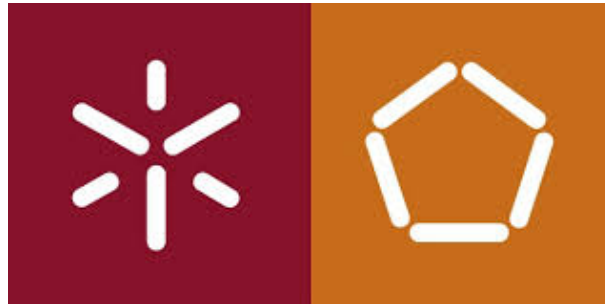


UNIVERSIDADE DO MINHO



Trabalho Prático 5a

IP Tables

MESTRADO INTEGRADO EM ENGENHARIA INFORMÁTICA

SEGURANÇA EM REDES
(1º SEMESTRE - 2018/2019)

a70565	Bruno Arieira
a73883	Cesário Perneta
a73974	Daniel Vieira
a78494	José Dias

15 de Dezembro de 2018

Resumo

"A **firewall** is a network security device that monitors incoming and outgoing network traffic and decides whether to allow or block specific traffic based on a defined set of security rules. **Firewalls** have been a first line of defense in network security for over 25 years. They establish a barrier between secured and controlled internal networks that can be trusted and untrusted outside networks, such as the Internet."

Este trabalho prático foi realizado no âmbito da unidade curricular Segurança em Redes, e tem como principal objetivo a interação e experiência com **firewalls** e respetiva configuração.

Conteúdo

1	Introdução	3
2	Contextualização	4
3	Logbook	5
3.1	Tarefa 1	5
3.1.1	Exercício 1	5
3.1.2	Exercício 2	7
3.1.3	Exercício 3	7
3.1.4	Exercício 4	8
3.1.5	Exercício 5	9
3.2	Tarefa 2	10
3.2.1	Exercício 1	10
3.2.2	Exercício 2	11
3.2.3	Exercício 3	11
3.2.4	Exercício 4	12
3.2.5	Exercício 5	13
3.3	Tarefa 3	14
3.3.1	Exercício 2	14
3.3.2	Exercício 3	15
3.3.3	Exercício 4	15
3.3.4	Exercício 5	16
3.3.5	Exercício 6	16
3.3.6	Exercício 7	17
3.4	Tarefa 4	18
4	Conclusão	19

1 Introdução

Neste quinto trabalho prático, temos como principal objetivo aplicar o conhecimento adquirido nas aulas de Seguranças em Redes, relativamente à matéria lecionada sobre *Protocolos de Internet* mais especificamente com a utilização do comando **iptables** com principal objetivo de adquirir algumas competências no âmbito da configuração de *firewalls* de modo a atingir algum nível de segurança.

Em conformidade com o enunciado proposto, para a realização deste trabalho, inicialmente foi necessária a instalação do sistema operativo *CentOS 6.10*, que será usado como servidor, e o *Kali* (tal como recomendado), que será utilizado como cliente. Como já foi mencionado, **Firewall** é um sistema informático constituído por hardware e software específico cuja função é reforçar a segurança entre duas redes, habitualmente a nossa rede interna ou Intranet e as redes externas que constituem a Internet.

Para o desenvolvimento deste trabalho, foi delineado que todos os elementos deviam analisar e perceber os conceitos implícitos para este guião, por forma a facilitar a resolução do trabalho proposto. Com a devida consolidação dos termos indispensáveis, leitura do enunciado e instalação das ferramentas necessárias, procedemos à discussão e elaboração das duas tarefas propostas, onde todos os elementos trabalharam de forma uniforme.

2 Contextualização

Para a interpretação e resolução deste trabalho prático é necessária a consolidação de alguns conceitos importantes.

Usufruindo da capacidade de processamento e filtragem de todo o tráfego que passa na pilha de protocolos, pode-se configurar uma máquina como router e como firewall, sendo que neste trabalho iremos abordar esta última vertente. Tal como já foi referido anteriormente uma *firewall* é uma barreira de proteção que ajuda a bloquear o acesso de conteúdo malicioso, mas sem impedir que os dados que precisam de transitar continuem fluindo. Este conceito existe na forma de software e de hardware, a combinação de ambos é chamado tecnicamente de "*appliance*", sendo que para este trabalho iremos abordar a nível de software.

Numa *firewall*, usando a tabela filter, todos os pacotes estão sujeitos a três cadeias primárias de regras:

- **Input:** aplicam-se a todos os pacotes que são endereçados para a máquina local;
- **Forward:** aplicam-se a todos os pacotes que chegam à máquina local e que se destinam a outras máquinas;
- **Output:** aplicam-se a todos os pacotes gerados na máquina local e que se destinam a outras máquinas;

Para configurar estas cadeias é necessário um administrador, que o pode fazer usando um comando do linux chamado **iptables**. A função primária do iptables é fazer a análise do tráfego de redes recebido pelo computador, no qual é feita uma comparação com o conjunto de regras preestabelecidas nas configurações. O processamento dos pacotes é feito pelo iptables a partir de uma estrutura que contém suas camadas (*tabelas*) e cadeias (*chains*). Assim sendo, temos 3 tabelas implícitas:

- **Filter:** nesta camada define-se a aceitação de um pacote, caso ele possa entrar ou não. Esta tabela contém as três cadeias referidas anteriormente (*Input*, *Forward* e *Output*). Quanto às ações que esta tabela pode aplicar temos, a *REJECT* (rejeita o pacote), a *ACCEPT* (aceita o pacote), a *DROP* (igual ao *Reject* mas não envia mensagem de erro) e a *LOG* (pacotes continuam a ser avaliados consoantes as regras não havendo ainda terminado a respetiva avaliação).
- **NAT (Network Address Translation):** realiza a tradução dos endereços que passam pelo router no qual opera, verificando se houve alteração de endereços IP origem ou destino. Esta tabela possui três cadeias: *PREROUTING* (aplicam regras aos pacotes que entram no firewall), *POSTROUTING* (pacotes que chegam à máquina local e destinam-se a outras máquinas) e *OUTPUT* (opera pacotes da máquina local). As ações desta tabela são a *SNAT* (faz a troca dos endereços IP de origem), *DNAT* (altera os endereços de IP de destino), *MASQUERADE* (mascara o IP) e *REDIRECT* (redireciona o pacote para uma porta local).
- **Mangle:** camada que tem a função de especificar ações especiais que devem ser aplicadas no tráfego que passa pelas cadeias. Esta tabela tem o poder de conseguir usar *chains* relativas à *NAT* ou *Filter*, sendo no total 5.

