



Universidade do Minho  
Escola de Engenharia

---

**Universidade do Minho**  
**Trabalho Prático 5**  
**Síntese das Tecnologias de Segurança em Redes**

---

1º Semestre – 2019/2020

Bruno Rodrigues Pg41066

Carlos Alves Pg41840

Paulo Bento a81139

## ÍNDICE

---

Introdução.....	3
Tarefa 1 .....	4
1.1    Alínea 1.....	4
1.2    Alínea 2.....	6
1.3    Alínea 3.....	6
1.3.1    Alínea 3 – A.....	6
1.3.2    Alínea 3 – B.....	7
1.4    Alínea 4.....	7
1.5    Alínea 5.....	8
2    Tarefa 2 .....	9
2.1    Alínea 1.....	10
2.2    Alínea 2.....	10
2.3    Alínea 3.....	11
2.4    Alínea 4.....	11
2.5    Alínea 5.....	11
3    Tarefa 3 .....	12
3.1    Alínea 1.....	12
3.2    Alínea 2.....	13
3.3    Alínea 3.....	14
3.4    Alínea 4.....	14
3.5    Alínea 5.....	15
3.6    Alínea 6.....	15
3.7    Alínea 7.....	16
4    Tarefa 3 – Melhorada - Conclusão .....	17
5    Conclusão .....	23
6    Referências.....	24

## INTRODUÇÃO

---

Este trabalho prático, sugerido em aula na unidade curricular Segurança em Redes tem como objetivo aplicar o conhecimento adquirido nas aulas sobre o tema - Síntese das Tecnologias de Segurança em Redes, como principal foco a configuração de *firewalls* de modo a atingir algum nível de segurança.

Isto através do *Kernel* do *Linux*, processar/filtrar todo o tráfego que passa na pilha de protocolos, neste trabalho usaremos essa funcionalidade para configurar uma máquina como firewall. Em especial o uso da **tabela filter – IPTables** – onde todos os pacotes são sujeitos a uma de três cadeias primárias de regras de cadeia de INPUT, FORWARD e OUTPUT.

Inicialmente foi necessário preparar o ambiente para realizar o trabalho, foi então necessário instalar o *Kali Linux* – cliente – e o sistema *CentOS 6.1*, usado como servidor. Foram encontrados alguns problemas, principalmente com o sistema *CentOS 6.1*.

Em suma, uma firewall nada mais é que um dispositivo pertencente a uma rede de computadores que tem como principal objetivo aplicar uma política de segurança num determinado ponto da rede. Em geral, estão associadas a redes TCP/IP.

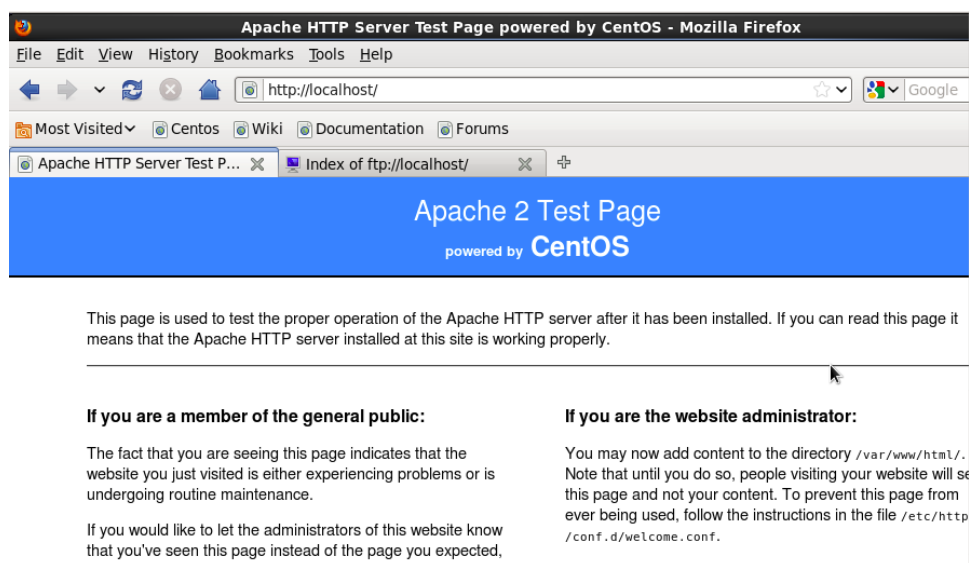
## TAREFA 1

### 1.1 ALÍNEA 1

Depois de ativar e ligar os serviços ssh ftp e http, executamos o comando netstat -l, que nos mostra todas as ligações que se encontram à escuta:

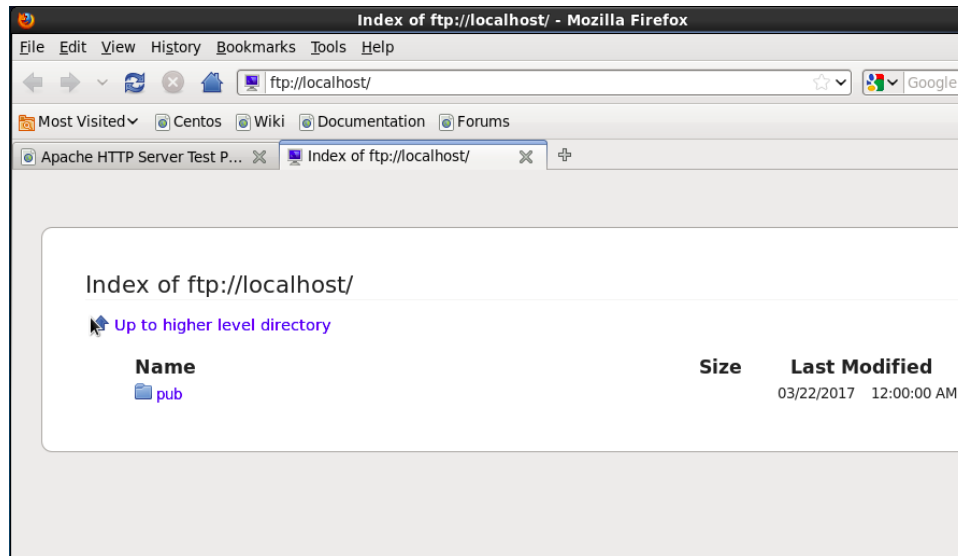
```
[root@localhost ~]# netstat -l
Active Internet connections (only servers)
Proto Recv-Q Send-Q Local Address           Foreign Address         Stat
e
tcp        0      0 *:59372                *:.*                     LIST
EN
tcp        0      0 *:sunrpc                *:.*                     LIST
EN
tcp        0      0 *:ftp                   *:.*                     LIST
EN
tcp        0      0 *:ssh                   *:.*                     LIST
EN
tcp        0      0 localhost.localdomain:ipp *:.*                     LIST
EN
tcp        0      0 localhost.localdomain:smtp *:.*                     LIST
EN
tcp        0      0 *:sunrpc                *:.*                     LIST
EN
tcp        0      0 *:http                  *:.*                     LIST
EN
tcp        0      0 *:ssh                   *:.*                     LIST
EN
tcp        0      0 localhost6.localdomain6:ipp *:.*                     LIST
```

Como podemos observar tanto o ftp, ssh e http estão a escuta. Para verificar o correto funcionamento dos serviços, foi aberto o serviço http no browser Mozilla Firefox:



E como podemos verificar pela página teste que aparece, o serviço está a correr corretamente.

Para verificar o bom funcionamento do serviço ftp, abrimo-lo também no browser, sendo que a página que abre demonstra que o serviço está a funcionar corretamente.



