SEED LAB REPORT 6

Linux Firewall Exploration

刘熙达

57117232

Task 1: Using Firewall

a) Prevent A from doing telnet to B

```
[09/19/20]seed@VM:~$ sudo ufw enable
Firewall is active and enabled on system startup
[09/19/20]seed@VM:~$
```

```
[09/19/20]seed@VM:~$ telnet 192.168.255.128
Trying 192.168.255.128...
^Z^Z^C
[09/19/20]seed@VM:~$
```

1.在主机 B 开启防火墙，通过主机 A 尝试 telnet 连接主机 B，发现连接失败

b) Prevent B from doing telnet to A

```
[09/19/20]seed@VM:~$ sudo ufw enable
Firewall is active and enabled on system startup
```

```
[09/19/20]seed@VM:~$ telnet 192.168.255.129
Trying 192.168.255.129...
^C
```

2.在主机 A 开启防火墙，通过主机 B 尝试 telnet 连接主机 A，发现连接失败

c) Prevent A from visiting an external web site

```
[09/19/20]seed@VM:~$ sudo ufw deny out to 58.192.118.142
Rule added
[09/19/20]seed@VM:~$ ping seu.edu.cn
PING seu.edu.cn (58.192.118.142) 56(84) bytes of data.
ping: sendmsg: Operation not permitted
ping: sendmsg: Operation not permitted
ping: sendmsg: Operation not permitted
ping: sendmsg: Operation not permitted
^Z
[7]+  Stopped                 ping seu.edu.cn
[09/19/20]seed@VM:~$
```

1.在 A 的防火墙规则中添加 deny out to 58.192.118.42，即 seu.edu.cn 的 IP，再尝试 ping seu.edu.cn，显示 ping 指令 sendmsg 无权限，说明防火墙设置成功

Task 2: Implementing a Simple Firewall

```c
#include<linux/module.h>
#include<linux/kernel.h>
#include<linux/netfilter.h>
#include<linux/netfilter_ipv4.h>
#include<linux/ip.h>
#include<linux/tcp.h>
#include<linux/socket.h>
#include<linux/in.h>
#include<linux/inet.h>
#include<net/net_namespace.h>
/* This is the structure we shall use to register our function */
static struct nf_hook_ops nfho;
/* This is the hook function itself */
static unsigned char *drop_ip="\xc0\xa8\xff\x8c";

unsigned int hook_func(void *priv, struct sk_buff *skb,
const struct nf_hook_state *state)
{
struct iphdr *ip_header = (struct iphdr *)skb_network_header(skb);
unsigned int src_ip = (unsigned int)ip_header->saddr;
unsigned int dest_ip = (unsigned int)ip_header->daddr;
if(src_ip==drop_ip){
return NF_DROP;
}
return NF_ACCEPT;
}
/* Initialization routine */
int init_module()
{ /* Fill in our hook structure */
nfho.hook = hook_func; /* Handler function */
nfho.hooknum = NF_INET_PRE_ROUTING; /* First hook for IPv4 */
nfho.pf = PF_INET;
nfho.priority = NF_IP_PRI_FIRST; /* Make our function first */
nf_register_hook(&nfho);
return 0;
}
/* Cleanup routine */
void cleanup_module()
{
nf_unregister_hook(&nfho);
}
```

1.使用 Netfilter 构建简易防火墙，程序代码如上图。防火墙的拦截策略为对来自 192.168.255.128 的所有包进行拦截，其他包放行。

```makefile
obj-m += netfilter.o
 PWD :=$(shell pwd)
all:
	make -C /lib/modules/$(shell uname -r)/build M=$(PWD) modules
clean:
	make -C /lib/modules/$(shell uname -r)/build M=$(PWD) clean
```