3 Logbook

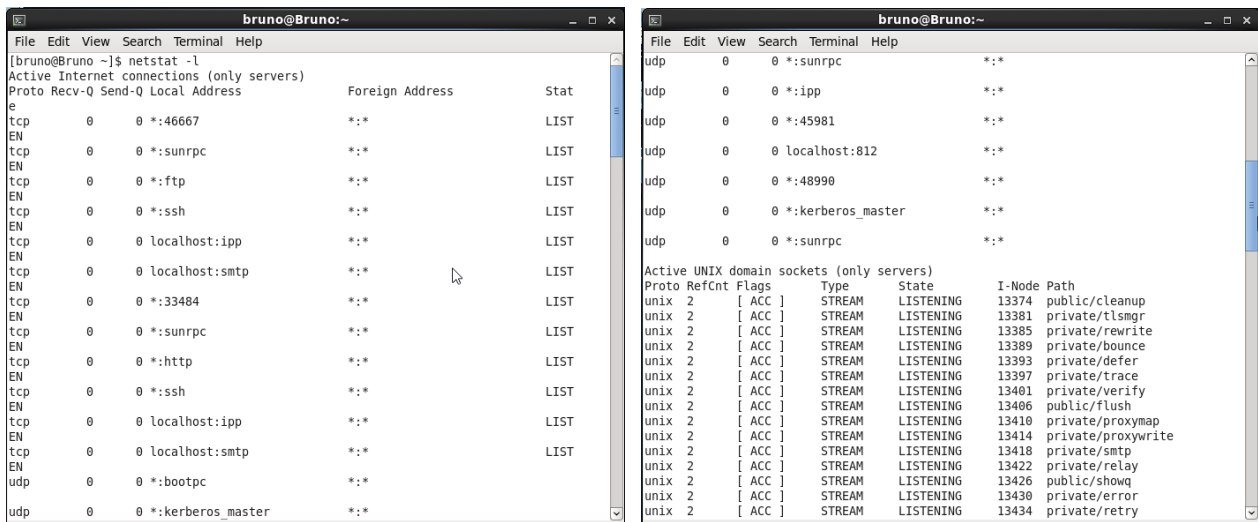
Com a criação deste *logbook* apresentamos as respostas relativamente às questões do enunciado que nos foi proposto, onde especificamos as opções que foram tomadas ao longo deste trabalho prático. Decidimos responder a todas as perguntas que contêm tarefas assinaladas a vermelho e a outras que achemos relevantes comentar, já que para algumas questões basta apenas seguir os passos.

3.1 Tarefa 1

3.1.1 Exercício 1

Questão: Execute o comando **netstat -l** que lhe permitirá verificar se os serviços desejados estão todos devidamente preparados; pode ainda tentar abrir a homepage e aceder por *ftp* e *ssh*, tudo no **localhost**; registe no seu *logbook* o resultado obtido e comente eventuais discrepâncias.

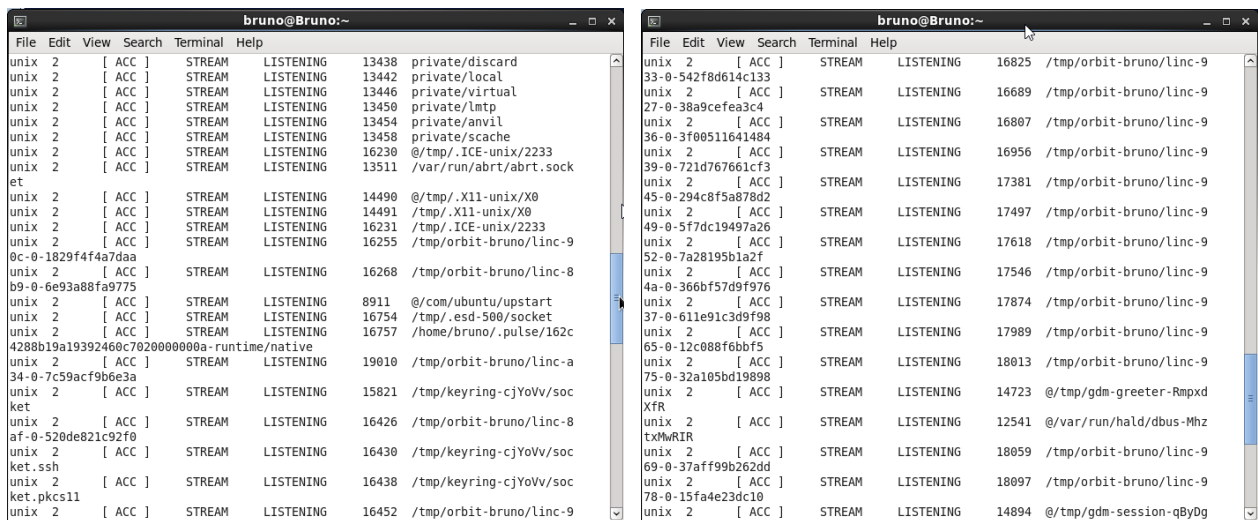
Resposta: Depois da devida instalação do sistema operativo *CentOS*, procedemos à instalação do serviço FTP e posteriormente verificamos se os serviços *sshd*, *httpd* e *vsftpd* estavam ativos e em execução, reparando que apenas o último mencionado não se encontrava ativo.



Proto	Recv-Q	Send-Q	Local Address	Foreign Address	Stat
tcp	0	0	*:*		LIST
tcp	0	0	*:sunrpc		LIST
tcp	0	0	*:ftp		LIST
tcp	0	0	*:ssh		LIST
tcp	0	0	localhost:ipp		LIST
tcp	0	0	localhost:smtp		LIST
tcp	0	0	*:33484		LIST
tcp	0	0	*:sunrpc		LIST
tcp	0	0	*:http		LIST
tcp	0	0	*:ssh		LIST
tcp	0	0	localhost:ipp		LIST
tcp	0	0	localhost:smtp		LIST
udp	0	0	*:bootpc		
udp	0	0	*:kerberos_master		

Proto	RefCnt	Flags	Type	State	I-Node	Path
unix	2	[ACC]	STREAM	LISTENING	13374	public/cleanup
unix	2	[ACC]	STREAM	LISTENING	13381	private/tlsgr
unix	2	[ACC]	STREAM	LISTENING	13385	private/rewrite
unix	2	[ACC]	STREAM	LISTENING	13389	private/bounce
unix	2	[ACC]	STREAM	LISTENING	13393	private/defer
unix	2	[ACC]	STREAM	LISTENING	13397	private/trace
unix	2	[ACC]	STREAM	LISTENING	13401	private/verify
unix	2	[ACC]	STREAM	LISTENING	13406	public/flush
unix	2	[ACC]	STREAM	LISTENING	13410	private/proxymap
unix	2	[ACC]	STREAM	LISTENING	13414	private/proxywrite
unix	2	[ACC]	STREAM	LISTENING	13418	private/smtp
unix	2	[ACC]	STREAM	LISTENING	13422	private/relay
unix	2	[ACC]	STREAM	LISTENING	13426	public/showq
unix	2	[ACC]	STREAM	LISTENING	13430	private/error
unix	2	[ACC]	STREAM	LISTENING	13434	private/retry

