

Linux Firewall Explorationn Lab

57117231 农禄 2020/09/19

实验环境

开启防火墙的操作系统：ubuntu 16.04(target 192.168.248.132)

其他主机：ubuntu 16.04(dxq 192.168.248.128)、ubuntu 16.04(third 192.168.248.133)

虚拟机载体：vmware

Task1: Using Firewall

用虚拟机 target 开启防火墙，ip：192.168.248.132

用虚拟机 dxq 访问 target，ip：192.168.248.128

使用 iptables 创建防火墙

通过 man ufw(或 ufw -h)命令查看 ufw 的帮助文档

修改配置文件/etc/default/ufw 文件， 默认对入方向的包进行 ACCEPT 操作

```
# Set the default input policy to ACCEPT, DROP, or REJECT. Please note that if
# you change this you will most likely want to adjust your rules.
DEFAULT_INPUT_POLICY="ACCEPT"
```

1. 阻止 target 对 dxq 的 telnet 连接

```
root@VM:/home/seed# ufw deny out to 192.168.248.128 port 23
Rule added
root@VM:/home/seed#
```

```
root@VM:/home/seed# ufw deny out to 192.168.248.128 port 23
Rule added
root@VM:/home/seed# telnet 192.168.248.128
Trying 192.168.248.128...
```

此时无法向 dxq 发起连接

2. 阻止 dxq 对 target 的 telnet 连接

```
root@VM:/home/seed# ufw deny proto tcp from 192.168.248.128 to 192.168.248.132 port 23
Rule added
root@VM:/home/seed#
```

```
root@VM:/home/seed/Course# telnet 192.168.248.132
Trying 192.168.248.132...
```

此时 dxq 也无法向 target 发起连接

3. 阻止 target 访问网站 120.79.40.129

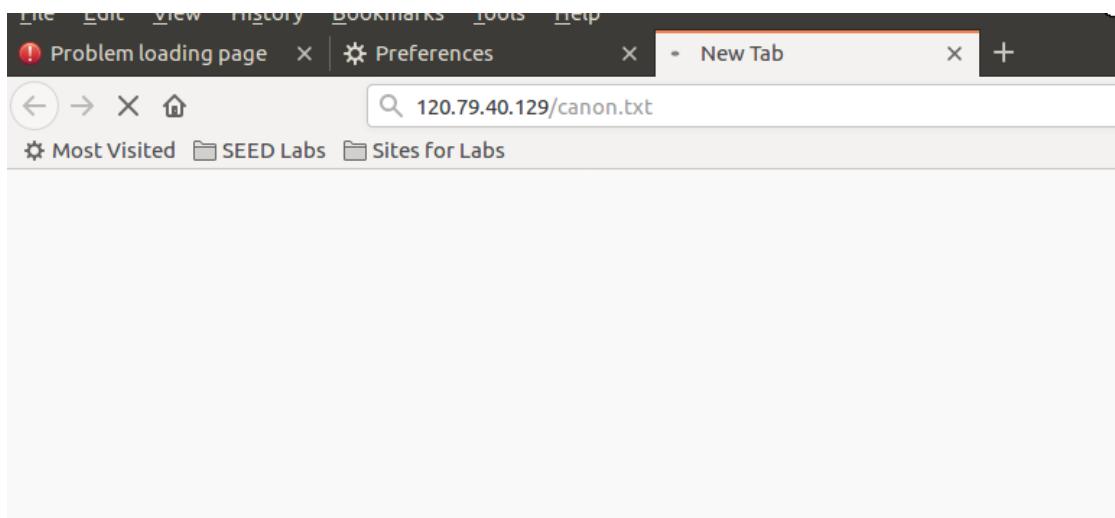
添加规则前正常访问

```
c3Ny0i8vZDNkM0xtZHziMmRzWlM1amIyMDZNvHBoZFhSb1gyTm9ZV2x1wDJFNLkyaGhZMmhoTwpBnmRHehpNUzR5WDNScFkydGxkRjloZfhSb09sbHVTbXha\UVLJczD84cFYZEtMEY2VwpCSktZHliM1Z3UFRWaExQTFza05RTlhCNU5qVmFLVFKkc3Ny0i8vZDNkM0xtZHziMmRzWlM1amIyMDZNanBoZFhSb1gyTm9zNt0TBiM0joY216dfBTWnlavzFoy210elBLWkLMWcxY0hsbu5Y0mxNaLphWLRBm04zbGhu3BCZVUxRE11ZFBvek11VFV0QmQw0VvieJpVkcSM1RXY21aM0rFl0WTJaa9uUnNjekv1Tw5MGFXtnJaWFjmWvhMGFEcFdNMnh3VGpKR2FD0F9iMkptYzNCaGntRnRQVTR5VlhT2VsVjRUv3BqZVU1RVZYVmxiV2hvWwpKdnUTBGd1RWUnd0R1Z1Y0dwTWjsWjFzbxmCZVU5dWEUmhivTEXwTBos2RrbEVUVFps0udeeFdyazFimkL6VGpBM04zbEp0V0zoUxpWd05tTTfjR1ZuTlhKUFZqkRKFVEvmlRMUExy0hrMK5WCD0V0Zwpzc3I6ly91bnayWkhNeExuaDRhbU10WkdSdwNSNTBhem8wTURBeE9trjFkr2hmWvdWek1USTRYMO5vWRFNllXvnpmVEkZkv1Z0YUdoaU1t0TFZvMmws5CewIzUnZjR0Z5MVcw0VRWUkpNMDFxvVRGUGJHeFdakkk1UkZKUkpuSmxiV0Z5YTNN0U5vD3RSVFUzTwxnMNWHrjJTVTLUTTkW1NdFpHuNvjeTuwYXpvME1EQxLPbUYxZEdoZLLXVnpNVEk0wDN0b1lURTZZV1Z6TFRJMUsMpabUK22Ed4ek1TNHlYMIJwTj0bGRGOWhkWFJv12xZemJQcCVVURlB3hXV2pJNVJGS1JKbKpsYldGeWezTTLOVxd0ULRVM0lsZzFjR0YyU105VE5ISmxAVGR3ZVR0c2Fu0mZ1bXLT0hsTFJFWTBTMUvtWjNKdmRYQTl07VCEII1Wn/rlwslv11CTEo5TEmlm11NmP4h0hnl11p5wHMSrEkvrgvkr117EhchaoQewVni1CF1PTut+hAv60Y77h1n6VAdGev1YMN111aknw7wle1N17V3F5hh111V+ct
```

