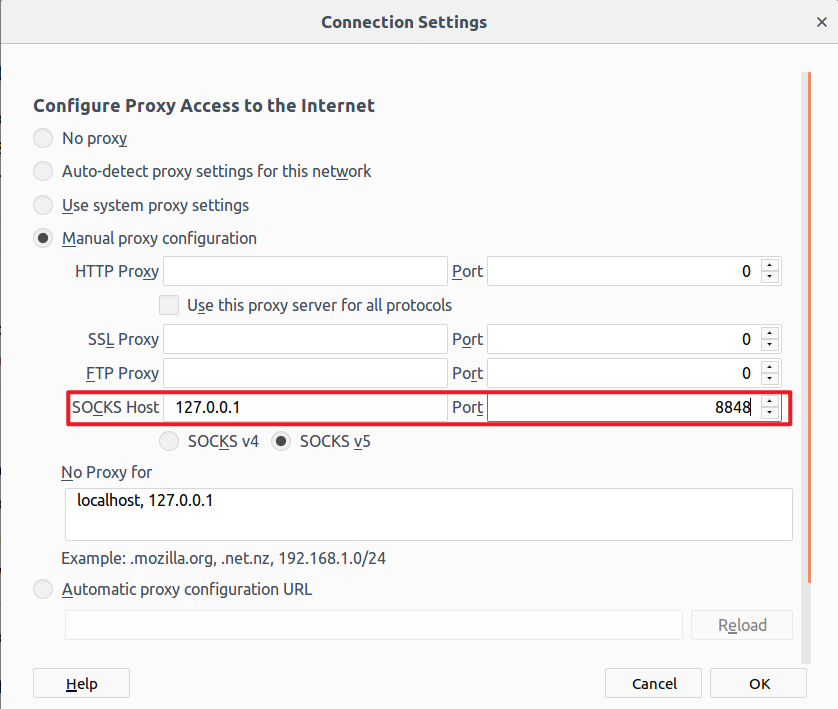


图 19 设置浏览器代理端口

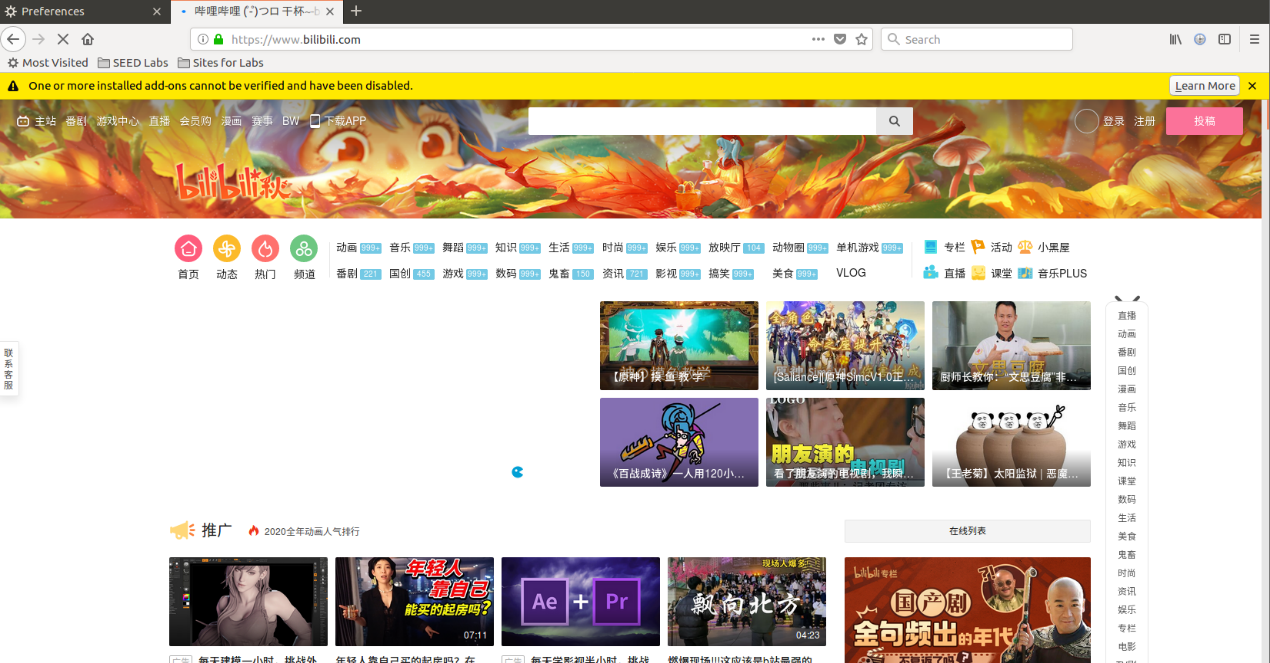
再次对bilibili.com进行访问，可以发现成功访问：  


图 20 成功访问bilibili

## 3.4 Task4：绕过入口过滤

机器A在防火墙后面运行Web服务器；因此，只有内部网络中的计算机才能访问此Web服务器。假设我们在家工作，需要访问此内部Web服务器。我们没有VPN，但还拥有SSH。我们确实在计算机A（或防火墙后面的另一台内部计算机）上有一个帐户，但是防火墙也阻止了传入的SSH连接，因此我们无法SSH到内部网络上的任何计算机。但是，防火墙不会阻止传出SSH连接，即，如果我们想连接到外部SSH服务器，则仍然可以这样做。

此任务的目的是能够从外部访问计算机A上的Web服务器。我们将使用两台机器来模拟设置。机器A是内部计算机，运行受保护的Web服务器；机器B是在家中的外部机器。在计算机A上，我们阻止计算机B访问其端口80（Web服务器）和22（SSH服务器）。您需要在计算机A上建立反向SSH隧道，因此一旦回到家中，仍然可以在家中访问A上受保护的Web服务器。