Figura 1: Resultados do comando netstat -l (1)



Proto	Recv-Q	Send-Q	Local Address	Foreign Address	Stat
tcp	0	0	*:46667		LIST
tcp	0	0	*:sunrpc		LIST
tcp	0	0	*:ftp		LIST
tcp	0	0	*:ssh		LIST
tcp	0	0	localhost:ipp		LIST
tcp	0	0	localhost:smtp		LIST
tcp	0	0	*:33484		LIST
tcp	0	0	*:sunrpc		LIST
tcp	0	0	*:http		LIST
tcp	0	0	*:ssh		LIST
tcp	0	0	localhost:ipp		LIST
tcp	0	0	localhost:smtp		LIST
udp	0	0	*:bootpc		
udp	0	0	*:kerberos_master		

Proto	RefCnt	Flags	Type	State	I-Node	Path
unix	2	[ACC]	STREAM	LISTENING	13438	private/discard
unix	2	[ACC]	STREAM	LISTENING	13442	private/local
unix	2	[ACC]	STREAM	LISTENING	13446	private/virtual
unix	2	[ACC]	STREAM	LISTENING	13450	private/lmtp
unix	2	[ACC]	STREAM	LISTENING	13454	private/anvil
unix	2	[ACC]	STREAM	LISTENING	13458	private/scache
unix	2	[ACC]	STREAM	LISTENING	16230	@/tmp/.ICE-unix/2233
unix	2	[ACC]	STREAM	LISTENING	13511	/var/run/abrt/abrt.sock
unix	2	[ACC]	STREAM	LISTENING	14490	@/tmp/.X11-unix/X0
unix	2	[ACC]	STREAM	LISTENING	14491	/tmp/.X11-unix/X0
unix	2	[ACC]	STREAM	LISTENING	16231	/tmp/.ICE-unix/2233
unix	2	[ACC]	STREAM	LISTENING	16255	/tmp/orbit-bruno/linc-9
unix	2	[ACC]	STREAM	LISTENING	16268	/tmp/orbit-bruno/linc-8
unix	2	[ACC]	STREAM	LISTENING	8911	@/com/ubuntu/upstart
unix	2	[ACC]	STREAM	LISTENING	16754	/tmp/.esd-500/socket
unix	2	[ACC]	STREAM	LISTENING	16757	/home/bruno/.pulse/162c
unix	2	[ACC]	STREAM	LISTENING	19010	/tmp/orbit-bruno/linc-a
unix	2	[ACC]	STREAM	LISTENING	15821	/tmp/keyring-cjVoVv/soc
unix	2	[ACC]	STREAM	LISTENING	16426	/tmp/orbit-bruno/linc-8
unix	2	[ACC]	STREAM	LISTENING	16430	/tmp/keyring-cjVoVv/soc
unix	2	[ACC]	STREAM	LISTENING	16438	/tmp/keyring-cjVoVv/soc
unix	2	[ACC]	STREAM	LISTENING	16452	/tmp/orbit-bruno/linc-9

Proto	RefCnt	Flags	Type	State	I-Node	Path
unix	2	[ACC]	STREAM	LISTENING	16825	/tmp/orbit-bruno/linc-9
unix	2	[ACC]	STREAM	LISTENING	16689	/tmp/orbit-bruno/linc-9
unix	2	[ACC]	STREAM	LISTENING	16807	/tmp/orbit-bruno/linc-9
unix	2	[ACC]	STREAM	LISTENING	16956	/tmp/orbit-bruno/linc-9
unix	2	[ACC]	STREAM	LISTENING	17381	/tmp/orbit-bruno/linc-9
unix	2	[ACC]	STREAM	LISTENING	17497	/tmp/orbit-bruno/linc-9
unix	2	[ACC]	STREAM	LISTENING	17618	/tmp/orbit-bruno/linc-9
unix	2	[ACC]	STREAM	LISTENING	17546	/tmp/orbit-bruno/linc-9
unix	2	[ACC]	STREAM	LISTENING	17874	/tmp/orbit-bruno/linc-9
unix	2	[ACC]	STREAM	LISTENING	17989	/tmp/orbit-bruno/linc-9
unix	2	[ACC]	STREAM	LISTENING	18013	/tmp/orbit-bruno/linc-9
unix	2	[ACC]	STREAM	LISTENING	14723	@/tmp/gdm-greeter-Rmxd
unix	2	[ACC]	STREAM	LISTENING	12541	@/var/run/hald/dbus-Mhz
unix	2	[ACC]	STREAM	LISTENING	18059	/tmp/orbit-bruno/linc-9
unix	2	[ACC]	STREAM	LISTENING	18097	/tmp/orbit-bruno/linc-9
unix	2	[ACC]	STREAM	LISTENING	14894	@/tmp/gdm-session-qByDg

Figura 2: Resultados do comando netstat -l (2)

```

unix 2      [ ACC ]     STREAM  LISTENING  14894  @/tmp/gdm-session-qByDg
OpI
unix 2      [ ACC ]     STREAM  LISTENING  18488  /tmp/orbit-bruno/linc-a
19-0-67723eab197e5
unix 2      [ ACC ]     STREAM  LISTENING  18496  /tmp/orbit-bruno/linc-9
f5-0-6545c2871e81e
unix 2      [ ACC ]     STREAM  LISTENING  18525  /tmp/orbit-bruno/linc-9
f3-0-6b3fc57b25d0e
unix 2      [ ACC ]     STREAM  LISTENING  18546  /tmp/orbit-bruno/linc-9
f4-0-42b9264f2dc50
unix 2      [ ACC ]     STREAM  LISTENING  12410  /var/run/cups/cups.sock
unix 2      [ ACC ]     STREAM  LISTENING  18571  /tmp/orbit-bruno/linc-9
f2-0-19b1064c37f46
unix 2      [ ACC ]     STREAM  LISTENING  11973  /var/run/rpcbind.sock
unix 2      [ ACC ]     STREAM  LISTENING  12049  /var/run/dbus/system_bu
s_socket
unix 2      [ ACC ]     STREAM  LISTENING  12534  @/var/run/hald/dbus-0mG
XFDgQIO
unix 2      [ ACC ]     STREAM  LISTENING  12495  /var/run/acpid.socket
unix 2      [ ACC ]     STREAM  LISTENING  19325  @/tmp/fam-root-
unix 2      [ ACC ]     STREAM  LISTENING  16110  @/tmp/dbus-isa0CGNMLq
[bruno@Bruno ~]$

```

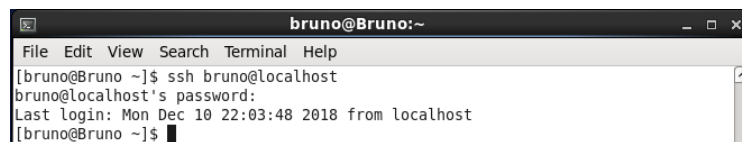
Figura 3: Resultados do comando `netstat -l` (3)

Podemos confirmar, a partir dos prints representados, todo o output depois de inserir o comando **netstat -l**. A partir deste comando pudemos verificar:

- Conexões TCP ativas (FTP, SSH, HTTP);
- Portas TCP e UDP;
- Estatísticas Ethernet;
- Estatísticas de IPv4 e IPv6;
- Tabela de routing IP;

Assim, com a execução do comando **netstat -l** podemos denotar a lista de todos os sockets abertos para conexão com a máquina local (onde se evidencia os sockets TCP: SSH, FTP e HTTP).