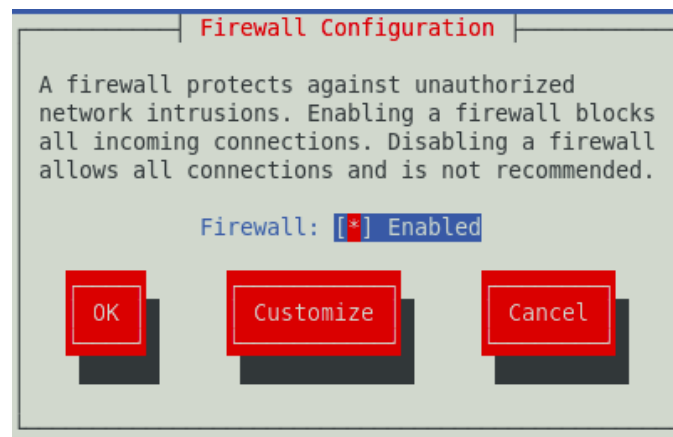
Para o ssh, o serviço foi aberto pelo terminal.

```
[root@localhost ~]# ssh localhost
The authenticity of host 'localhost (127.0.0.1)' can't be established.
RSA key fingerprint is 4b:84:0c:ab:9c:9f:63:8f:2f:c1:36:27:e2:16:19:aa.
Are you sure you want to continue connecting (yes/no)? y
Please type 'yes' or 'no': yes
Warning: Permanently added 'localhost' (RSA) to the list of known hosts.
root@localhost's password:
[root@localhost ~]# ls
anaconda-ks.cfg  Documents  Music      post-install  Public  Videos
Desktop          Downloads  Pictures   post-install.log  Templates
```

Como podemos observar na figura, o serviço encontra-se a funcionar corretamente.

## 1.2 ALÍNEA 2

Executamos o comando `system-config-firewall-tui`, mas não foi preciso ativar a firewall, visto que esta já se encontrava ativa.



## 1.3 ALÍNEA 3

Para verificar as regras estabelecidas pela firewall, introduzimos no terminal o comando `iptables -L -v`. Em baixo podemos verificar o output:

```
[root@localhost ~]# iptables -L -v
Chain INPUT (policy ACCEPT 0 packets, 0 bytes)
pkts bytes target      prot opt in      out     source       destination
  0      0 ACCEPT     all  --  any    any     anywhere     anywhere
    state RELATED,ESTABLISHED
  0      0 ACCEPT     icmp --  any    any     anywhere     anywhere
  0      0 ACCEPT     all  --  lo     any     anywhere     anywhere
  0      0 ACCEPT     tcp  --  any    any     anywhere     anywhere
    state NEW tcp dpt:ssh
  0      0 REJECT     all  --  any    any     anywhere     anywhere
    reject-with icmp-host-prohibited

Chain FORWARD (policy ACCEPT 0 packets, 0 bytes)
pkts bytes target      prot opt in      out     source       destination
  0      0 REJECT     all  --  any    any     anywhere     anywhere
    reject-with icmp-host-prohibited

Chain OUTPUT (policy ACCEPT 0 packets, 0 bytes)
pkts bytes target      prot opt in      out     source       destination
```

### 1.3.1 Alínea 3 – A

#### INPUT

Podemos observar na figura que para esta cadeia as políticas estabelecidas permitem:

- Que qualquer pacote que tenha o estado RELATED (pacote inicializa uma nova conexão, mas está associado a uma nova conexão) e ESTABLISHED (o pacote está associado a uma conexão) sejam aceites;
- Que qualquer pacote, desde que o protocolo utilizado seja ICMP, seja aceite;

- Que pacotes que entrem pela interface lo (localhost) sejam aceites;
- Que pacotes com o protocolo tcp, que vêm através de ssh, sejam aceites;
- Qualquer outro pacote que não se encaixe nas regras acima são rejeitados com “icmp-host-prohibited”.

## FORWARD

Aqui a política estabelecida pela firewall é que todos os pacotes recebidos por qualquer interface e de qualquer destino são rejeitados. Isto acontece porque a firewall não está configurada para routing.

## OUTPUT

Não existem regras estabelecidas pela firewall nesta cadeia, logo não existe restrições, sendo que qualquer pacote passa sem qualquer tipo de permissão.

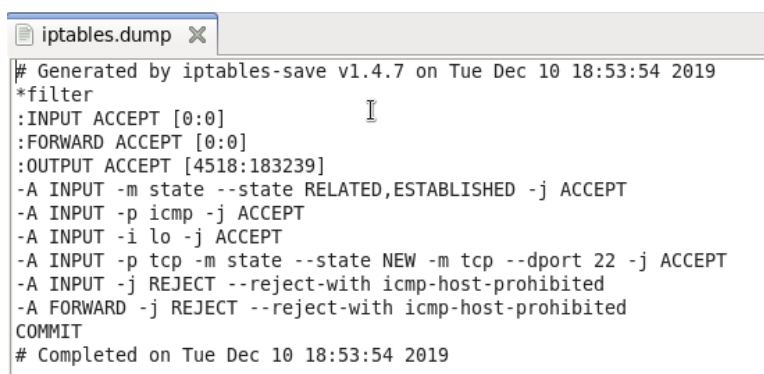
### 1.3.2 Alínea 3 – B

Ao observar os resultados da figura, podemos concluir que as regras estabelecidas não proporcionam um nível de segurança elevado, nem são de complexidade elevada. Existem várias falhas que podem ser observadas com base nas permissões oferecidas.

Um exemplo é a permissão de todos os pacotes com protocolo ICMP, que pode permitir ataques DoS, como um Ping Flood, também conhecido como ICMP Flood. Outro exemplo é a permissão de todos os pacotes que entram na rede localhost. Qualquer pessoa que tenha acesso a mesma rede do servidor poderá aceder ao servidor e explorar as suas vulnerabilidades.

## 1.4 ALÍNEA 4

Como recomendado, executamos o comando “iptables-save > iptables.dump” para gravar as iptables estabelecidas.

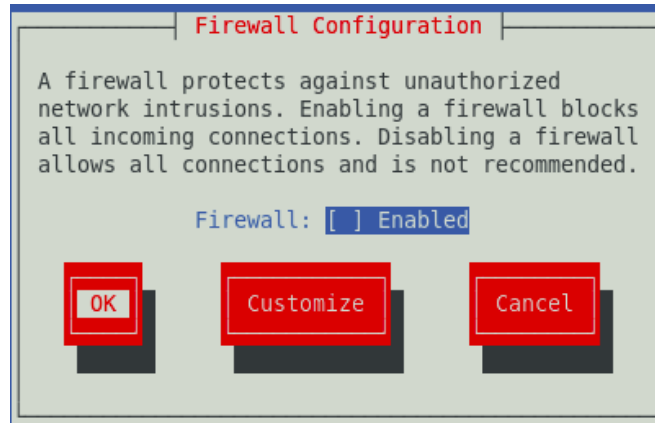


```

# Generated by iptables-save v1.4.7 on Tue Dec 10 18:53:54 2019
*filter
:INPUT ACCEPT [0:0]
:FORWARD ACCEPT [0:0]
:OUTPUT ACCEPT [4518:183239]
-A INPUT -m state --state RELATED,ESTABLISHED -j ACCEPT
-A INPUT -p icmp -j ACCEPT
-A INPUT -i lo -j ACCEPT
-A INPUT -p tcp -m state --state NEW -m tcp --dport 22 -j ACCEPT
-A INPUT -j REJECT --reject-with icmp-host-prohibited
-A FORWARD -j REJECT --reject-with icmp-host-prohibited
COMMIT
# Completed on Tue Dec 10 18:53:54 2019
  
