Lab6 实验报告

57117230 刘玉洁

Linux Firewall Exploration Lab

本实验需要三台虚拟机

Machine A ： 10.0.2.8

Machine B ： 10.0.2.9

Machine C ： 10.0.2.10

Task 1: Using Firewall

首先修改默认策略文件：

```
# 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"
```

- 阻止 A 对 B 做 telnet 连接

开启主机 B 的防火墙：

```
[09/16/20]seed@VM:~$ sudo ufw enable
Firewall is active and enabled on system startup
[09/16/20]seed@VM:~$ sudo ufw default deny
Default incoming policy changed to 'deny'
(be sure to update your rules accordingly)
```

主机 A 无法通过 telnet 连接到主机 B：

```
[09/16/20]seed@VM:~$ telnet 10.0.2.9
Trying 10.0.2.9...
```

关闭主机 B 的防火墙：

```
[09/16/20]seed@VM:~$ sudo ufw disable
Firewall stopped and disabled on system startup
```

主机 A 可以通过 telnet 连接到主机 B：

```
[09/16/20]seed@VM:~$ telnet 10.0.2.9
Trying 10.0.2.9...
Connected to 10.0.2.9.
Escape character is '^]'.
Ubuntu 16.04.2 LTS
```

- 阻止 B 对 A 做 telnet 连接

开启主机 A 的防火墙：

```
[09/16/20]seed@VM:~$ sudo ufw enable
Firewall is active and enabled on system startup
[09/16/20]seed@VM:~$ sudo ufw default deny
Default incoming policy changed to 'deny'
(be sure to update your rules accordingly)
```

主机 B 无法通过 telnet 连接到主机 A：

```
[09/16/20]seed@VM:~$ telnet 10.0.2.8
Trying 10.0.2.8...
```

关闭主机 A 的防火墙：

```
[09/16/20]seed@VM:~$ sudo ufw disable
Firewall stopped and disabled on system startup
```

主机 B 可以通过 telnet 连接到主机 A：

```
[09/16/20]seed@VM:~$ telnet 10.0.2.8
Trying 10.0.2.8...
Connected to 10.0.2.8.
Escape character is '^]'.
Ubuntu 16.04.2 LTS
```

- 阻止 A 访问外部网站

禁用主机 A 特定 IP 地址 58.192.118.142 ：

```
[09/16/20]seed@VM:~$ sudo ufw enable
Firewall is active and enabled on system startup
[09/16/20]seed@VM:~$ sudo ufw default deny
Default incoming policy changed to 'deny'
(be sure to update your rules accordingly)
[09/16/20]seed@VM:~$ sudo ufw deny out to 58.192.118.142
Skipping adding existing rule
[09/16/20]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
```

A 访问连接外网失败。

Task 2: Implementing a Simple Firewall

编写代码模块，拒绝目标 TCP 端口是 23 (telnet)的数据包，同时设置外出检测点 HOOK：NF_INET_POST_ROUTING ：

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

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;
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);
```

加载编写好的模块：

```
[09/17/20]seed@VM:~$ make
make -C /lib/modules/4.8.0-36-generic/build M=/home/seed modules
make[1]: Entering directory '/usr/src/linux-headers-4.8.0-36-generic'
  CC [M]  /home/seed/hook.o
  Building modules, stage 2.
  MODPOST 1 modules
  CC      /home/seed/hook.mod.o
  LD [M]  /home/seed/hook.ko
make[1]: Leaving directory '/usr/src/linux-headers-4.8.0-36-generic'
```

安装内核模块：

```
[09/17/20]seed@VM:~$ sudo insmod hook.ko
[09/17/20]seed@VM:~$ lsmod
Module                  Size  Used by
hook                   16384  0
ip6t_REJECT            16384  0
nf_reject_ipv6         16384  1 ip6t_REJECT
nf_log_ipv6            16384  0
xt_recent              20480  0
xt_hl                  16384  0
ip6t_rt                16384  0
nf_conntrack_ipv6      20480  0
nf_defrag_ipv6         24576  1 nf_conntrack_ipv6
ipt_REJECT             16384  0
nf_reject_ipv4         16384  1 ipt_REJECT
nf_log_ipv4            16384  0
nf_log_common          16384  2 nf_log_ipv6,nf_log_ipv4
xt_LOG                 16384  0
xt_limit               16384  0
xt_tcpudp              16384  0
xt_addrtype            16384  0
nf_conntrack_ipv4      16384  0
nf_defrag_ipv4         16384  1 nf_conntrack_ipv4
```

telnet 访问连接失败：