添加规则

```
root@VM:/home/seed# ufw deny proto tcp from 192.168.248.132 to 120.79.40.129
Rule added
root@VM:/home/seed#
```

清空缓存，访问 120.79.40.129



访问失败

Task2: Implementing a Simple Firewall

进入过滤在 NF_IP_PRE_ROUTING、NF_IP_LOCAL_IN、NF_IP_FORWARD 放置监测点。外出过滤在 NF_IP_POST_ROUTING、NF_IP_LOCAL_OUT 放置监测点。

利用 LKM 和 Netfilter 实现防火墙规则

先将 ufw 关闭

尝试编写 LKM

```
#include <linux/module.h>
#include <linux/kernel.h>
#include <linux/init.h>

static int kmodule_init(void) {
    printk(KERN_INFO "Initializing this module\n");
    return 0;
}

static void kmodule_exit(void) {
    printk(KERN_INFO "Module cleanup\n");
}

module_init(kmodule_init);
module_exit(kmodule_exit);

MODULE_LICENSE("GPL");
~
```

编写 Makefile

```
obj-m := test.o

all:      make -C /lib/modules/$(shell uname -r)/build M=$(PWD) modules
clean:
make -C /lib/modules/$(shell uname -r)/build M=$(PWD) clean
~
```

编译模块

```
root@VM:/home/seed/code# make
make -C /lib/modules/4.8.0-36-generic/build M=/home/seed/code modules
make[1]: Entering directory '/usr/src/linux-headers-4.8.0-36-generic'
  CC [M]  /home/seed/code/test.o
  Building modules, stage 2.
  MODPOST 1 modules
  CC      /home/seed/code/test.mod.o
  LD [M]  /home/seed/code/test.ko
make[1]: Leaving directory '/usr/src/linux-headers-4.8.0-36-generic'
```

安装模块

```
root@VM:/home/seed/code# insmod test.ko
root@VM:/home/seed/code# lsmod | grep test
test                  16384  0
root@VM:/home/seed/code#
```

```
root@VM:/home/seed/code# dmesg
```

```
[65542.136100] Initializing this module
root@VM:/home/seed/code#
```

编写 Netfilter, drop 掉所有 telnet 报文

头文件

```
#include <linux/module.h>
#include <linux/kernel.h>
#include <linux/init.h>
#include <linux/skbuff.h>
#include <linux/ip.h>
#include <linux/tcp.h>
#include <linux/netfilter.h>
#include <linux/netfilter_ipv4.h>
```