Para verificar que todos os serviços estão a funcionar como o pretendido, decidimos realizar conexões para cada um deles.

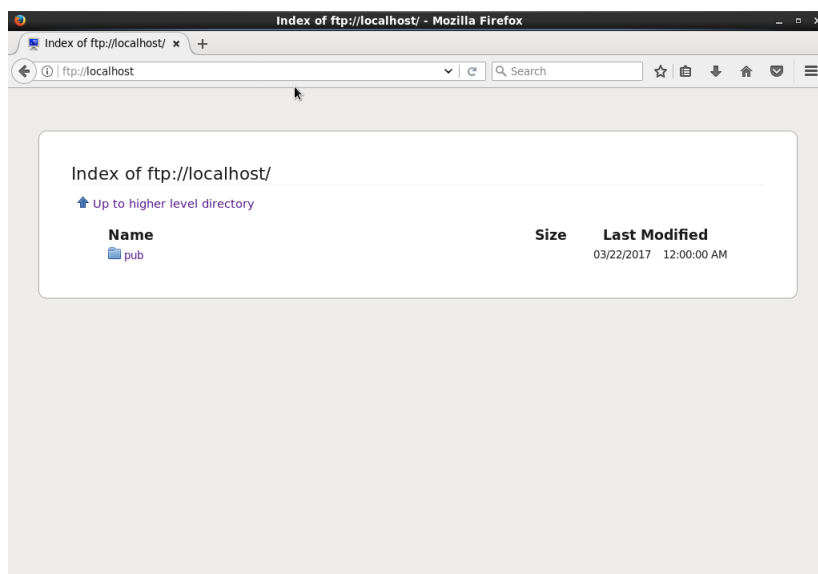


```

bruno@Bruno:~
File Edit View Search Terminal Help
[bruno@Bruno ~]$ ssh bruno@localhost
bruno@localhost's password:
Last login: Mon Dec 10 22:03:48 2018 from localhost
[bruno@Bruno ~]$

```

Figura 4: Conexão ssh



Index of ftp://localhost/ - Mozilla Firefox

Index of ftp://localhost/

Up to higher level directory

Name	Size	Last Modified
pub		03/22/2017 12:00:00 AM

Figura 5: Conexão ftp



Figura 6: Conexão http

3.1.2 Exercício 2

Depois da execução do comando **system-config-firewall-tui** como super-utilizador, podemos observar que a firewall já se encontrava ativa.

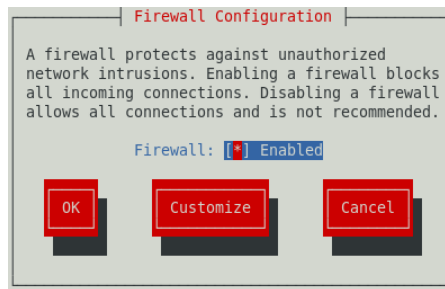
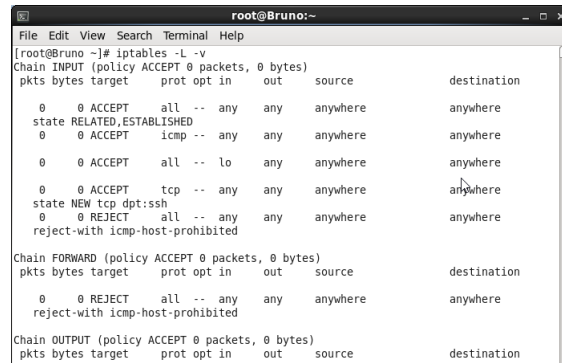


Figura 7: Janela consequente do comando system-config-firewall-tui

3.1.3 Exercício 3

Questão A: Registe o resultado obtido e indique, justificando quais são as políticas por defeito para cada uma das cadeias **INPUT**, **FORWARD** e **OUTPUT**.

Resposta (A): Como se pode verificar, para todas as cadeias está implícita a política ACCEPT.



```
root@Bruno:~  
[root@Bruno ~]# 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
```

Figura 8: Resultado do comando `iptables -L -v`

Para a cadeia **INPUT** são aceites pacotes com:

- Quaisquer origem e destino, desde que o estado conntrack seja *RELATED* (o pacote é utilizado para inicializar uma nova ligação) ou *ESTABLISHED* (pacotes com conexão já estabelecida);
- Em qualquer interface, com qualquer origem e destino, desde que o protocolo seja ICMP;
- Um qualquer estado que entrem pela interface *lo* (localhost);
- Do protocolo TCP em qualquer interface vindo de qualquer origem para qualquer destino, desde que seja através de ssh, no estado conntrack NEW (força que seja somente aceite o pacote correspondente ao início de sessão);

Qualquer outro pacote é rejeitado, apresentando a mensagem de erro associada a *icmp-host-prohibited*.

A cadeia **FORWARD** rejeita todos os pacotes em todas as interfaces, todos os protocolos de qualquer origem e para qualquer destino. Como o servidor não está configurado como router, todos os pacotes na cadeia forward são rejeitados (*icmp-host-prohibited*).

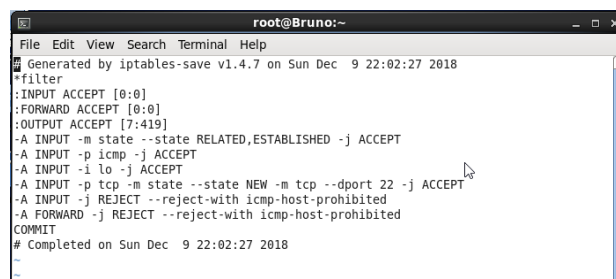
Para a cadeia **OUTPUT** não existem quaisquer regras pois todos os pacotes passam sem qualquer tipo de permissão, como se pode observar.

Questão B: Comente o nível de segurança e demais informação que consiga extrair do resultado da execução do comando.

Resposta (B): Como podemos denotar após o estabelecimento da conexão não existe qualquer tipo de filtro dos pacotes. O dispositivo está sujeito a ataques por *ICMP flood*, sendo que aceita qualquer pacote ICMP.

