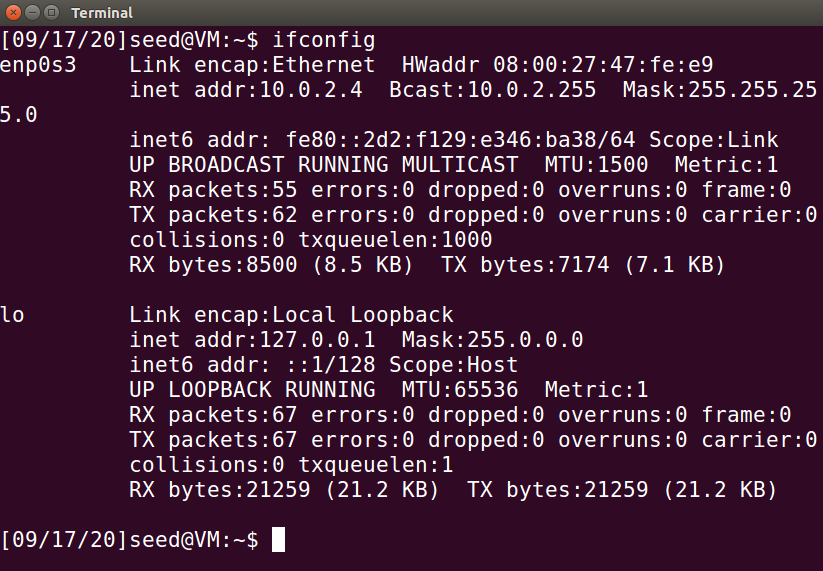
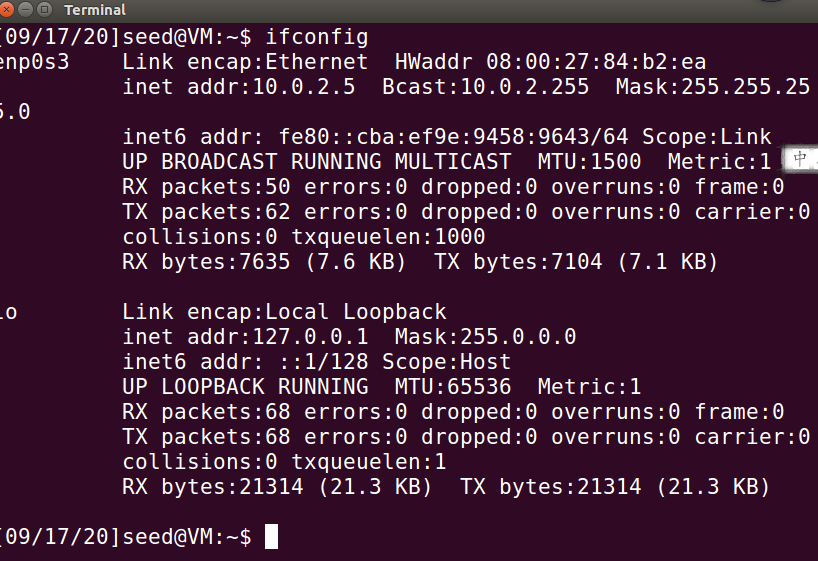
**实验六：Linux Firewall Exploration Lab**