```

## 1.5 ALÍNEA 5

Como pedido, desligamos a firewall:



E voltamos a reintroduzir o comando iptables -L -v:

```
[root@localhost ~]# iptables -L -v
Chain INPUT (policy ACCEPT 0 packets, 0 bytes)
 pkts bytes target    prot opt in     out     source         destination
Chain FORWARD (policy ACCEPT 0 packets, 0 bytes)
 pkts bytes target    prot opt in     out     source         destination
Chain OUTPUT (policy ACCEPT 0 packets, 0 bytes)
 pkts bytes target    prot opt in     out     source         destination
[root@localhost ~]#
```

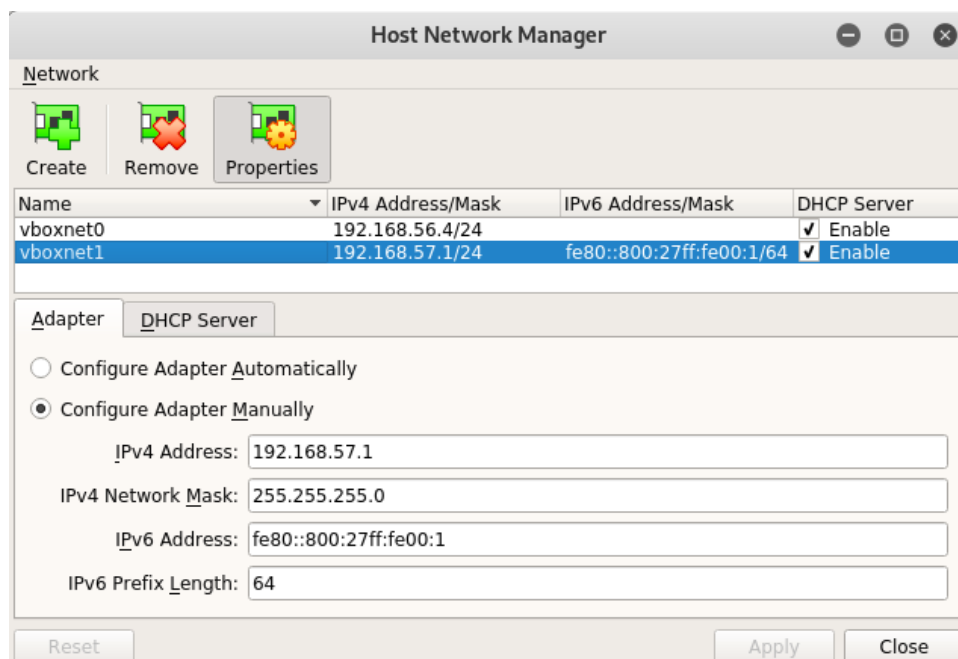
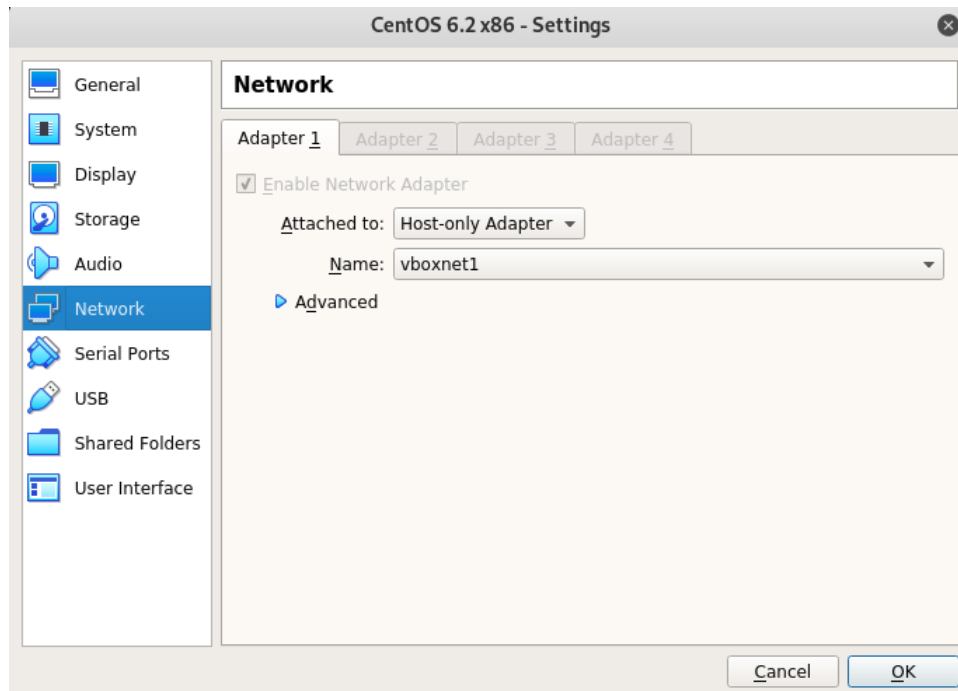
Com o firewall desligado, podemos ver que não existe nenhuma regra estabelecida. Com isto o sistema fica vulnerável a qualquer ataque, visto que o sistema agora aceita qualquer tipo de pacote. Ao fim desta análise a firewall foi ligado novamente, e como podemos ver em baixo, as regras que se encontravam anteriormente foram repostas:

```
[root@localhost ~]# iptables -L -v
Chain INPUT (policy ACCEPT 0 packets, 0 bytes)
 pkts bytes target    prot opt in     out     source         destination
    0    0 ACCEPT    all  --  any    any    anywhere       anywhere
    state RELATED,ESTABLISHED
    0    0 ACCEPT    icmp --  any    any    anywhere       anywhere
    0    0 ACCEPT    all  --  lo     any    anywhere       anywhere
    0    0 ACCEPT    tcp  --  any    any    anywhere       anywhere
    state NEW tcp dpt:ssh
    0    0 REJECT    all  --  any    any    anywhere       anywhere
    reject-with icmp-host-prohibited
Chain FORWARD (policy ACCEPT 0 packets, 0 bytes)
 pkts bytes target    prot opt in     out     source         destination
    0    0 REJECT    all  --  any    any    anywhere       anywhere
    reject-with icmp-host-prohibited
Chain OUTPUT (policy ACCEPT 0 packets, 0 bytes)
 pkts bytes target    prot opt in     out     source         destination
```



## 2 TAREFA 2

Para realizar a tarefa 2 foi utilizado o kali Linux como cliente. Como o kali linux já se encontrava instalado nativamente no computador, foi realizado uma conexão host-only Adapter entre o kali Linux e o VMCentOS.