3.1.4 Exercício 4

Questão: Execute o comando `iptables-save > iptables.dump` e guarde o ficheiro “iptables.dump” (que deverá listar no seu *logbook*, em anexo), por questões de segurança.



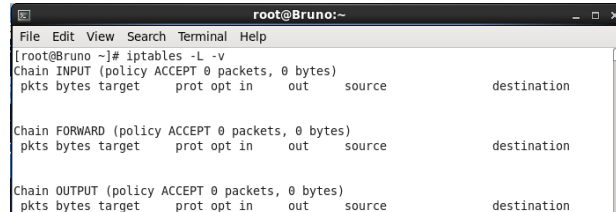
```
root@Bruno:~  
Generated by iptables-save v1.4.7 on Sun Dec 9 22:02:27 2018  
#filter  
:INPUT ACCEPT [0:0]  
:FORWARD ACCEPT [0:0]  
:OUTPUT ACCEPT [7:419]  
-A INPUT -m state --state RELATED,ESTABLISHED -j ACCEPT  
-A INPUT -p icmp -j ACCEPT  
-A INPUT -i lo -j ACCEPT  
-A INPUT -p tcp -m state --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 Sun Dec 9 22:02:27 2018
```

Figura 9: Comando que guarda as configurações

3.1.5 Exercício 5

Questão: Registre e analise as novas regras, procurando verificar se são mais seguras ou não e porquê.

Resposta: Ao desativar a firewall, todas as regras foram removidas de todas as tabelas. Com isto, o sistema está sujeito a qualquer ataque, aceitando todos os pacotes que entram, que saem ou que são reencaminhados pelo sistema.

A terminal window titled 'root@Bruno:~' showing the output of the command 'iptables -L -v'. The output lists three chains: INPUT, FORWARD, and OUTPUT, all with a policy of ACCEPT. Each chain has a table with columns for pkts, bytes, target, protocol, options, in, out, source, and destination. The INPUT chain has 0 packets and 0 bytes. The FORWARD chain has 0 packets and 0 bytes. The OUTPUT chain has 0 packets and 0 bytes.

```
root@Bruno:~# 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
```

Figura 10: Comando iptables -L -v com o firewall desabilitado

3.2 Tarefa 2

3.2.1 Exercício 1

Questão: Comece por verificar a conectividade, executando o comando `ping <ip_address>` onde `<ip_address>` representa, naturalmente, o endereço IP do servidor (essa mesma convicção é usada nas restantes fases e tarefas).

Resposta: Para esta fase utilizamos dois sistemas operativos virtualizados, na mesma máquina. Inicialmente, foi necessário configurar na *VirtualBox* os sistemas, de modo a que estes partilhassem uma rede interna. Posteriormente, definimos o IP 192.168.0.1 para o sistema servidor e o IP 192.168.0.2 para o sistema cliente.

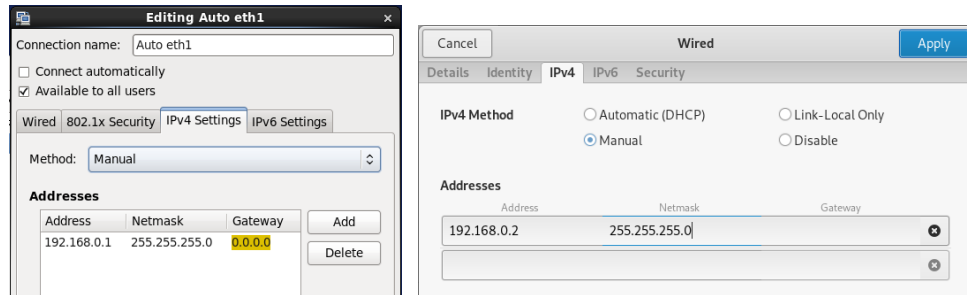


Figura 11: Configuração do Ip do servidor e do cliente para a interface da rede interna

Depois de definidos os IP's, verificamos a sua integridade executando o comando `ifconfig` em ambas as máquinas.

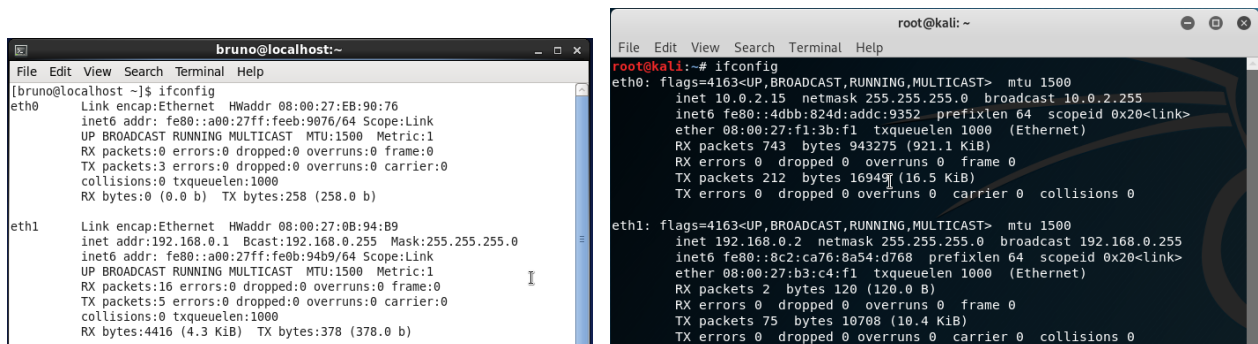


Figura 12: Resultado do comando ifconfig do servidor e do cliente

De seguida, executamos o ping em cada uma das máquina por forma a ser verificada a conectividade entre as mesmas.

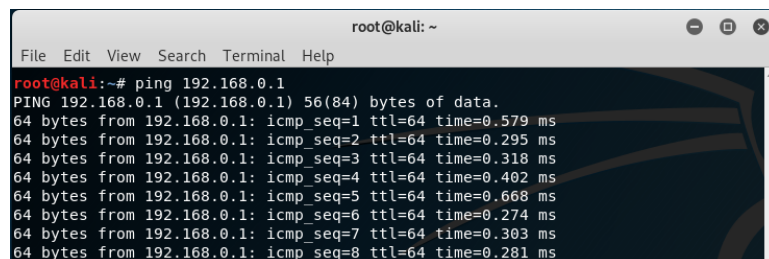
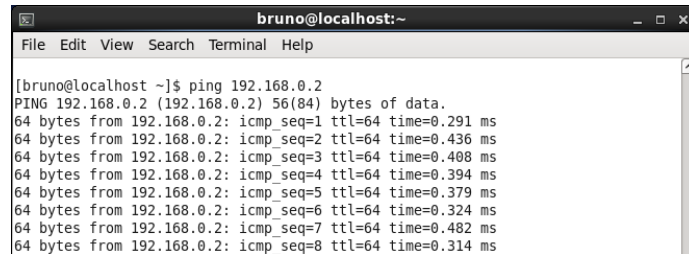