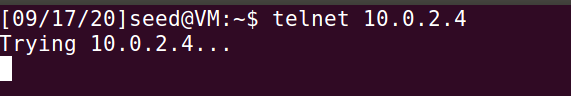
57117222 舒卓卓

**任务一：Using Firewall**

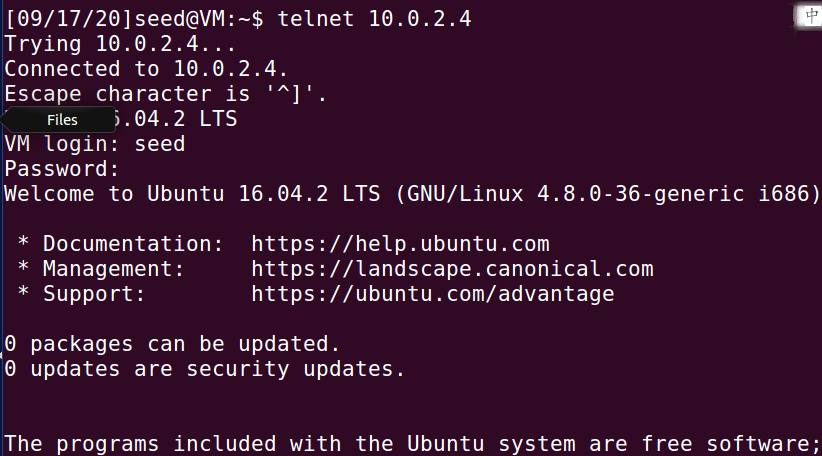
1. 在这里我们配置两个虚拟机，其ip地址分别为A:10.0.2.4和B:10.0.2.5：



1. 阻止A对B发起telnet：



1. 然后在A中使用命令sudo ufw disable，在B中连接A，发现成功，同理可以完成B对A的阻止访问操作：



1. 阻止A访问特定的网页，在这里我们查询到了百度的ip地址为：180.101.49.11，因此我们使用命令添加过滤规则，可以查看到规则被更新，使用ping命令发现行不通：



**任务二：Implementing a Single Firewall**

1. 我们构造如下的代码：

#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;  
/\*IP address we want to drop from\*/  
static unsigned char \*drop\_ip = "\x0a\x00\x02\x05";

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 = hook\_func;

nfho.hooknum = NF\_INET\_PRE\_ROUTING;

nfho. pf = PF\_INET;

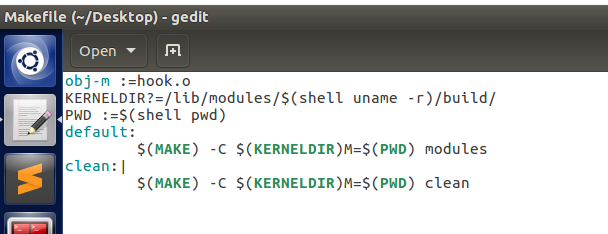
nfho.priority =NF\_IP\_PRI\_FIRST;

nf\_register\_hook(&nfho);

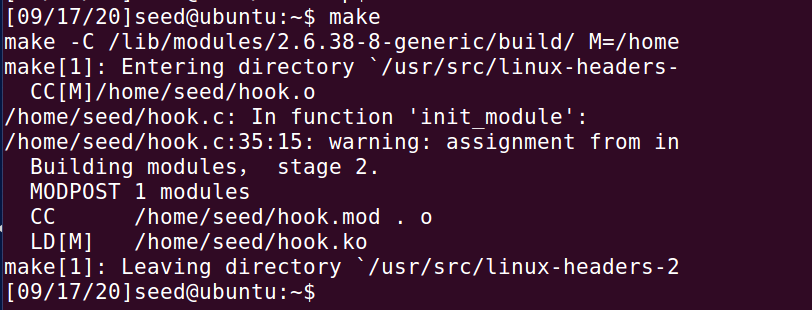
return 0 ;

