Lab6-report

57117134-张家康

Task 1: Using Firewall

这里需要用到2台主机：

```
192.168.43.236      //主机A的IP地址
192.168.43.177      //主机B的IP地址
```

1.1 阻止 A， B 之间进行 telnet 连接

未进行任何操作时，A 和 B 可以互相连接：

```
[09/18/20]seed@VM:~$ telnet 192.168.43.177
Trying 192.168.43.177...
Connected to 192.168.43.177.
Escape character is '^]'.
Ubuntu 16.04.6 LTS
user-VirtualBox login: user
Password:
Last login: Fri Sep 11 19:26:40 CST 2020 from user-VirtualBox on pts/18
Welcome to Ubuntu 16.04.6 LTS (GNU/Linux 4.15.0-117-generic x86_64)

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

76 个可升级软件包。
0 个安全更新。

*** 需要重启系统 ***
user@user-VirtualBox:~$
```

```
user@user-VirtualBox:~$ telnet 192.168.43.236
Trying 192.168.43.236...
Connected to 192.168.43.236.
Escape character is '^]'.
Ubuntu 16.04.2 LTS
VM login: seed
Password:
/usr/lib/update-notifier/update-motd-fsck-at-reboot:[:59: integer expression exp
ected:            0
Welcome to Ubuntu 16.04.2 LTS (GNU/Linux 4.8.0-36-generic i686)
```

在主机 A 处使用 `sudo ufw enable` 开启防火墙，此时主机B无法对A进行 `telnet` 连接，如下图所示：

```
user@user-VirtualBox:~$ telnet 192.168.43.236
Trying 192.168.43.236...
```

如上图所示，很长时间没有响应，连接失败。

同理，在主机B处使用 `sudo ufw enable` 开启防火墙，此时主机A无法对B进行 `telnet` 连接，如下图所示：

```
[09/18/20]seed@VM:~$ telnet 192.168.43.177
Trying 192.168.43.177...
```

同样，很长时间没有响应，连接失败。

1.2 阻止 A 访问某个外部网页

首先， `ping www.iqiyi.com` ，结果如下图所示：

```
[09/18/20]seed@VM:~$ ping www.iqiyi.com
PING ipv6-static.dns.iqiyi.com (112.29.146.149) 56(84) bytes of data.
64 bytes from 112.29.146.149: icmp_seq=1 ttl=52 time=39.9 ms
64 bytes from 112.29.146.149: icmp_seq=2 ttl=52 time=64.6 ms
64 bytes from 112.29.146.149: icmp_seq=3 ttl=52 time=58.2 ms
64 bytes from 112.29.146.149: icmp_seq=4 ttl=52 time=61.7 ms
64 bytes from 112.29.146.149: icmp_seq=5 ttl=52 time=55.5 ms
```

然后，使用命令 `sudo ufw deny out to 112.29.146.149` 添加规则，拒绝访问，再次 `ping 112.29.146.149` ，结果如下图所示：

```
[09/18/20]seed@VM:~$ ping 112.29.146.149
PING 112.29.146.149 (112.29.146.149) 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
ping: sendmsg: Operation not permitted
ping: sendmsg: Operation not permitted
ping: sendmsg: Operation not permitted
ping: sendmsg: Operation not permitted
```

最后，使用命令 `sudo ufw status` 查看防火请规则，如下图所示：

```
[09/18/20]seed@VM:~$ sudo ufw status
Status: active

To                         Action      From
--                         ------      ----
112.29.146.149             DENY OUT    Anywhere
```

Task 2: Implementing a Simple Firewall

使用 `LKM` 和 `Netfilter` 来实现包过滤模块。

在监测点 `NF_INET_PRE_ROUTING` 设置阻止主机 B 的任何访问。新建文件 `hook.c` ，写入代码如下：

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

static struct nf_hook_ops nfho;

static unsigned char *drop_ip = "\xc0\xa8\x2b\xb1";

```

```c
unsigned int hook_func(unsigned int hooknum,
                struct sk_buff **skb,
                const struct net_device *in,
                const struct net_device *out,
                int (*okfn)(struct sk_buff *))
{
    struct sk_buff *sb = *skb;
    if(ip_hdr(sb)->saddr == *(unsigned int *)drop_ip)
    {
        return NF_DROP;
    }else{
        return NF_ACCEPT;
    }
}

int init_module()
{
    nfho.hook = (nf_hookfn *)hook_func;
    nfho.hooknum = NF_INET_PRE_ROUTING;
    nfho.pf = PF_INET;
    nfho.priority = NF_IP_PRI_FIRST;
    nf_register_hook(&nfho);

    return 0;
}

```

其中，`*drop_ip`为组织访问的主机的IP地址，这里填入`192.168.43.177`的16进制形式。

编写`makefile`，代码如下：

```
obj-m :=hook.o

KERNELDIR?=/lib/modules/$(shell uname -r)/build/
PWD :=$(shell pwd)

default:
    $(MAKE) -C $(KERNELDIR) M=$(PWD) modules
clean:
    $(MAKE) -C $(KERNELDIR) M=$(PWD) clean

```

编译，如下图所示：

```
[09/18/20]seed@VM:~$ sudo 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'
[09/18/20]seed@VM:~$
```

输入`sudo insmod hook.ko`，可见在内核中添加了一块新的`hook`，如下图所示：

```
[09/18/20]seed@VM:~$ sudo insmod hook.ko
[09/18/20]seed@VM:~$ lsmod
Module                  Size  Used by
hook                   16384  0
```

此时，主机B就无法访问到主机A，如下图所示：