```
[09/17/20]seed@VM:~$ telnet 10.0.2.9
Trying 10.0.2.9...
^C
```

使用 dmesg 命令查看内核缓冲区，防火墙拒绝了本机的 telnet 访问：

```
[14487.082764] Dropping telnet packet to 10.0.2.9
[14488.101643] Dropping telnet packet to 10.0.2.9
[14490.172033] Dropping telnet packet to 10.0.2.9
[14494.181431] Dropping telnet packet to 10.0.2.9
[14502.373379] Dropping telnet packet to 10.0.2.9
[14518.520696] Dropping telnet packet to 10.0.2.9
```

移除内核模块，telnet 访问连接成功：

```
[09/17/20]seed@VM:~$  sudo rmmod hook.ko
[09/17/20]seed@VM:~$ telnet 10.0.2.9
Trying 10.0.2.9...
Connected to 10.0.2.9.
Escape character is '^]'.
Ubuntu 16.04.2 LTS
```

通过 LKM 向内核添加一个新的模块， Netfilter 设置外出检测点 HOOK，不需要重新编译整个内核即可实现防火墙功能。

Task 3: Evading Egress Filtering

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

上一个实验 Task2 已建立防火墙阻止所有向外部 telnet 服务器发送流量，为了绕过防火墙与 B 建立 telnet 连接，需要在主机 A 和主机 B 之间建立 SSH 隧道：

```
[09/17/20]seed@VM:~$ ssh -l 8000:10.0.2.10:23 seed@10.0.2.9
The authenticity of host '10.0.2.9 (10.0.2.9)' can't be established.
ECDSA key fingerprint is SHA256:p1zAio6c1bI+8HDp5xa+eKRi561aFDaPE1/xq1eYzCI.
Are you sure you want to continue connecting (yes/no)? y
Please type 'yes' or 'no': yes
Warning: Permanently added '10.0.2.9' (ECDSA) to the list of known hosts.
seed@10.0.2.9'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

0 packages can be updated.
0 updates are security updates.


The programs included with the Ubuntu system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by
applicable law.
```

wireshark 检测到主机 A 和主机 B 之间的 telnet 数据包：

```
10.0.2.8        10.0.2.9        TELNET    107 Telnet Data ...
10.0.2.9        10.0.2.8        TCP        68 38442 → 23 [ACK] Seq=2610422591 Ack=3502358
10.0.2.9        10.0.2.8        TELNET    143 Telnet Data ...
10.0.2.8        10.0.2.9        TCP        68 23 → 38442 [ACK] Seq=3502358689 Ack=2610422
10.0.2.8        10.0.2.9        TELNET     71 Telnet Data ...
10.0.2.9        10.0.2.8        TELNET     71 Telnet Data ...
10.0.2.8        10.0.2.9        TELNET     71 Telnet Data ...
10.0.2.9        10.0.2.8        TELNET     71 Telnet Data ...
10.0.2.8        10.0.2.9        TELNET     88 Telnet Data ...
```

由此，通过建立 SSH 隧道，可以绕过防火墙实现 telnet 连接。

Task 3.b: Connect to Facebook using SSH Tunnel

首先通过 ufw 工具禁止主机 A 访问 www.baidu.com：

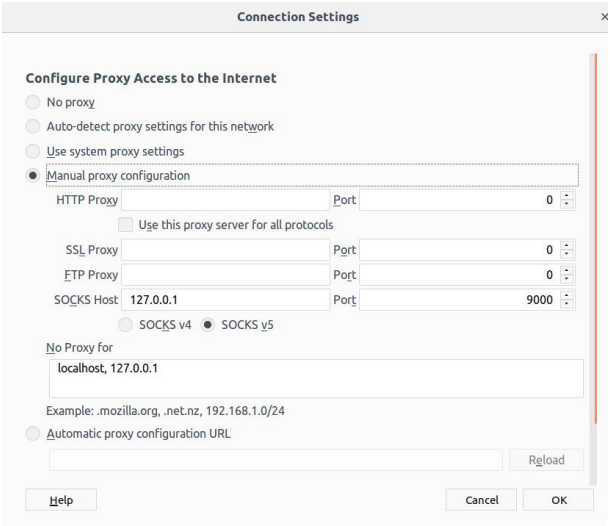
```
;; ANSWER SECTION:
www.baidu.com.          930     IN      CNAME   www.a.shifen.com.
www.a.shifen.com.       48      IN      A       112.80.248.75
www.a.shifen.com.       48      IN      A       112.80.248.76
```