Onde o Kali linux tem o endereço 192.168.57.1:

```
vboxnet1: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    inet 192.168.57.1 netmask 255.255.255.0 broadcast 192.168.57.255
    inet6 fe80::800:27ff:fe00:1 prefixlen 64 scopeid 0x20<link>
    ether 0a:00:27:00:00:01 txqueuelen 1000 (Ethernet)
    RX packets 0 bytes 0 (0.0 B)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 36 bytes 6170 (6.0 KiB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0
```

E o CentOS tem o endereço 192.168.57.3:

```
eth0: Link encap:Ethernet HWaddr 08:00:27:68:90:87
    inet addr:192.168.57.3 Bcast:192.168.57.255 Mask:255.255.255.0
    inet6 addr: fe80::a00:27ff:fe68:9087/64 Scope:Link
    UP BROADCAST RUNNING MULTICAST MTU:1500 Metric:1
    RX packets:2 errors:0 dropped:0 overruns:0 frame:0
    TX packets:20 errors:0 dropped:0 overruns:0 carrier:0
    collisions:0 txqueuelen:1000
    RX bytes:1180 (1.1 KiB) TX bytes:3926 (3.8 KiB)
```

## 2.1 ALÍNEA 1

Realizamos o ping para 192.168.57.3, para verificar a conectividade entre o Kali Linux e o CentOS:

```
root@kali:~# ping 192.168.57.3
PING 192.168.57.3 (192.168.57.3) 56(84) bytes of data:
64 bytes from 192.168.57.3: icmp_seq=1 ttl=64 time=0.278 ms
64 bytes from 192.168.57.3: icmp_seq=2 ttl=64 time=0.291 ms
64 bytes from 192.168.57.3: icmp_seq=3 ttl=64 time=0.192 ms
64 bytes from 192.168.57.3: icmp_seq=4 ttl=64 time=0.265 ms
64 bytes from 192.168.57.3: icmp_seq=5 ttl=64 time=0.271 ms
^C
--- 192.168.57.3 ping statistics ---
5 packets transmitted, 5 received, 0% packet loss, time 4086ms
rtt min/avg/max/mdev = 0.192/0.259/0.291/0.034 ms
```

Como podemos ver, foram obtidos pacotes vindos do servidor, concluindo que existe conectividade entre os dois sistemas.

## 2.2 ALÍNEA 2

### Nmap -sS

O comando nmap -sS é utilizado para encontrar portas TCP abertas. Este scan envia pacotes SYN como se fosse abrir uma conexão real com a porta e depois espera pela resposta.

Como podemos verificar na figura abaixo, existe uma porta aberta, que está associado ao serviço ssh.

```
root@kali:~# nmap -sS 192.168.57.3
Starting Nmap 7.80 ( https://nmap.org ) at 2019-12-10 18:22 WET
Nmap scan report for 192.168.57.3
Host is up (0.00041s latency).
Not shown: 999 filtered ports
PORT      STATE SERVICE
22/tcp    open  ssh
MAC Address: 08:00:27:68:90:87 (Oracle VirtualBox virtual NIC)
Nmap done: 1 IP address (1 host up) scanned in 5.29 seconds
```

### 2.3 ALÍNEA 3

Ao executar o comando podemos verificar que não é possível utilizar o serviço, visto que na firewall está definido para não aceitar conexões http, logo não conseguimos ver nenhuma página.

```
root@kali:~# w3m http://192.168.57.3
w3m: Can't load http://192.168.57.3.
```

### 2.4 ALÍNEA 4

Ao executar o comando ftp, podemos verificar que não obtemos resposta do servidor, concluindo que a firewall está efetivamente a bloquear pedidos de conexões ftp.

```
root@kali:~# ftp 192.168.57.3
ftp: connect: No route to host
```

### 2.5 ALÍNEA 5

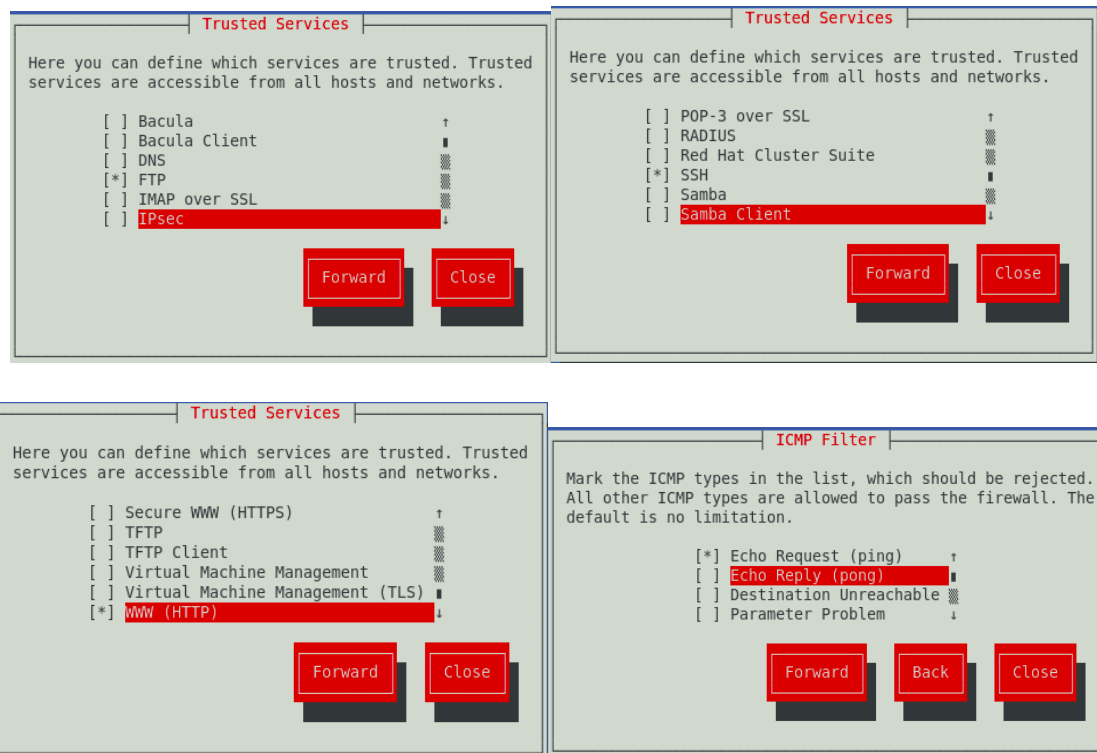
Ao realizar a conexão através de ssh, esta foi conseguida com sucesso. Isto acontece pelo facto que a firewall permite qualquer conexão ssh através do protocolo tcp

```
root@kali:~# ssh 192.168.57.3
The authenticity of host '192.168.57.3 (192.168.57.3)' can't be established.
RSA key fingerprint is SHA256:1yAMrleDSyQDiSnVspB/iM9xh4n7+bhvpJ04J7Vbj0.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '192.168.57.3' (RSA) to the list of known hosts.
root@192.168.57.3's password:
[root@localhost ~]# ls
anaconda-ks.cfg  Downloads  Pictures  Public
Desktop          iptables.dump  post-install  Templates
Documents        Music        post-install.log  Videos
```

### 3 TAREFA 3

#### 3.1 ALÍNEA 1

De modo a permitir ligações através de http, ssh e ftp, definimos estes serviços como de confiança e possíveis de aceder através de qualquer rede e host, e marcamos também, na opção de filtragens do protocolo icmp a opção Echo Request, que vai impedir que sejam realizados pings para o servidor.



### 3.2 ALÍNEA 2

Depois de configurar a firewall para permitir acesso através de ssh, ftp e http, introduzimos novamente o comando iptables -L -v:

```
[root@localhost ~]# iptables -L -v
Chain INPUT (policy ACCEPT 0 packets, 0 bytes)
  pkts bytes target     prot opt in     out     source         destination
    0      0 ACCEPT     all  --  any    any     anywhere       anywhere
    0      0 state RELATED,ESTABLISHED
    0      0 REJECT     icmp  --  any    any     anywhere       anywhere
    0      0 icmp echo-request reject-with icmp-host-prohibited
    0      0 ACCEPT     icmp  --  any    any     anywhere       anywhere
    0      0 ACCEPT     all  --  lo     any     anywhere       anywhere
    0      0 ACCEPT     tcp  --  any    any     anywhere       anywhere
    0      0 state NEW tcp dpt:ssh
    0      0 ACCEPT     tcp  --  any    any     anywhere       anywhere
    0      0 state NEW tcp dpt:http
    0      0 ACCEPT     tcp  --  any    any     anywhere       anywhere
    0      0 state NEW tcp dpt:ftp
    0      0 REJECT     all  --  any    any     anywhere       anywhere
    0      0 reject-with icmp-host-prohibited

Chain FORWARD (policy ACCEPT 0 packets, 0 bytes)
  pkts bytes target     prot opt in     out     source         destination
    0      0 REJECT     all  --  any    any     anywhere       anywhere
    0      0 reject-with icmp-host-prohibited

Chain OUTPUT (policy ACCEPT 0 packets, 0 bytes)
  pkts bytes target     prot opt in     out     source         destination
```

Como podemos verificar na figura acima, as regras definidas anteriormente foram alteradas, e agora está definido a permissão de acesso através dos protocolos ssh, http e ftp, como também a rejeição de qualquer pacote com o protocolo ICMP.