}

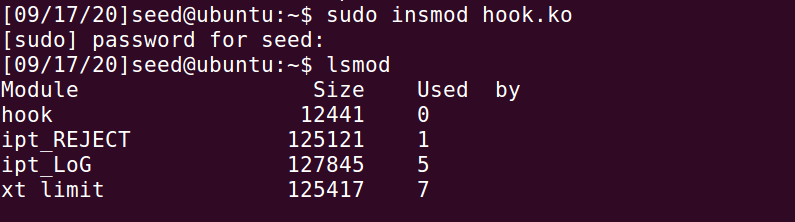
1. 构造makefile文件：

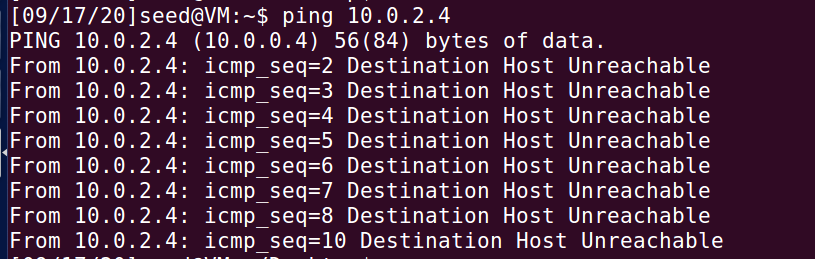


1. 运行对应防火墙程序：



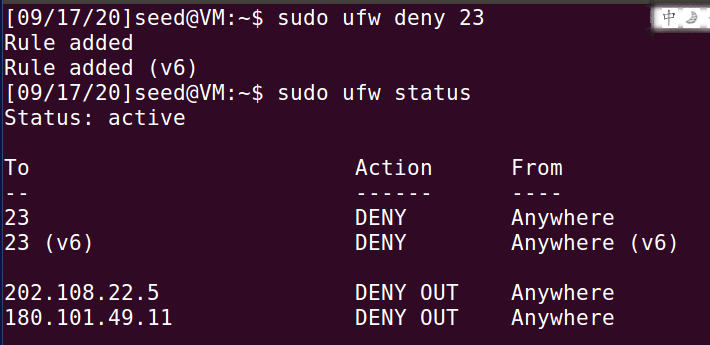
1. 所得结果如图所示：





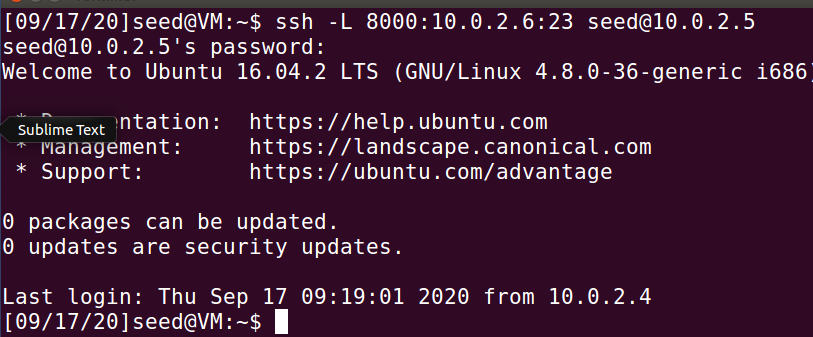
**任务三：Evading Egress Filtering**

1. 设置三台虚拟机A:10.0.2.4 B:10.0.2.5 C:10.0.2.6
2. A设置在防火墙后，并且禁止所有的23端口访问：

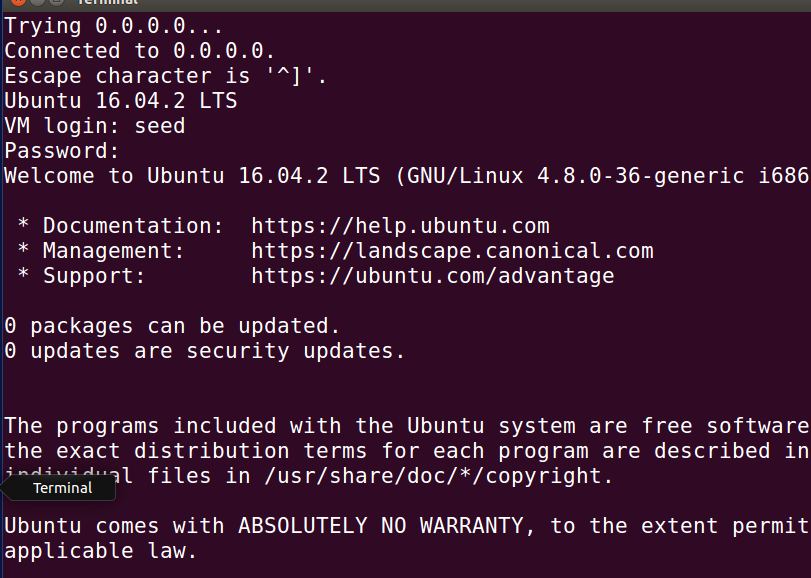


**任务三.a: Telnet to Machine B through the firewall**

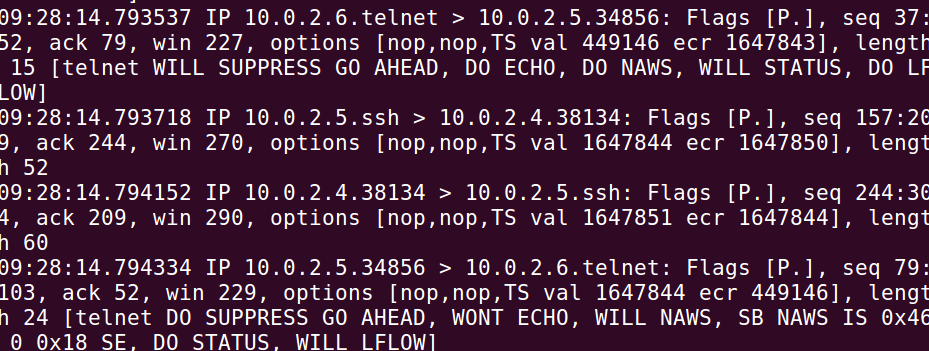
1. 在A中向B发起ssh请求，以B为跳板访问C：



1. 发现连接成功

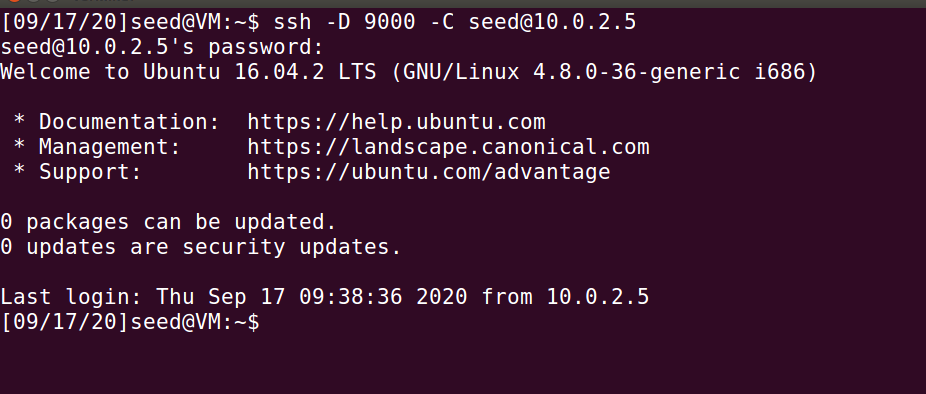