Figura 13: Resultado do ping do Cliente para o Servidor



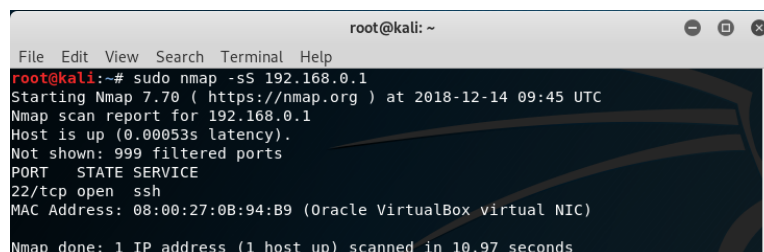
```
bruno@localhost:~  
File Edit View Search Terminal Help  
[bruno@localhost ~]$ ping 192.168.0.2  
PING 192.168.0.2 (192.168.0.2) 56(84) bytes of data.  
64 bytes from 192.168.0.2: icmp_seq=1 ttl=64 time=0.291 ms  
64 bytes from 192.168.0.2: icmp_seq=2 ttl=64 time=0.436 ms  
64 bytes from 192.168.0.2: icmp_seq=3 ttl=64 time=0.408 ms  
64 bytes from 192.168.0.2: icmp_seq=4 ttl=64 time=0.394 ms  
64 bytes from 192.168.0.2: icmp_seq=5 ttl=64 time=0.379 ms  
64 bytes from 192.168.0.2: icmp_seq=6 ttl=64 time=0.324 ms  
64 bytes from 192.168.0.2: icmp_seq=7 ttl=64 time=0.482 ms  
64 bytes from 192.168.0.2: icmp_seq=8 ttl=64 time=0.314 ms
```

Figura 14: Resultado do ping do Servidor para o Cliente

3.2.2 Exercício 2

Questão: Execute o comando `nmap -sS <ip_address>`. Que informação lhe forneceu o programa?

Resposta: O comando `nmap` podemos visualizar o numero de portas que se encontram abertas.



```
root@kali: ~  
File Edit View Search Terminal Help  
root@kali:~# sudo nmap -sS 192.168.0.1  
Starting Nmap 7.70 ( https://nmap.org ) at 2018-12-14 09:45 UTC  
Nmap scan report for 192.168.0.1  
Host is up (0.00053s latency).  
Not shown: 999 filtered ports  
PORT      STATE SERVICE  
22/tcp    open  ssh  
MAC Address: 08:00:27:0B:94:B9 (Oracle VirtualBox virtual NIC)  
Nmap done: 1 IP address (1 host up) scanned in 10.97 seconds
```

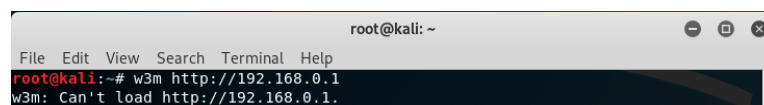
Figura 15: Resultado do scan com nmap do Cliente para Servidor

Apartir do resultado demonstrado, verificamos apenas o serviço ssh, na porta 22 através do protocolo TCP está exposto às conexões exteriores, pois a firewall permite pacotes relativos a novas conexões através de ssh.

3.2.3 Exercício 3

Questão: Execute o comando `w3m http://<ip_address>`. Conseguiu visualizar uma página? Registe a resposta que obteve.

Resposta:



```
root@kali: ~  
File Edit View Search Terminal Help  
root@kali:~# w3m http://192.168.0.1  
w3m: Can't load http://192.168.0.1.
```

Figura 16: Resultado do comando w3m para ip do Servidor

Como seria de esperar, não é possível realizar novas conexões http, devido às regras definidas na firewall. Para conseguir-mos gerar um resultado, desabilitamos a firewall, gerando a página teste do Apache.

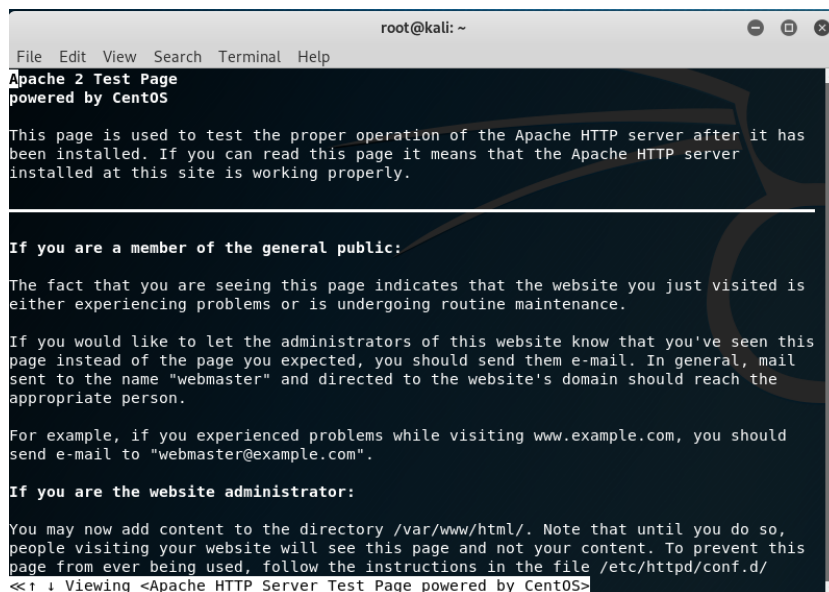


Figura 17: Página obtida com o comando w3m com a desativação da firewall

3.2.4 Exercício 4

Questão: Execute de seguida o comando `ftp <ip_address>`. Conseguiu aceder ao servidor? Registe a resposta que obteve.

Resposta: Como se está a tentar estabelecer uma conexão via ftp de fora para o servidor, com a firewall ativa, os pacotes não serão aceites.

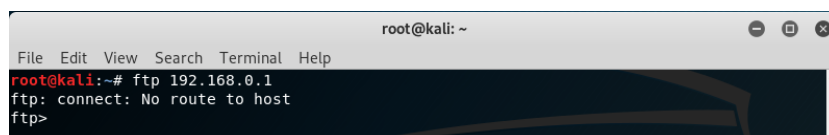


Figura 18: Tentiva de conexão via ftp com o servidor

Para permitir a conexão via ftp com o servidor, foi necessário desativar a firewall.