Como resultado, as novas regras são as seguintes:

#### INPUT

Podemos observar na figura que para esta cadeia as políticas estabelecidas permitem:

- Que qualquer pacote que tenha o estado RELATED (pacote inicializa uma nova conexão, mas está associado a uma nova conexão) e ESTABLISHED (o pacote está associado a uma conexão) sejam aceites;
- Que pacotes com o protocolo ICMP são rejeitados;
- Que pacotes que entrem pela interface lo (localhost) sejam aceites;
- Que pacotes com o protocolo tcp, que vêm através de ssh, http ou ftp, sejam aceites;
- Qualquer outro pacote que não se encaixe nas regras acima são rejeitados com "icmp-host-prohibited".

## FORWARD

Não houve alterações, então todos os pacotes recebidos por qualquer interface e de qualquer destino são rejeitados.

## OUTPUT

Não existem regras estabelecidas pela firewall nesta cadeia, logo não existe restrições, sendo que qualquer pacote passa sem qualquer tipo de permissão.

### 3.3 ALÍNEA 3

Depois de realizar o ping para o servidor, e como consequência das novas regras da firewall, podemos ver que realizar o ping resulta uma resposta “Destination Host Prohibited”. Isto deve-se ao facto dos ping requests terem sido impedidos de serem feitos no passo anterior.

```
root@kali:~# ping 192.168.57.3
PING 192.168.57.3 (192.168.57.3) 56(84) bytes of data:
From 192.168.57.3 icmp_seq=1 Destination Host Prohibited
From 192.168.57.3 icmp_seq=2 Destination Host Prohibited
From 192.168.57.3 icmp_seq=3 Destination Host Prohibited
From 192.168.57.3 icmp_seq=4 Destination Host Prohibited
From 192.168.57.3 icmp_seq=5 Destination Host Prohibited
^C
--- 192.168.57.3 ping statistics ---
5 packets transmitted, 0 received, +5 errors, 100% packet loss, time 4077ms
```

### 3.4 ALÍNEA 4

Como anteriormente, voltamos a reintroduzir o comando **w3m http://192.168.57.3**, e como nas regras do firewall ficou definindo que era permitido ligações http, conseguimos agora visualizar a pagina web, que é a pagina de teste para ligações http, logo podemos concluir que a conexão foi realizada com sucesso.

Apache 2 Test Page  
powered by CentOS

This page is used to test the proper operation of the Apache HTTP server after it has been installed. If you can read this page it means that the Apache HTTP server installed at this site is working properly.

If you are a member of the general public:

The fact that you are seeing this page indicates that the website you just visited is either experiencing problems or is undergoing routine maintenance.

If you would like to let the administrators of this website know that you've seen this page instead of the page you expected, you should send them e-mail. In general, mail sent to the name "webmaster" and directed to the website's domain should reach the appropriate person.

For example, if you experienced problems while visiting [www.example.com](http://www.example.com), you should send e-mail to "webmaster@example.com".

If you are the website administrator:

You may now add content to the directory `/var/www/html/`. Note that until you do so, people visiting your website will see this page and not your content. To prevent this page from ever being used, follow the instructions in the file `/etc/httpd/conf.d/welcome.conf`.

You are free to use the images below on Apache and CentOS Linux powered HTTP servers. Thanks for using Apache and CentOS!

[ Powered by Apache ] [ Powered by CentOS Linux ] No servidor:

About CentOS:

The Community Enterprise Operating System (CentOS) is an Enterprise-class Linux Distribution derived from sources freely provided to the public by a prominent North American Enterprise Linux vendor. CentOS conforms fully with the upstream vendors redistribution policy and aims to be 100% binary compatible. (CentOS mainly changes packages to remove upstream vendor branding and artwork.) The CentOS Project is the organization that builds CentOS.

For information on CentOS please visit the [CentOS website](http://www.centos.org).

### 3.5 ALÍNEA 5

Voltamos a introduzir o comando “[ftp 192.168.57.3](#)”, e como foi notado no guião, quando tentamos conectar através de utilizadores existentes, esta conexão não foi bem-sucedida.

```
root@kali:~# ftp 192.168.57.3
Connected to 192.168.57.3.
220 (vsFTPD 2.2.2)
Name (192.168.57.3:root): root
530 Permission denied.
Login failed.
```

```
root@kali:~# ftp 192.168.57.3
Connected to 192.168.57.3.
220 (vsFTPD 2.2.2)
Name (192.168.57.3:root): centos
331 Please specify the password.
Password:
500 OOPS: cannot change directory:/home/centos
Login failed.
```

Quando a ligação é feita através do usuário *anonymous*, a ligação é bem-sucedida. Isto acontece porque foi definido nas regras da firewall a permissão de conexões através do serviço ftp.

```
root@kali:~# ftp 192.168.57.3
Connected to 192.168.57.3.
220 (vsFTPD 2.2.2)
Name (192.168.57.3:root): anonymous
331 Please specify the password.
Password:
230 Login successful.
Remote system type is UNIX.
Using binary mode to transfer files.
```

### 3.6 ALÍNEA 6

Uma vez efetuado novamente o scan `-sS`, podemos verificar que agora existem mais portas abertas, correspondendo aos serviços que foram identificados como de confiança nas regras de firewall, e permitidas conexões. Estas portas correspondem ao ftp, ssh e http.

```
root@kali:~# nmap -sS 192.168.57.3
Starting Nmap 7.80 ( https://nmap.org ) at 2019-12-10 18:49 WET
Nmap scan report for 192.168.57.3
Host is up (0.00040s latency).
Not shown: 997 filtered ports
PORT      STATE SERVICE
21/tcp    open  ftp
22/tcp    open  ssh
80/tcp    open  http
MAC Address: 08:00:27:68:90:87 (Oracle VirtualBox virtual NIC)
Conclusão:
Nmap done: 1 IP address (1 host up) scanned in 5.07 seconds
```

### 3.7 ALÍNEA 7

Introduzindo novamente o comando “**iptables -L -v**”, podemos agora verificar que houve alterações do número de pacotes trocados pelo servidor.

```
[root@localhost ~]# iptables -L -v
Chain INPUT (policy ACCEPT 0 packets, 0 bytes)
pkts bytes target      prot opt in      out     source         destination
 51  4077 ACCEPT      all  --  any     any     anywhere       anywhere
    state RELATED,ESTABLISHED
  5   420 REJECT      icmp --  any     any     anywhere       anywhere
    icmp echo-request reject-with icmp-host-prohibited
  0     0 ACCEPT      icmp --  any     any     anywhere       anywhere
  0     0 ACCEPT      all  --  lo      any     anywhere       anywhere
  1    44 ACCEPT      tcp  --  any     any     anywhere       anywhere
    state NEW tcp dpt:ssh
  2   104 ACCEPT      tcp  --  any     any     anywhere       anywhere
    state NEW tcp dpt:http
  5   284 ACCEPT      tcp  --  any     any     anywhere       anywhere
    state NEW tcp dpt:ftp
1985 87340 REJECT      all  --  any     any     anywhere       anywhere
    reject-with icmp-host-prohibited

Chain FORWARD (policy ACCEPT 0 packets, 0 bytes)
pkts bytes target      prot opt in      out     source         destination
  0     0 REJECT      all  --  any     any     anywhere       anywhere
    reject-with icmp-host-prohibited

Chain OUTPUT (policy ACCEPT 67 packets, 10243 bytes)
pkts bytes target      prot opt in      out     source         destination
```