首先，我们在主机A上添加拒绝主机B访问其80端口和22端口的规则：

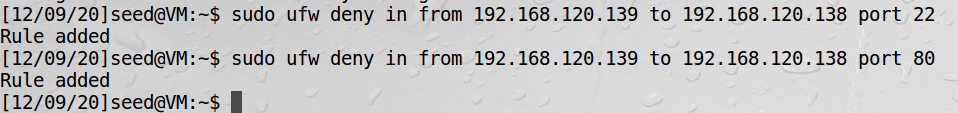


图 21 添加防火墙规则

然后使用以下命令在主机A上使SSH创建反向SSH隧道：

ssh -R 8866:localhost:80 seed@192.168.120.139

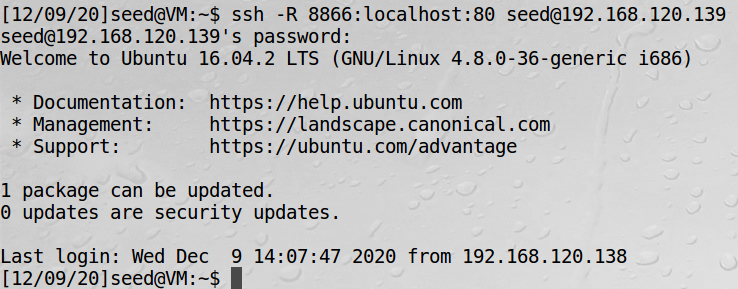


图 22 创建反向SSH隧道

此时可以在主机B使用浏览器通过本地的8866端口访问主机A的网站,反向SSH隧道建立成功：

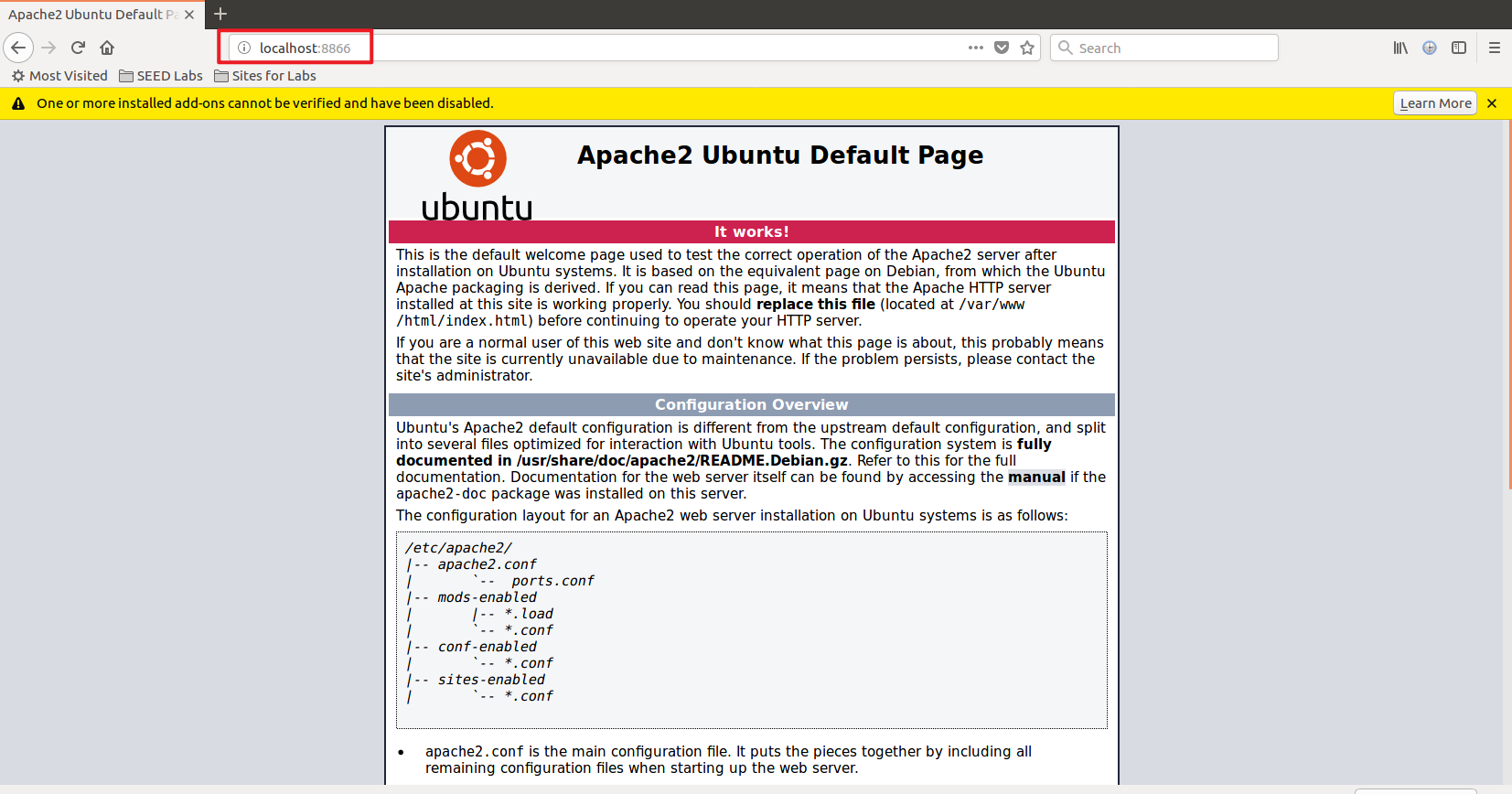


图 23 主机B成功访问主机A的网站

# 四、实验结果与结论

通过本次实验，我学习了Linux防火墙的基本使用方法，加深了对TELNET和SSH的理解，并能够根据给定规则编写简单的Linux防火墙。除此外，我还学习了通过SSH绕过防火墙的方法。

最后，根据实验过程，我们可以根据该攻击的过程来制定相应的预防措施。目前较为常见的防御措施如下：

1. 使用ip白名单；

仅允许白名单上的ip能够通过SSH或TELNET连接到主机，来避免防火墙被攻击绕过。

1. 使用sshd\_config限制用户能够转发的端口；
2. 通过 iptables限制某些主机端口对。

# 五、参考文献

1. 杜文亮.计算机安全导论：深度实践[M].高等教育出版社:北京,2020:1-
2. Hev. 防止 SSH 端口转发穿透防火墙. https://hev.cc/1186.html
3. StackExchange. How does reverse SSH tunneling work?. https://unix.stackexchange.com/questions/46235/how-does-reverse-ssh-tunneling-work
4. 常名先生. 使用SSH反向隧道进行内网穿透——持续更新中. https://blog.csdn.net/jiangbenchu/article/details/84438959
5. 骏马金龙. SSH隧道：端口转发功能详解. https://www.cnblogs.com/f-ck-need-u/p/10482832.html