```
[09/17/20]seed@VM:~$ sudo ufw deny out to 112.80.248.75
Rule added
[09/17/20]seed@VM:~$ sudo ufw deny out to 112.80.248.76
Rule added
[09/17/20]seed@VM:~$ sudo ufw enable
Firewall is active and enabled on system startup
[09/17/20]seed@VM:~$ ping www.baidu.com
PING www.a.shifen.com (112.80.248.76) 56(84) bytes of data.
ping: sendmsg: Operation not permitted
ping: sendmsg: Operation not permitted
ping: sendmsg: Operation not permitted
```

建立主机 A 和主机 B 之间的 SSH 隧道，在主机 A 上即可访问 www.baidu.com：

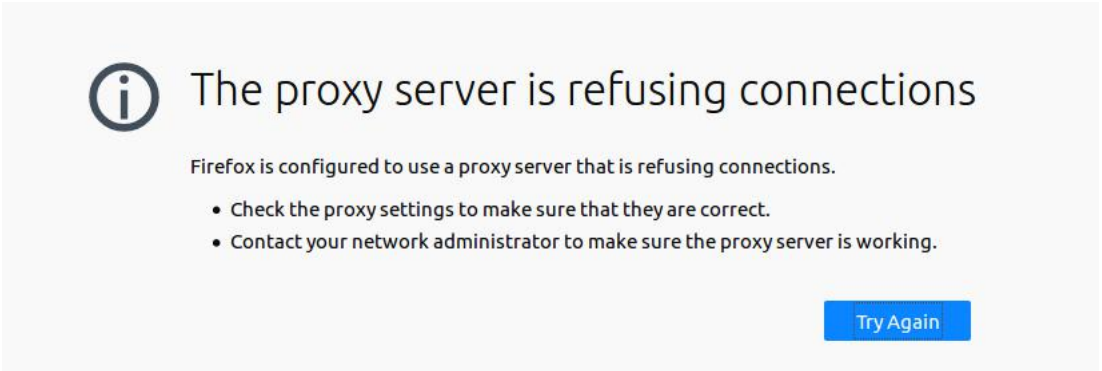
```
[09/17/20]seed@VM:~$ ssh -D 9000 -C seed@10.0.2.9
seed@10.0.2.9's password:
Welcome to Ubuntu 16.04.2 LTS (GNU/Linux 4.8.0-36-generic i686)

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

0 packages can be updated.
0 updates are security updates.

Last login: Thu Sep 17 07:06:03 2020 from 10.0.2.8
[09/17/20]seed@VM:~$ ping www.baidu.com
PING www.a.shifen.com (112.80.248.75) 56(84) bytes of data.
64 bytes from 112.80.248.75: icmp_seq=1 ttl=53 time=42.4 ms
64 bytes from 112.80.248.75: icmp_seq=2 ttl=53 time=29.7 ms
```

对浏览器进行连接设置：



在浏览器可以访问 www.baidu.com：

断开 SSH 隧道，清除浏览器缓存，则该网站无法访问：



重新建立 SSH 隧道，可以重新访问：

使用 wireshark 检测主机 A 和主机 B 之间的 TCP 数据包：

10.0.2.8	10.0.2.9	SSH	104 Client: Encrypted packet (len=36)
10.0.2.9	10.0.2.8	SSH	112 Server: Encrypted packet (len=44)
10.0.2.8	10.0.2.9	TCP	68 42020 → 22 [ACK] Seq=1364276423 Ack=1757065
10.0.2.8	10.0.2.9	SSH	104 Client: Encrypted packet (len=36)
10.0.2.9	10.0.2.8	SSH	104 Server: Encrypted packet (len=36)

主机 B 相当于一个中介，事实上是主机 B 去访问 www.baidu.com，然后 www.baidu.com 返回一些 TCP 数据包给主机 B，主机 B 再返回一些 SSH 数据给主机 A，最后实现了主机 A 虽然被禁止访问却通过 SSH 隧道访问到了 www.baidu.com。

Task 4: Evading Ingress Filtering

首先关闭主机 A 的 22 和 80 端口，拒绝主机 B 的 SSH 和 WEB 访问：