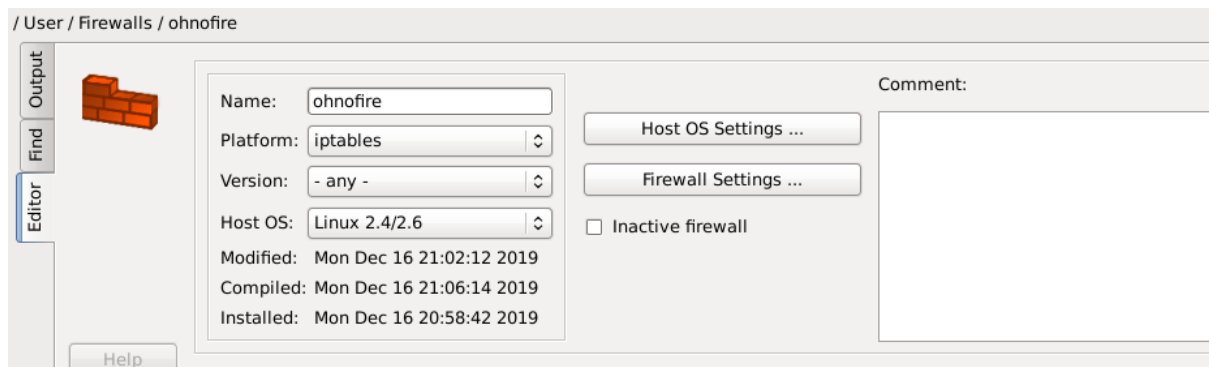
Como resultado, podemos verificar que agora existe trocas de pacotes através dos serviços http, ftp e ssh.



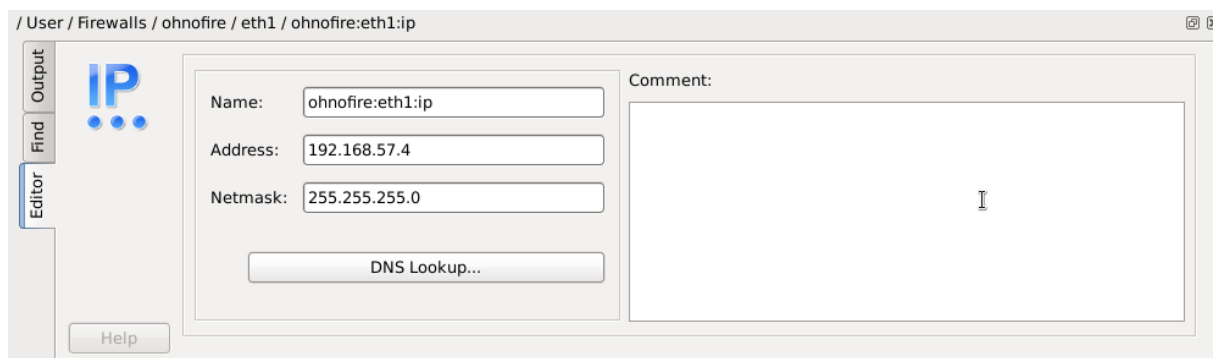
## 4 TAREFA 3 – MELHORADA - CONCLUSÃO

Para consolidar conhecimentos sobre iptables e conhecer ferramentas que nos permitem a sua manipulação, instalamos uma interface gráfica para realizar os melhoramentos à proteção implementada anteriormente. A interface escolhida foi o fwbuilder.

Primeiramente criamos uma firewall nova, a que lhe chamamos “**ohnofire**”.

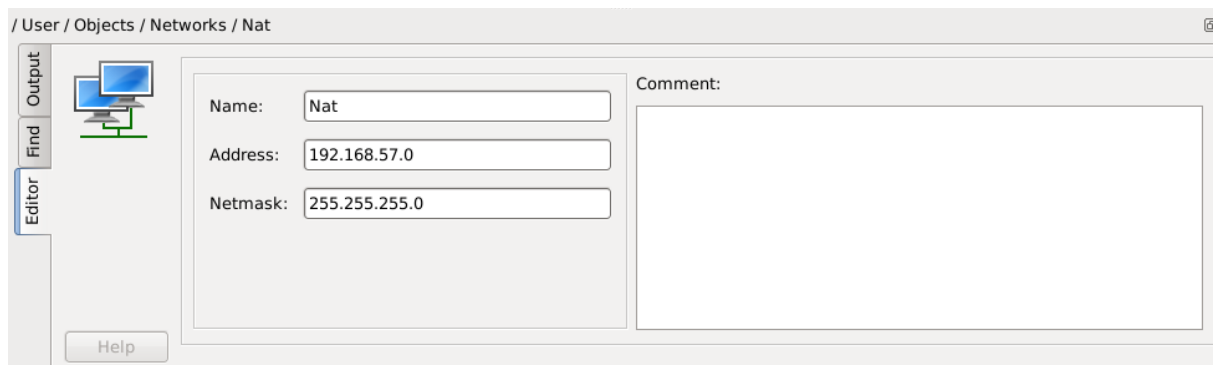


Criamos a interface “eth1”, e definimos o seu IP:



Configuramos também uma rede, que vai ser aquela onde será possível comunicar com a firewall. Vamos utilizar a rede local como único canal de comunicação com a firewall.

A esta rede chamamos “Nat”.



Passamos de seguida à definição das regras:

	Source	Destination	Service	Interface	Direction	Action	Time	Options
0	ohnofire	Any	ICMP ping reply	All	Both	Reject:ICMP host ...	Any	
1	Any	ohnofire	ICMP ping request	All	Both	Reject:ICMP host ...	Any	
2	Nat	ohnofire	TCP http TCP https	All	Both	Accept	Any	
3	Nat	ohnofire	TCP ftp	All	Both	Accept	Any	
4	Nat	ohnofire	TCP ssh	All	Both	Accept	Any	
5	Any	ohnofire	Any	All	Both	Deny	Any	

Como pedido, restringimos o acesso aos serviços ssh ftp e http (e https só porque sim) para serem acedidas apenas por utilizadores que se encontram ligados a rede Nat (rede local).

Também rejeitamos qualquer pedido de ping feitos por qualquer rede ou interface com um **“Reject: ICMP host prohibited”**. Por fim negamos acesso a qualquer outro serviço que venha de qualquer source ou interface. Todos estas regras têm a **função log ligada**.

Depois de acabar de adicionar regras, compilamos e instalamos a firewall no sistema.