2.编写 Makefile 文件

```
[09/19/20]seed@VM:~/lab7$ make
make -C /lib/modules/4.8.0-36-generic/build M=/home/seed/lab7 modules
make[1]: Entering directory '/usr/src/linux-headers-4.8.0-36-generic'
  CC [M]  /home/seed/lab7/netfilter.o
/home/seed/lab7/netfilter.c: In function 'hook_func':
/home/seed/lab7/netfilter.c:22:10: warning: comparison between pointer and integ
er
 if(src_ip==drop_ip){
          ^
/home/seed/lab7/netfilter.c:21:14: warning: unused variable 'dest_ip' [-Wunused-
variable]
 unsigned int dest_ip = (unsigned int)ip_header->daddr;
              ^
  Building modules, stage 2.
  MODPOST 1 modules
  CC      /home/seed/lab7/netfilter.mod.o
  LD [M]  /home/seed/lab7/netfilter.ko
make[1]: Leaving directory '/usr/src/linux-headers-4.8.0-36-generic'
```

```
[09/19/20]seed@VM:~/lab7$ ls
Makefile       Module.symvers  netfilter.ko     netfilter.mod.o
modules.order  netfilter.c     netfilter.mod.c  netfilter.o
[09/19/20]seed@VM:~/lab7$ sudo insmod netfilter.ko
```

3.运行 make 指令后得到 module，对新得到的 module 进行加载

```
root@kali:~# telnet 192.168.255.129
Trying 192.168.255.129...
```

4.在 IP 为 192.168.255.128 的主机尝试 telnet 连接设置了防火墙的主机，发现连接超时，说明防火墙实现了对 192.168.255.128 的包拦截的功能。

Task 3: Evading Egress Filtering

```
[09/19/20]seed@VM:~$ sudo ufw deny 23
Rule added
Rule added (v6)
[09/19/20]seed@VM:~$ sudo ufw status
Status: active

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

58.192.118.142             DENY OUT    Anywhere
```

1.在 IP 为 192.168.255.129 的主机中添加防火墙规则，拦截所有发往本机 23 端口的数据包

```
[09/19/20]seed@VM:~$ telnet 192.168.255.129
Trying 192.168.255.129...
^Z^Z
^C
```

2.在 192.168.255.128 中尝试使用 telnet 连接，发现链接失败

```
[09/19/20]seed@VM:~$ dig www.baidu.com

; <<>> DiG 9.10.3-P4-Ubuntu <<>> www.baidu.com
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 44818
;; flags: qr rd ra; QUERY: 1, ANSWER: 3, AUTHORITY: 0, ADDITIONAL: 1

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; MBZ: 0005 , udp: 4096
;; QUESTION SECTION:
;www.baidu.com.                 IN      A

;; ANSWER SECTION:
www.baidu.com.          5       IN      CNAME   www.a.shifen.com.
www.a.shifen.com.       5       IN      A       182.61.200.7
www.a.shifen.com.       5       IN      A       182.61.200.6

;; Query time: 4 msec
;; SERVER: 127.0.1.1#53(127.0.1.1)
;; WHEN: Sat Sep 19 12:20:00 EDT 2020
;; MSG SIZE  rcvd: 101
```

3.通过 dig 指令得到 www.baidu.com 的某个 IP 为 182.61.200.7

```
[09/19/20]seed@VM:~$ sudo ufw deny out to 182.61.200.7
Rule added
```