```
user@user-VirtualBox:~$ ping 192.168.43.236
PING 192.168.43.236 (192.168.43.236) 56(84) bytes of data.
^C
--- 192.168.43.236 ping statistics ---
60 packets transmitted, 0 received, 100% packet loss, time 60422ms
```

最后，使用 `sudo rmmod hook.ko` 将其移出，再次使用主机B访问主机A，成功连接，如下图所示：

```
user@user-VirtualBox:~$ ping 192.168.43.236
PING 192.168.43.236 (192.168.43.236) 56(84) bytes of data.
64 bytes from 192.168.43.236: icmp_seq=1 ttl=64 time=0.768 ms
64 bytes from 192.168.43.236: icmp_seq=2 ttl=64 time=0.535 ms
64 bytes from 192.168.43.236: icmp_seq=3 ttl=64 time=0.399 ms
64 bytes from 192.168.43.236: icmp_seq=4 ttl=64 time=0.388 ms
64 bytes from 192.168.43.236: icmp_seq=5 ttl=64 time=0.446 ms
```

task 3 Evading Egress Filtering

本实验需要3台主机，分别如下所示：

```
192.168.43.236          //主机A，已禁止23端口
192.168.43.79           //主机B，不设防火墙
192.168.43.177          //主机C，作为telnet服务器
```

主机A上，已经禁止23端口的访问，如下图所示：

```
[09/18/20]seed@VM:~$ sudo ufw status
Status: active

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

112.29.146.149             DENY OUT    Anywhere
```

3.1 Telnet to Machine B through the firewall

令主机 A 穿过防火墙对主机 B 进行 telnet 访问：

首先，确保已经安装并开启了telnet服务器（可以使用 `sudo apt-get install telnetd` 和 `sudo /etc/init.d/openbsd-inetd restart` 安装和开启）

主机 A 向 C 发起 SSH 访问请求，然后以其为跳板 telnet 访问 B。如下图所示：

```
[09/18/20]seed@VM:~$ ssh user@192.168.43.177
user@192.168.43.177's password:
Welcome to Ubuntu 16.04.6 LTS (GNU/Linux 4.15.0-117-generic x86_64)

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

76 个可升级软件包。
0 个安全更新。

New release '18.04.5 LTS' available.
Run 'do-release-upgrade' to upgrade to it.

Last login: Fri Sep 18 17:52:35 2020 from 192.168.43.236
user@user-VirtualBox:~$ telnet 192.168.43.79
Trying 192.168.43.79...
Connected to 192.168.43.79.
Escape character is '^]'.
Ubuntu 16.04.6 LTS
user-VirtualBox login: user
Password:
Welcome to Ubuntu 16.04.6 LTS (GNU/Linux 4.15.0-117-generic i686)
```

3.2 Connect to Facebook using SSH Tunnel

使用 SSH 请求连接一个被禁止访问的网址，这里以 `112.29.146.149` 为例：

如图所示， `112.29.146.149` 被主机*A*所禁止：

```
[09/18/20]seed@VM:~$ sudo ufw status
Status: active

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

112.29.146.149             DENY OUT    Anywhere
```

通过 SSH 通道将主机 B 作为跳板 ， 然后可以顺利通信，如下图所示：

```
[09/18/20]seed@VM:~$ ssh user@192.168.43.177
user@192.168.43.177's password:
Welcome to Ubuntu 16.04.6 LTS (GNU/Linux 4.15.0-117-generic x86_64)

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

76 个可升级软件包。
0 个安全更新。

New release '18.04.5 LTS' available.
Run 'do-release-upgrade' to upgrade to it.

Last login: Fri Sep 18 18:00:42 2020 from 192.168.43.236
user@user-VirtualBox:~$ ping 112.29.146.149
PING 112.29.146.149 (112.29.146.149) 56(84) bytes of data.
64 bytes from 112.29.146.149: icmp_seq=1 ttl=52 time=79.8 ms
64 bytes from 112.29.146.149: icmp_seq=2 ttl=52 time=37.4 ms
64 bytes from 112.29.146.149: icmp_seq=3 ttl=52 time=35.6 ms
```

退出SSH后，又不能访问，如下图所示：

```
[09/18/20]seed@VM:~$ ping  112.29.146.149
PING 112.29.146.149 (112.29.146.149) 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
ping: sendmsg: Operation not permitted
```

Task 4: Evading Ingress Filtering

本实验需要三台主机，相关信息如下所示：

```
192.168.43.236      //主机A
192.168.43.177      //主机B
192.168.43.200      //主机C
```

首先，在主机A上，禁止所有对22端口和80端口的外部访问（使用命令`sudo ufw deny ssh`和`sudo ufw deny http`），查看防火墙规则，如下图所示：

```
[09/18/20]seed@VM:~$ sudo ufw status
Status: active

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

112.29.146.149             DENY OUT    Anywhere
```

在主机A上，使用命令：

```
ssh -f -N -R 10000:localhost:22 user@192.168.43.177
```

反向连接主机B，如下图所示：

```
[09/18/20]seed@VM:~$ ssh -f -N -R 10000:localhost:22 user@192.168.43.177
user@192.168.43.177's password:
[09/18/20]seed@VM:~$
```

在主机 B 上 做正向代理，用来做转发，具体指令为：

```
ssh -fCNL *:10001:localhost:10000 user@localhost
```

自此，`10001`端口为本地转发端口，负责和外网进行通信，并将数据转发的`10000`端口，实现了可以从其他机器访问的功能。同时，`*`号表示可以接受任何IP的访问。

最后，从主机C上，使用命令：

```
ssh -p 10001 user@192.168.43.177
```

成功连接到主机A，如下图所示：