A partir daqui, ao introduzir novamente o comando “**iptables -v -L**”, podemos verificar que as novas regras se encontram ativas:

```
Chain FORWARD (policy ACCEPT 0 packets, 0 bytes)
pkts bytes target      prot opt in      out     source           destination

Chain OUTPUT (policy ACCEPT 0 packets, 0 bytes)
pkts bytes target      prot opt in      out     source           destination

[root@localhost ~]# iptables -v -L
Chain INPUT (policy DROP 0 packets, 0 bytes)
pkts bytes target      prot opt in      out     source           destination

 151 13548 ACCEPT      all  --  any     any     anywhere         anywhere
    state RELATED,ESTABLISHED
    0    0 RULE_0      icmp --  any     any     192.168.57.4     anywhere
    icmp type 0 code 0
  12 1008 RULE_1      icmp --  any     any     anywhere         anywhere
    icmp type 8 code 0
    1   60 RULE_2      tcp  --  any     any     192.168.57.0/24  anywhere
    tcp multiport dports http,https state NEW
    2  120 RULE_3      tcp  --  any     any     192.168.57.0/24  anywhere
    tcp dpt:ftp state NEW
    2  120 RULE_4      tcp  --  any     any     192.168.57.0/24  anywhere
    tcp dpt:ssh state NEW
    2   80 RULE_5      all  --  any     any     anywhere         anywhere

Chain FORWARD (policy DROP 0 packets, 0 bytes)
pkts bytes target      prot opt in      out     source           destination

    0    0 ACCEPT      all  --  any     any     anywhere         anywhere
    state RELATED,ESTABLISHED

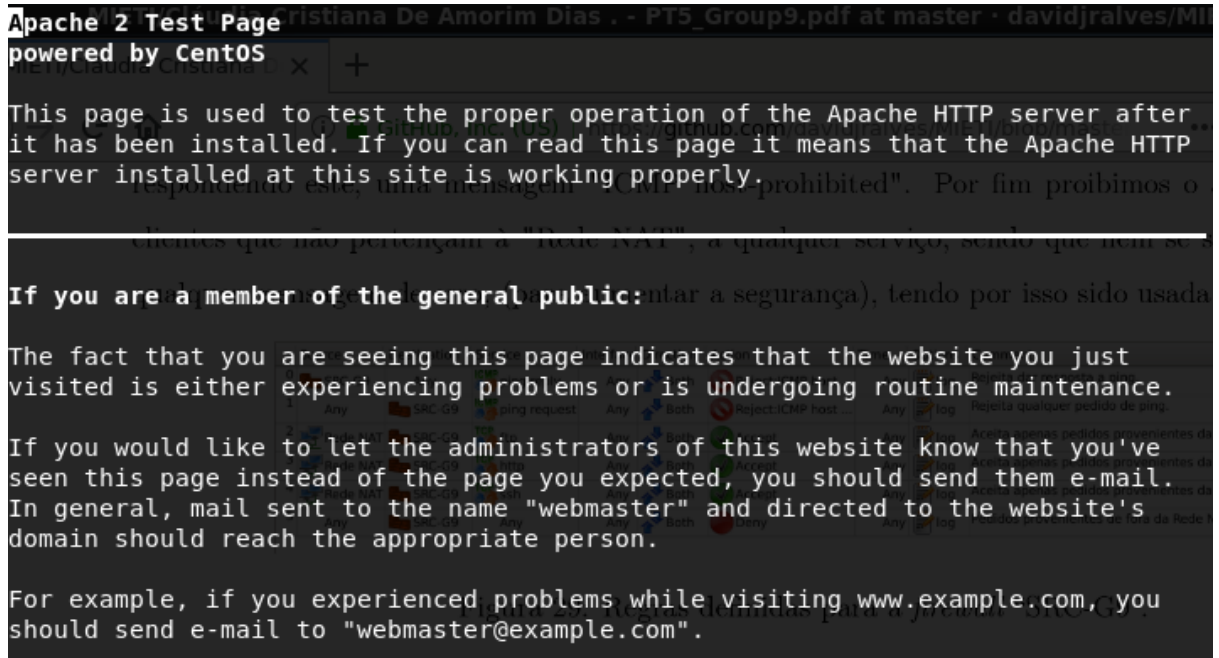
Chain OUTPUT (policy DROP 202 packets, 12820 bytes)
pkts bytes target      prot opt in      out     source           destination

 142 20525 ACCEPT      all  --  any     any     anywhere         anywhere
    state RELATED,ESTABLISHED
    0    0 RULE_0      icmp --  any     any     anywhere         anywhere
    icmp type 0 code 0
    0    0 RULE_1      icmp --  any     any     anywhere         192.168.57.4
    icmp type 8 code 0
    5  300 RULE_5      all  --  any     any     anywhere         192.168.57.4
```

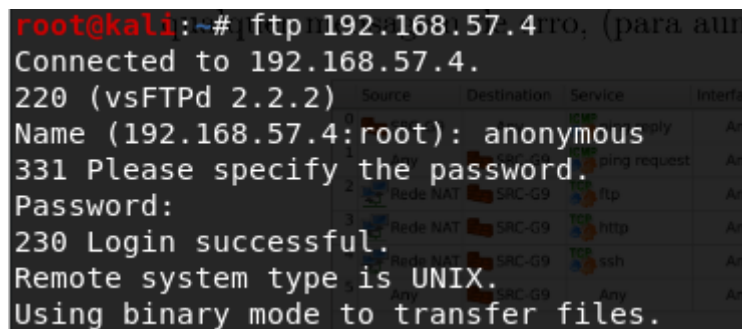
Chain OUTPUT (policy DROP 202 packets, 12820 bytes)								
pkts	bytes	target	prot	opt	in	out	source	destination
142	20525	ACCEPT	all	--	any	any	anywhere	anywhere
		state RELATED,ESTABLISHED						
0	0	RULE_0	icmp	--	any	any	anywhere	anywhere
		icmp type 0 code 0						
0	0	RULE_1	icmp	--	any	any	anywhere	192.168.57.4
		icmp type 8 code 0						
5	300	RULE_5	all	--	any	any	anywhere	192.168.57.4
Chain RULE_0 (2 references)								
pkts	bytes	target	prot	opt	in	out	source	destination
0	0	LOG	all	--	any	any	anywhere	anywhere
		LOG level info prefix `RULE 0 -- REJECT`						
0	0	REJECT	all	--	any	any	anywhere	anywhere
		reject-with icmp-host-prohibited						
Chain RULE_1 (2 references)								
pkts	bytes	target	prot	opt	in	out	source	destination
12	1008	LOG	all	--	any	any	anywhere	anywhere
		LOG level info prefix `RULE 1 -- REJECT`						
12	1008	REJECT	all	--	any	any	anywhere	anywhere
		reject-with icmp-host-prohibited						
Chain RULE_2 (1 references)								
pkts	bytes	target	prot	opt	in	out	source	destination
1	60	LOG	all	--	any	any	anywhere	anywhere
		LOG level info prefix `RULE 2 -- ACCEPT`						
1	60	ACCEPT	all	--	any	any	anywhere	anywhere
Chain RULE_3 (1 references)								
pkts	bytes	target	prot	opt	in	out	source	destination
2	120	LOG	all	--	any	any	anywhere	anywhere
		LOG level info prefix `RULE 3 -- ACCEPT`						

Podemos também verificar que ao realizar novamente os comandos “w3m <http://192.168.57.4>”, “ssh 192.168.57.4” e “ftp 192.168.57.4”, ainda temos acesso aos serviços.