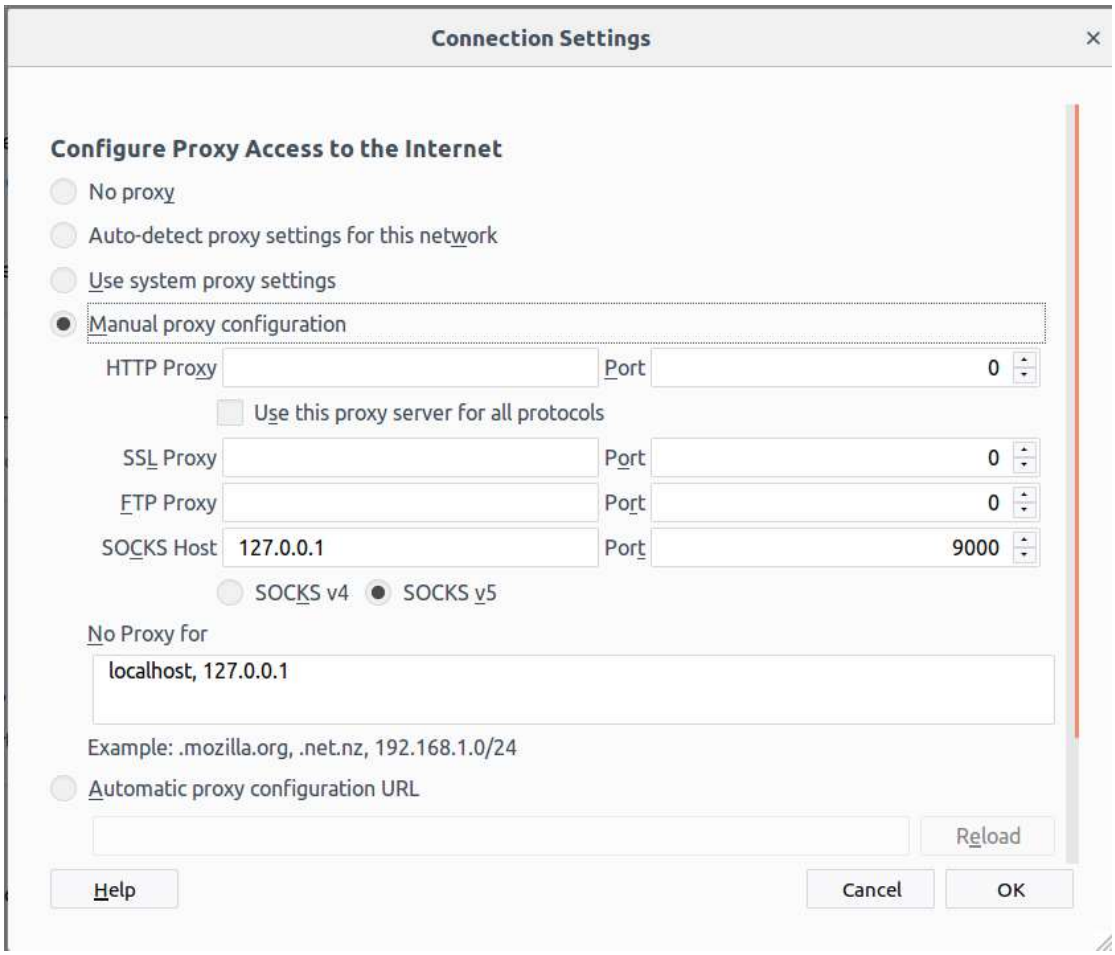
4.在 192.168.255.129 的主机中添加防火墙规则，拦截所有发往 182.61.200.7 的包

Task 3.a: Telnet to Machine B through the firewall

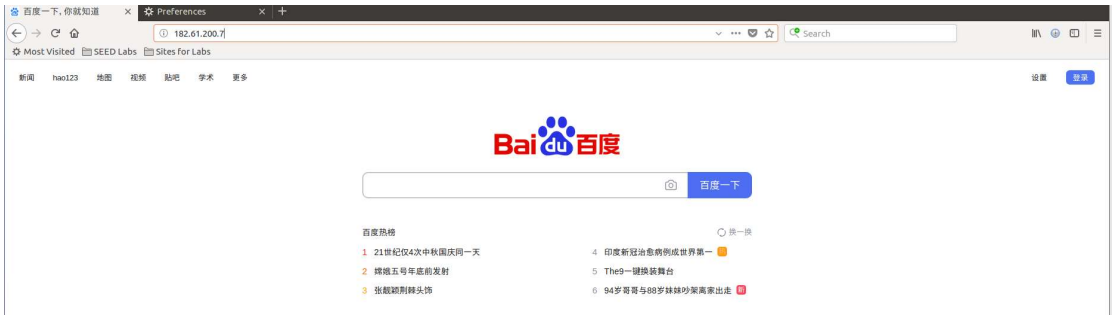
```
[09/19/20]seed@VM:~$ ssh -L 8000:192.168.255.129:23 seed@192.168.255.129
seed@192.168.255.129's password:
Welcome to Ubuntu 16.04.2 LTS (GNU/Linux 4.8.0-36-generic i686)

 * 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: Sat Sep 19 11:57:43 2020 from 192.168.255.128
[09/19/20]seed@VM:~$ telnet localhost 8000
Trying 127.0.0.1...
Connected to localhost.
Escape character is '^]'.
```