1. 我们在B虚拟机中提前开启了tcpdump查看到了对应的结果

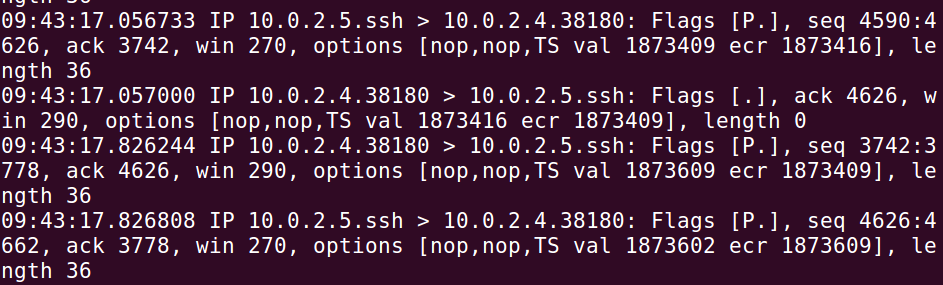


**任务三.b: Connect to Facebook using SSH Tunnel**

1. 选择禁止访问百度[www.baidu.com](http://www.baidu.com)
2. 使用命令建立对应的中介连接：



1. 因此B相当于中介，事实是B去访问baidu，然后百度返回一些tcp数据包，b再返回一些ssh数据给A，A虽然被禁止访问baidu，但是通过SSH，访问到了baidu

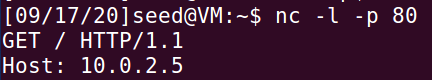


**任务四：Evading Ingress Filtering**

1. 与上文操作配置相似，只不过在A机器中使用命令：

ssh -fCNR 10.0.2.6:2333:10.0.2.5:2334 [root@10.0.2.6](mailto:root@10.0.2.6)

1. 对B的2333端口发起反向ssh隧道
2. 相似的，我们可以在A中看到：



1. 说明A虽然阻止了外部访问80端口，但是B通过建立的反向ssh，成功访问到了A内部的80端口。