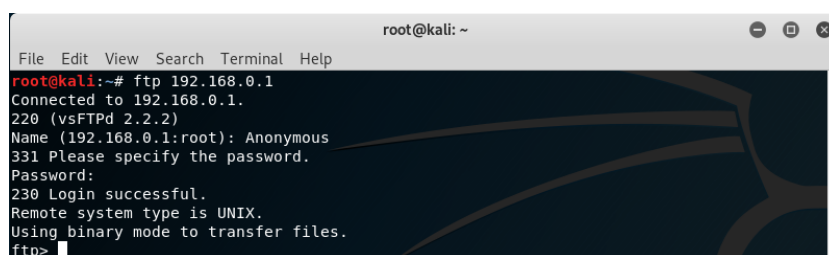
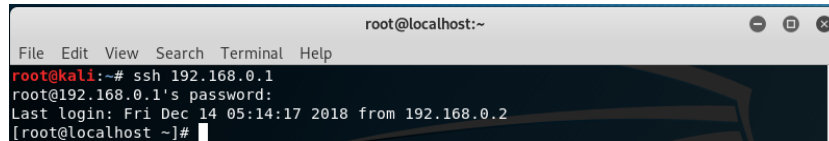


Figura 19: Tentiva de conexão via ftp com o servidor, com a firewall desabilitada

3.2.5 Exercício 5

Questão: Execute finalmente o comando `ssh <ip_address>`. Conseguiu aceder ao servidor? Registe a resposta que obteve.

Resposta: Como era de prever, observando o mapeamento de portas ativas no exercício 2 desta tarefa e tendo em conta as regras da firewall do servidor enunciadas anteriormente, a nova conexão ssh da máquina do cliente com o servidor foi aceite.

A terminal window titled 'root@localhost:~' with a menu bar (File, Edit, View, Search, Terminal, Help). The terminal shows the command 'ssh 192.168.0.1' being executed. The prompt changes to 'root@192.168.0.1's password:', followed by a password input (indicated by dots). The terminal then displays 'Last login: Fri Dec 14 05:14:17 2018 from 192.168.0.2' and the prompt returns to '[root@localhost ~]#'.

```
root@localhost:~  
File Edit View Search Terminal Help  
root@kali:~# ssh 192.168.0.1  
root@192.168.0.1's password:  
Last login: Fri Dec 14 05:14:17 2018 from 192.168.0.2  
[root@localhost ~]#
```

Figura 20: Conexão via ssh com o servidor

3.3 Tarefa 3

3.3.1 Exercício 2

Questão: Execute novamente o comando **iptables -L -v**. Registre as alterações que observa e procure interpretar as diversas regras que foram alteradas, à luz das opções escolhidas na operação anterior.

Resposta:

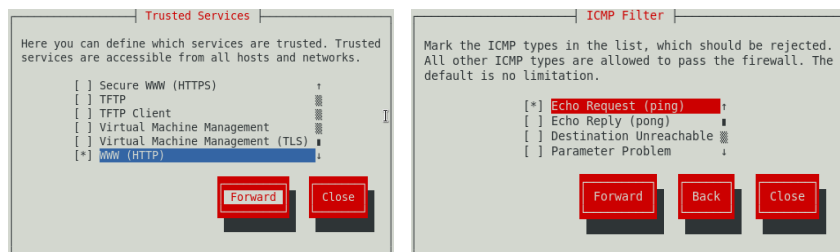


Figura 21: Configurações necessárias na firewall

Depois da configuração da firewall com os protocolos selecionados pedidos no enunciado e com a opção **Echo Request** selecionada, executamos o comando **iptables -L -v** para verificar as alterações.

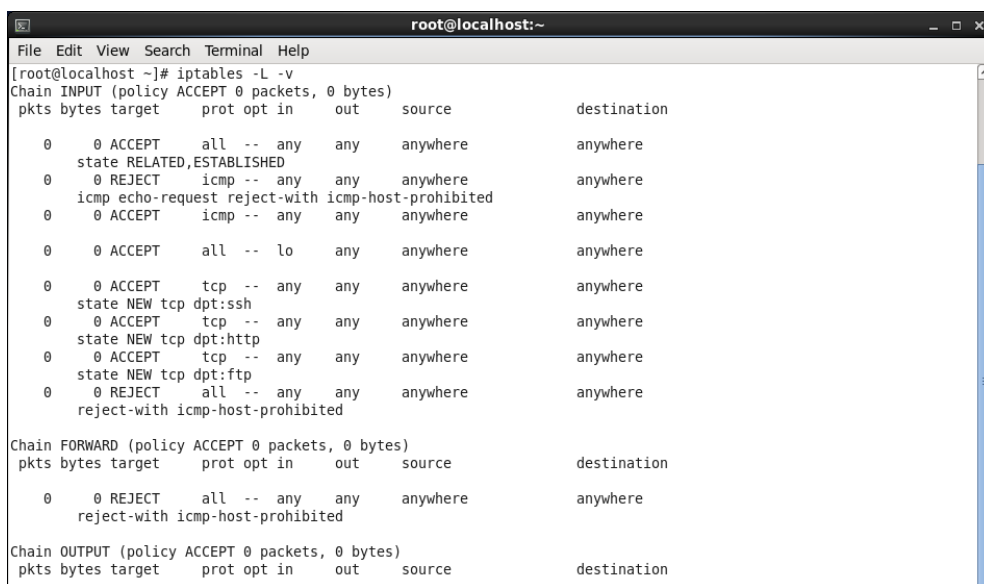


Figura 22: Resultado do comando **iptables -L -v** depois das configurações na firewall

Com o resultado gerado, denotamos que foram adicionadas 3 regras á iptables padrão, na tabela **INPUT**:

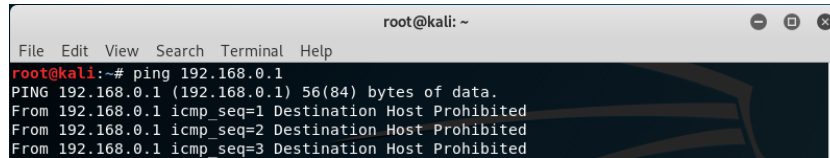
- Permissão de novas conexões http, sob o protocolo *HTTP*;
- Permissão de novas conexões ftp, sob o protocolo *FTP*;
- Rejeição de pacotes echo-request através do protocolo ICMP, com a mensagem de erro *icmp-host-prohibited*;

Foi também executada a permissão para o protocolo ssh, mas como a firewall padrão já permitia esse tipo de conexões, não se tornou uma nova regra.

3.3.2 Exercício 3

Questão: Execute o comando `ping <ip_address>`. Registre a resposta obtida e comente-a.