```
[09/17/20]seed@VM:~$ sudo ufw deny 22
Rule added
Rule added (v6)
[09/17/20]seed@VM:~$ sudo ufw deny 80
Rule added
Rule added (v6)
[09/17/20]seed@VM:~$ sudo ufw enable
Firewall is active and enabled on system startup
```

此时，wireshark 检测到主机 A 和主机 B 之间的 tcp 数据包无法正常通信：

10.0.2.8	10.0.2.9	TCP	68 33644 → 23 [ACK] Seq=1041548680 Ack=339801
10.0.2.8	10.0.2.9	TCP	68 33644 → 23 [FIN, ACK] Seq=1041548680 Ack=3
10.0.2.9	10.0.2.8	TCP	68 23 → 33644 [ACK] Seq=33980167 Ack=10415486
10.0.2.9	10.0.2.8	TCP	76 44730 → 23 [SYN] Seq=3161615497 Win=29200
::1	::1	UDP	64 60211 → 51177 Len=0
10.0.2.9	10.0.2.8	TCP	76 [TCP Retransmission] 44730 → 23 [SYN] Seq=
10.0.2.9	10.0.2.8	TCP	76 [TCP Retransmission] 44730 → 23 [SYN] Seq=
10.0.2.9	10.0.2.8	TCP	76 [TCP Retransmission] 44730 → 23 [SYN] Seq=

为了能使主机 B 访问主机 A，需要使用反向 SSH 隧道技术，主机 A 通过 SSH 的 2222 端口连接主机 B:

```
[09/17/20]seed@VM:~$ ssh -NfR 2222:localhost:22 seed@10.0.2.9
seed@10.0.2.9's password:
[09/17/20]seed@VM:~$ Warning: remote port forwarding failed for listen port 2222
```

在主机 B 上使用如下命令进行确认，输入主机 A 的密码则访问成功：

```
[09/17/20]seed@VM:~$ ssh localhost -p 2222
The authenticity of host '[localhost]:2222 ([127.0.0.1]:2222)' 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]:2222' (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

0 packages can be updated.
0 updates are security updates.

Last login: Wed Sep 16 22:08:44 2020 from 10.0.2.9
```

使用 wireshark 检测主机 A 和主机 B 之间的访问流量：

```
10.0.2.9        10.0.2.8        TCP       68 22 → 42280 [ACK] Seq=2298274738 Ack=3667024
10.0.2.8        10.0.2.9        SSHv2    360 Client: Encrypted packet (len=292)
10.0.2.9        10.0.2.8        TCP       68 22 → 42280 [ACK] Seq=2298274738 Ack=3667024
127.0.0.1       127.0.0.1       SSHv2    128 Server: Encrypted packet (len=60)
127.0.0.1       127.0.0.1       TCP       68 33590 → 22 [ACK] Seq=1611240819 Ack=2518837
10.0.2.8        10.0.2.9        SSHv2    168 Client: Encrypted packet (len=100)
10.0.2.9        10.0.2.8        TCP       68 22 → 42280 [ACK] Seq=2298274738 Ack=3667024
10.0.2.9        10.0.2.8        SSHv2    176 Server: Encrypted packet (len=108)
```

在主机 B 上查询 IP 地址，得到的 IP 地址为主机 A 的地址 10.0.2.8，说明远程登录成功：

```
[09/17/20]seed@VM:~$ ifconfig
enp0s3    Link encap:Ethernet  HWaddr 08:00:27:a9:a3:fd
          inet addr:10.0.2.8  Bcast:10.0.2.255  Mask:255.255.255.0
          inet6 addr: fe80::6245:239e:6f5b:cc67/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:3746 errors:0 dropped:0 overruns:0 frame:0
          TX packets:2760 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:2316166 (2.3 MB)  TX bytes:407070 (407.0 KB)

lo        Link encap:Local Loopback
          inet addr:127.0.0.1  Mask:255.0.0.0
          inet6 addr: ::1/128 Scope:Host
          UP LOOPBACK RUNNING  MTU:65536  Metric:1
          RX packets:6321 errors:0 dropped:0 overruns:0 frame:0
          TX packets:6321 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1
          RX bytes:2677253 (2.6 MB)  TX bytes:2677253 (2.6 MB)
```

利用反向 SSH 隧道技术，可以实现外网访问内网主机。