1.使用 ssh，首先连接 192.168.255.129:22 的 ssh 服务,建立 ssh 连接后，再通过 telnet 连接本机,实现穿墙 telnet

Task 3.b: Connect to Facebook using SSH Tunnel



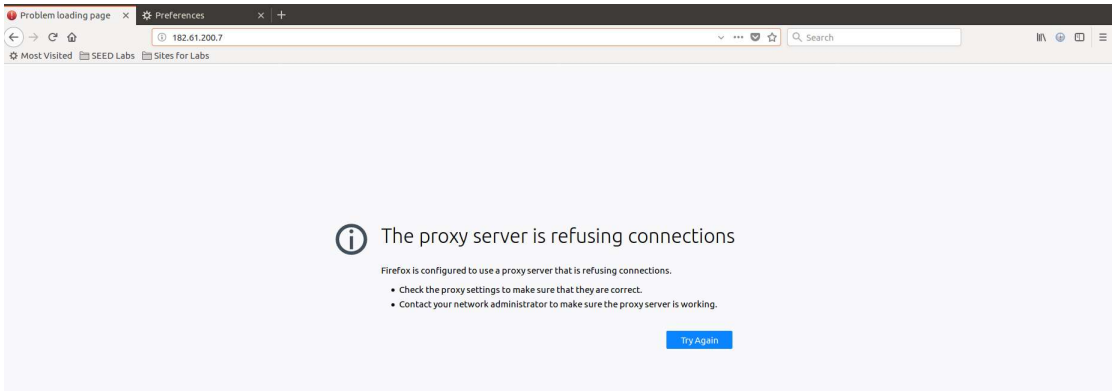
1.在 192.168.255.128 使用 ssh 连接 192.168.255.129，之后配置 128 主机的浏览器选项



2.通过浏览器直接访问之前被防火墙拦截的 182.61.200.7，发现可以访问

```
[09/19/20]seed@VM:~$ exit
logout
Connection to 192.168.255.129 closed.
```

3.关闭 128 到 129 的 ssh 连接



4.再次尝试访问 182.61.200.7，发现访问失败

Task 4: Evading Ingress Filtering

```
[09/19/20]seed@VM:~$ sudo ufw status
Status: active

To                         Action      From
--                         ------      ----
22                         ALLOW       Anywhere
192.168.255.129 22         DENY        192.168.255.128
192.168.255.129 80         DENY        192.168.255.128
22 (v6)                    ALLOW       Anywhere (v6)
```

1.在 129 主机中添加防火墙规则，屏蔽 128 对其 22 和 80 端口的访问

```
[09/19/20]seed@VM:~$ ssh -p 22 -qngfNTR 8000:localhost:22 seed@192.168.255.128
seed@192.168.255.128's password:
```

2.在 129 主机建立通往 128 的 ssh 反向连接

```
[09/19/20]seed@VM:~$ ssh -p 8000 seed@localhost
The authenticity of host '[localhost]:8000 ([127.0.0.1]:8000)' can't be establis
hed.
ECDSA key fingerprint is SHA256:p1zAio6c1bI+8HDp5xa+eKRi561aFDaPE1/xq1eYzCI.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added '[localhost]:8000' (ECDSA) to the list of known hosts
.
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
```

3.128 主机通过上一步的反向链接即可成功连接到内网的 129 主机