定义 hook

```
static struct nf_hook_ops telnetFilterHook;

unsigned int telnetFilter(void *priv, struct sk_buff *skb,
                         const struct nf_hook_state *state)
{
    struct iphdr *iph;
    struct tcphdr *tcph;

    iph = ip_hdr(skb);
    tcph = (void *)iph+iph->ihl*4;

    if(iph->protocol == IPPROTO_TCP && tcph->dest == htons(23)){
        printk(KERN_INFO "Dropping telnet packet to %d.%d.%d.%d\n",
               ((unsigned char *)&iph->daddr)[0],
               ((unsigned char *)&iph->daddr)[1],
               ((unsigned char *)&iph->daddr)[2],
               ((unsigned char *)&iph->daddr)[3]);
        return NF_DROP;
    }else{
        return NF_ACCEPT;
    }
}
```

```
int setUpFilter(void) {
    printk(KERN_INFO "Registering a Telnet filter.\n");
    telnetFilterHook.hook = telnetFilter;
    telnetFilterHook.hooknum = NF_INET_POST_ROUTING;
    telnetFilterHook(pf = PF_INET;
    telnetFilterHook.priority = NF_IP_PRI_FIRST;

    //register the hook.
    nf_register_hook(&telnetFilterHook);
    return 0;
}

void removeFilter(void) {
    printk(KERN_INFO "Telnet filter is being removed.\n");
    nf_unregister_hook(&telnetFilterHook);
}

module_init(setUpFilter);
module_exit(removeFilter);
```

编译并安装

尝试进行 telnet 连接

```
root@VM:/home/seed/code# insmod telnetFilter.ko
root@VM:/home/seed/code# telnet 192.168.248.128
Trying 192.168.248.128...
```

用 dmesg 命令查看内核日志

```
[68905.386648] Registering a Telnet filter.
[68906.892171] Dropping telnet packet to 192.168.248.128
[68907.910807] Dropping telnet packet to 192.168.248.128
root@VM:/home/seed/code#
```

连接失败

卸载 telnetFilter, 重新尝试 telnet 连接

```
root@VM:/home/seed/code# rmmod telnetFilter
root@VM:/home/seed/code# telnet 192.168.248.128
Trying 192.168.248.128...
Connected to 192.168.248.128.
Escape character is '^]'.
Ubuntu 16.04.7 LTS
VM login:
```

连接迅速得到响应

Task3: Evading Egress Filtering

利用 ssh 隧道穿越防火墙

设置禁止向外建立 telnet 连接

```
root@VM:/home/seed/code# ufw deny out to any port 23
Skipping adding existing rule
Skipping adding existing rule (v6)
root@VM:/home/seed/code#
```

设置禁止访问 www.syr.com (128.230.18.200)

```
root@VM:/home/seed/code# ufw deny out to 128.230.18.200
Rule added
root@VM:/home/seed/code# ufw status
Status: active
```

To	Action	From
--	-----	-----
23	DENY OUT	Anywhere
128.230.18.200	DENY OUT	Anywhere
23 (v6)	DENY OUT	Anywhere (v6)

Task3.a Telnet to Machine B through the firewall

直接尝试 telnet 连接，失败

```
root@VM:/home/seed/code# telnet 192.168.248.128
Trying 192.168.248.128...
```

使用 ssh tunnel 绕过防火墙进行 telnet 连接。将本地端口 8000 与 192.168.248.132 建立 ssh 连接，通过 192.168.248.132 代理，向 192.168.248.132 发起 telnet 连接

```
root@VM:/home/seed/code# ssh -L 8000:192.168.248.128:23 seed@192.168.248.133
```

ssh 登录后，开启一个新终端

输入命令 telnet localhost 8000 进行 telnet 登录，登录成功。观察 wireshark 的数据

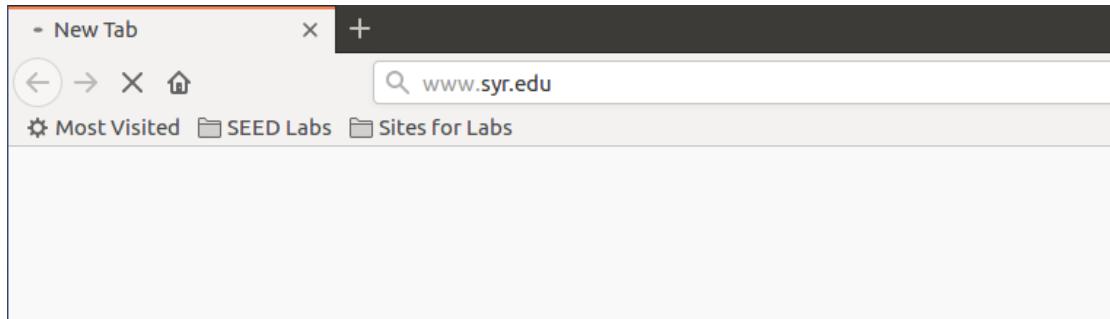
1 0.000000	192.168.248.132	192.168.248.133	SSH	166 Client: Encrypted packet (len=100)
2 0.000518	192.168.248.133	192.168.248.128	TCP	74 41310 → 23 [SYN] Seq=0 Win=29200 Len=0 MSS=1460 SACK_PERM=1 TSval=4891883 TSecr=0
3 0.000884	192.168.248.128	192.168.248.133	TCP	74 23 → 41310 [SYN, ACK] Seq=0 Ack=1 Win=65160 Len=0 MSS=1460 SACK_PERM=1 TSval=4891883 TSecr=0
4 0.001202	192.168.248.132	192.168.248.128	TCP	66 41310 → 23 [ACK] Seq=1 Ack=1 Win=29312 Len=0 TSval=4891883 TSecr=0
5 0.001298	192.168.248.133	192.168.248.132	SSH	110 Server: Encrypted packet (len=44)