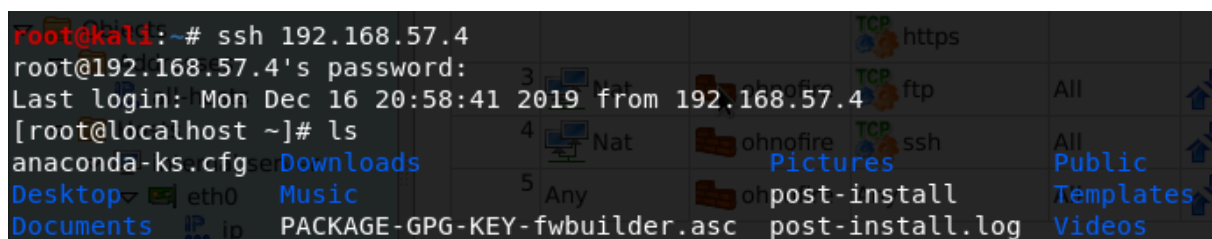
w3m <http://192.168.57.4>:



ftp 192.168.57.4:



ssh 192.168.57.4:



E que pings direcionados ao servidor são rejeitados:

```
root@kali:~# ping 192.168.57.4
PING 192.168.57.4 (192.168.57.4) 56(84) bytes of data.
From 192.168.57.4 icmp_seq=1 Destination Host Prohibited
From 192.168.57.4 icmp_seq=2 Destination Host Prohibited
From 192.168.57.4 icmp_seq=3 Destination Host Prohibited
From 192.168.57.4 icmp_seq=4 Destination Host Prohibited
From 192.168.57.4 icmp_seq=5 Destination Host Prohibited
From 192.168.57.4 icmp_seq=6 Destination Host Prohibited
From 192.168.57.4 icmp_seq=7 Destination Host Prohibited
From 192.168.57.4 icmp_seq=8 Destination Host Prohibited
From 192.168.57.4 icmp_seq=9 Destination Host Prohibited
From 192.168.57.4 icmp_seq=10 Destination Host Prohibited
```

Sendo que esta informação é depois registada no log:

```
Dec 16 20:58:58 localhost kernel: RULE 1 -- REJECT IN=eth1 OUT= MAC=08:00:27:14:fd:5f:0a:00:27:00:00:01:08:00
SRC=192.168.57.1 DST=192.168.57.4 LEN=84 TOS=0x00 PREC=0x00 TTL=64 ID=52231 DF PROTO=ICMP TYPE=8 CODE=0 ID=12294 SEQ=1
Dec 16 20:58:59 localhost kernel: RULE 1 -- REJECT IN=eth1 OUT= MAC=08:00:27:14:fd:5f:0a:00:27:00:00:01:08:00
SRC=192.168.57.1 DST=192.168.57.4 LEN=84 TOS=0x00 PREC=0x00 TTL=64 ID=52446 DF PROTO=ICMP TYPE=8 CODE=0 ID=12294 SEQ=2
Dec 16 20:59:00 localhost kernel: RULE 1 -- REJECT IN=eth1 OUT= MAC=08:00:27:14:fd:5f:0a:00:27:00:00:01:08:00
SRC=192.168.57.1 DST=192.168.57.4 LEN=84 TOS=0x00 PREC=0x00 TTL=64 ID=52529 DF PROTO=ICMP TYPE=8 CODE=0 ID=12294 SEQ=3
Dec 16 20:59:01 localhost kernel: RULE 1 -- REJECT IN=eth1 OUT= MAC=08:00:27:14:fd:5f:0a:00:27:00:00:01:08:00
SRC=192.168.57.1 DST=192.168.57.4 LEN=84 TOS=0x00 PREC=0x00 TTL=64 ID=52730 DF PROTO=ICMP TYPE=8 CODE=0 ID=12294 SEQ=4
Dec 16 20:59:02 localhost kernel: RULE 1 -- REJECT IN=eth1 OUT= MAC=08:00:27:14:fd:5f:0a:00:27:00:00:01:08:00
SRC=192.168.57.1 DST=192.168.57.4 LEN=84 TOS=0x00 PREC=0x00 TTL=64 ID=52944 DF PROTO=ICMP TYPE=8 CODE=0 ID=12294 SEQ=5
Dec 16 20:59:03 localhost kernel: RULE 1 -- REJECT IN=eth1 OUT= MAC=08:00:27:14:fd:5f:0a:00:27:00:00:01:08:00
SRC=192.168.57.1 DST=192.168.57.4 LEN=84 TOS=0x00 PREC=0x00 TTL=64 ID=53137 DF PROTO=ICMP TYPE=8 CODE=0 ID=12294 SEQ=6
Dec 16 20:59:04 localhost kernel: RULE 1 -- REJECT IN=eth1 OUT= MAC=08:00:27:14:fd:5f:0a:00:27:00:00:01:08:00
SRC=192.168.57.1 DST=192.168.57.4 LEN=84 TOS=0x00 PREC=0x00 TTL=64 ID=53200 DF PROTO=ICMP TYPE=8 CODE=0 ID=12294 SEQ=7
Dec 16 20:59:05 localhost kernel: RULE 1 -- REJECT IN=eth1 OUT= MAC=08:00:27:14:fd:5f:0a:00:27:00:00:01:08:00
SRC=192.168.57.1 DST=192.168.57.4 LEN=84 TOS=0x00 PREC=0x00 TTL=64 ID=53325 DF PROTO=ICMP TYPE=8 CODE=0 ID=12294 SEQ=8
Dec 16 20:59:06 localhost kernel: RULE 1 -- REJECT IN=eth1 OUT= MAC=08:00:27:14:fd:5f:0a:00:27:00:00:01:08:00
SRC=192.168.57.1 DST=192.168.57.4 LEN=84 TOS=0x00 PREC=0x00 TTL=64 ID=53543 DF PROTO=ICMP TYPE=8 CODE=0 ID=12294 SEQ=9
Dec 16 20:59:07 localhost kernel: RULE 1 -- REJECT IN=eth1 OUT= MAC=08:00:27:14:fd:5f:0a:00:27:00:00:01:08:00
SRC=192.168.57.1 DST=192.168.57.4 LEN=84 TOS=0x00 PREC=0x00 TTL=64 ID=53765 DF PROTO=ICMP TYPE=8 CODE=0 ID=12294 SEQ=10
Dec 16 20:59:08 localhost kernel: RULE 1 -- REJECT IN=eth1 OUT= MAC=08:00:27:14:fd:5f:0a:00:27:00:00:01:08:00
SRC=192.168.57.1 DST=192.168.57.4 LEN=84 TOS=0x00 PREC=0x00 TTL=64 ID=53975 DF PROTO=ICMP TYPE=8 CODE=0 ID=12294 SEQ=11
Dec 16 20:59:09 localhost kernel: RULE 1 -- REJECT IN=eth1 OUT= MAC=08:00:27:14:fd:5f:0a:00:27:00:00:01:08:00
SRC=192.168.57.1 DST=192.168.57.4 LEN=84 TOS=0x00 PREC=0x00 TTL=64 ID=54120 DF PROTO=ICMP TYPE=8 CODE=0 ID=12294 SEQ=12
```

## 5 CONCLUSÃO

---

Com a realização deste trabalho pratico, acreditamos que a criação e a manipulação de regras de firewall não são tarefa fácil, e consiste num trabalho árduo que necessita de melhoramentos constantes para oferecer ao sistema uma proteção eficaz.

Ao realizar as tarefas propostas, e depois aplicar os mesmos passos numa ferramenta com interface gráfica, gostaríamos de adicionar que estas facilitam e muito a configuração e construção de uma firewall, sendo que o processo é bastante simples e intuitivo. A construção de regras do fwbuilder consiste apenas em o ato de fazer um “drag and drop” das diferentes definições que queremos, e podemos construir um conjunto de regras complexo com uma interface bastante simples.

## 6 REFERÊNCIAS

---

Rusty Russell, M. N. (s.d.). Obtido de <https://linux.die.net/man/8/iptables>

vkfwb. (s.d.). Obtido de howtoforge: <https://www.howtoforge.com/getting-started-with-firewall-builder>

A VM usada é o *CentOS 6.1*.