可以看到 132 向 133 发送 ssh 数据，133 再将数据转发给 128。然后 128 与 133 通信，133 再将结果发送给 132。这样 132 就实现了通过隧道与 128 建立 telnet 连接。

Task3.b: Connect to Facebook using SSH Tunnel

使用动态端口转发技术

建立隧道前，连接失败



建立隧道

```
root@VM:/home/seed/code# ssh -D 9000 -C seed@192.168.248.133
```

设置浏览器将数据转发至 9000 端口

● Manual proxy configuration

HTTP Proxy	Port	0
<input type="checkbox"/> Use this proxy server for all protocols		
SSL Proxy	Port	0
FTP Proxy	Port	0
SOCKS Host	Port	9000

访问 www.syr.edu, 不再超时, 说明现在能够绕过防火墙访问 www.syr.edu, 但存在安全问题(其他原因导致)

The screenshot shows a Firefox browser window. The address bar displays <https://www.syracuse.edu>. The main content area shows an error message: "Secure Connection Failed". It states: "An error occurred during a connection to www.syracuse.edu. The OCSP response is not yet valid (contains a date in the future). Error code: SEC_ERROR_OCSP_FUTURE_RESPONSE". Below this, there are two bullet points: "The page you are trying to view cannot be shown because the authenticity of the received data could not be verified." and "Please contact the website owners to inform them of this problem." A "Learn more..." link is also present.

观察 wireshark 的数据流, 132 通过 133 的代理访问 128.230.18.200。

21 5.891209	192.168.248.132	192.168.248.133	SSH	102 Client: Encrypted p
22 5.891477	192.168.248.133	192.168.248.132	TCP	66 22 → 35856 [ACK] Se
23 5.891883	192.168.248.133	128.230.18.200	TCP	74 41722 → 443 [SYN] S

关闭 ssh 隧道, 代理随即关闭

The screenshot shows a Firefox browser window. The address bar displays <https://www.syracuse.edu>. The main content area shows an error message: "The proxy server is refusing connections". It states: "Firefox is configured to use a proxy server that is refusing connections." Below this, there are two bullet points: "Check the proxy settings to make sure that they are correct." and "Contact your network administrator to make sure the proxy server is working." A "Try Again" button is located at the bottom right.

Task4: Evading Ingress Filtering

添加规则，禁止 192.168.248.133 访问 192.168.248.132 的 80 端口和 22 端口

```
root@VM:/etc/bind# ssh seed@192.168.248.132
```

此时 192.168.248.133 向 192.168.248.132 发起 ssh 连接失败

在 192.168.248.132 设置反向代理，建立 192.168.248.133 的 8000 端口与 192.168.248.132 的 22 端口之间的隧道，也就是将 192.168.248.132 的 22 端口暴露给 192.168.248.133

```
root@VM:/home/seed/code# ssh -R 192.168.248.133:8000:192.168.248.132:22 seed@192.168.248.133
seed@192.168.248.133's password:
Welcome to Ubuntu 16.04.2 LTS (GNU/Linux 4.8.0-36-generic i686)

 * Documentation: https://help.ubuntu.com
 * Management: https://landscape.canonical.com
 * Support: https://ubuntu.com/advantage

1 package can be updated.
0 updates are security updates.

Last login: Thu Sep 17 21:51:46 2020 from 192.168.248.132
[09/17/20]seed@VM:~$
```

在 192.168.248.133 主机上向 8000 端口发起 ssh 连接

```
root@VM:/etc/bind# ssh -p 8000 seed@localhost
seed@localhost's password:
Welcome to Ubuntu 16.04.2 LTS (GNU/Linux 4.8.0-36-generic i686)

 * Documentation: https://help.ubuntu.com
 * Management: https://landscape.canonical.com
 * Support: https://ubuntu.com/advantage

1 package can be updated.
0 updates are security updates.

Last login: Fri Sep 11 09:13:30 2020 from 192.168.248.132
[09/11/20]seed@VM:~$ ip a | grep inet
    inet 127.0.0.1/8 scope host lo
        inet6 ::1/128 scope host
            inet 192.168.248.132/24 brd 192.168.248.255 scope global dynamic ens33
                inet6 fe80::5149:3b64%277:c7c4/64 scope link
[09/11/20]seed@VM:~$
```

成功通过反向代理向防火墙内的 192.168.248.132 发起 ssh 连接！