Resposta:



```
root@kali: ~  
File Edit View Search Terminal Help  
root@kali:~# ping 192.168.0.1  
PING 192.168.0.1 (192.168.0.1) 56(84) bytes of data.  
From 192.168.0.1 icmp_seq=1 Destination Host Prohibited  
From 192.168.0.1 icmp_seq=2 Destination Host Prohibited  
From 192.168.0.1 icmp_seq=3 Destination Host Prohibited
```

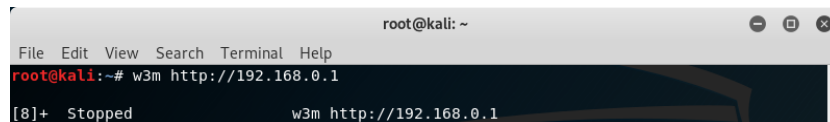
Figura 23: Resultado do comando ping do cliente para o servidor, com as mudanças na firewall

Na alínea anterior, adicionamos uma regra de que os pedidos *echo-request*(ping), através do protocolo ICMP, seriam rejeitados pela firewall. Portanto foi esse o output que obtivemos. Apesar de haver uma outra regra diz que um pacote *ping* é aceite, dará este output visto que esta última regra encontra-se depois da primeira. Logo faz match com a 1ª regra resultando no output da imagem.

3.3.3 Exercício 4

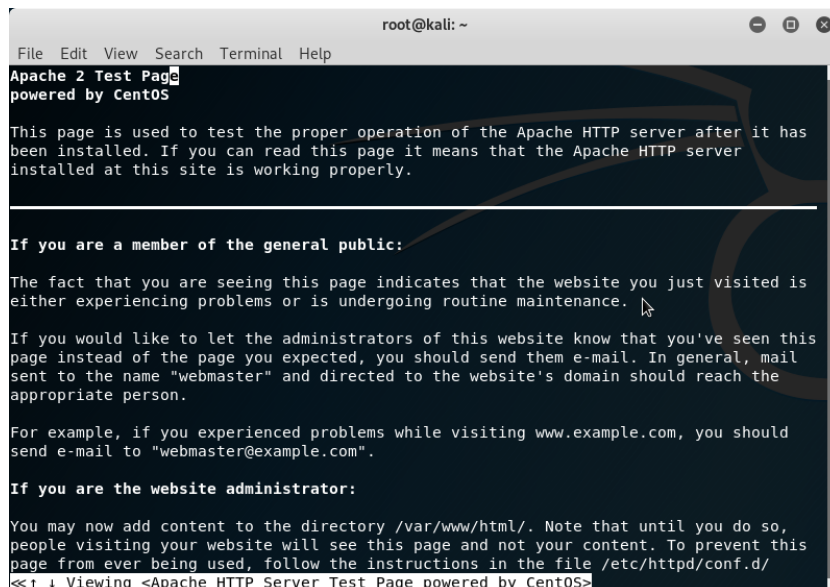
Questão: Execute o comando `w3m http://<ip_address>`. Desta vez conseguiu visualizar uma página? Registre a resposta obtida.

Resposta:



```
root@kali: ~  
File Edit View Search Terminal Help  
root@kali:~# w3m http://192.168.0.1  
[8]+ Stopped w3m http://192.168.0.1
```

Figura 24: Comando w3m do cliente para o servidor, com as mudanças na firewall



```
root@kali: ~  
File Edit View Search Terminal Help  
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, 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/  
<<↑ Viewing <Apache HTTP Server Test Page powered by CentOS>
```

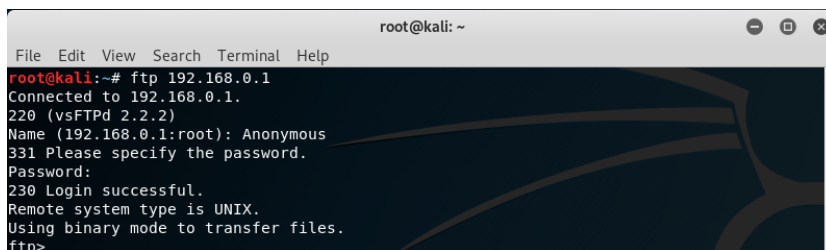
Figura 25: Página HTTP padrão

Visto que adicionamos novas permissões em que possibilitava o acesso a conexões do protocolo **HTTP**, conseguimos obter uma ligação, como era de prever.

3.3.4 Exercício 5

Questão: Execute de seguida o comando **ftp <ip_address>**. Desta vez conseguiu obter uma ligação? Registe a resposta obtida.

Resposta: Resultado do comando ping do cliente para o servidor, com as mudanças na firewall



```
root@kali: ~  
File Edit View Search Terminal Help  
root@kali:~# ftp 192.168.0.1  
Connected to 192.168.0.1.  
220 (vsFTPd 2.2.2)  
Name (192.168.0.1:root): Anonymous  
331 Please specify the password.  
Password:  
230 Login successful.  
Remote system type is UNIX.  
Using binary mode to transfer files.  
ftp>
```

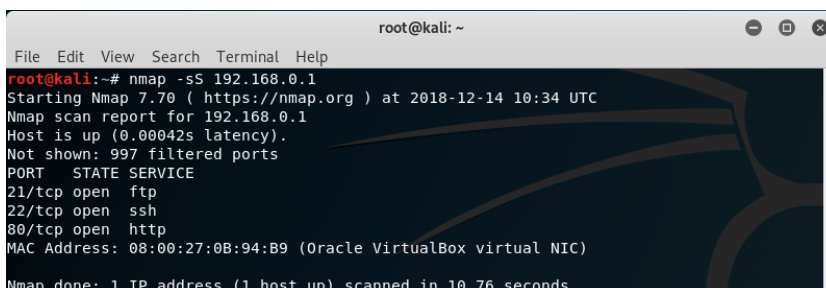
Figura 26: Resultado do comando ftp do cliente para o servidor, com as mudanças na firewall

Tal como na alínea anterior, adicionando a regra de acesso ao servidor **ftp**, fez com que houvesse uma ligação.

3.3.5 Exercício 6

Questão: Execute o comando **nmap -sS <ip_address>**. Registe a informação lhe forneceu desta vez o programa. Compare-a com a que obteve anteriormente e reflita sobre o nível de segurança atual.

Resposta:



```
root@kali: ~  
File Edit View Search Terminal Help  
root@kali:~# nmap -sS 192.168.0.1  
Starting Nmap 7.70 ( https://nmap.org ) at 2018-12-14 10:34 UTC  
Nmap scan report for 192.168.0.1  
Host is up (0.00042s latency).  
Not shown: 997 filtered ports  
PORT      STATE SERVICE  
21/tcp    open  ftp  
22/tcp    open  ssh  
80/tcp    open  http  
MAC Address: 08:00:27:0B:94:B9 (Oracle VirtualBox virtual NIC)  
Nmap done: 1 IP address (1 host up) scanned in 10.76 seconds
```

Figura 27: Resultado do comando nmap do cliente para o servidor, com as mudanças na firewall

Como podemos ver na imagem e, ao contrário do que acontecia na tarefa 2, há neste momento 3 portas ativas. Logo visto que há mais portas abertas, há mais possibilidades de haver ataques através dos protocolos *http* e *ftp*.

3.3.6 Exercício 7

Questão: Execute novamente o comando **iptables -L -v**. Registre as alterações que observa e procure justificar o que é possível observar, à luz da atividade desta tarefa.

Resposta:

```
[root@localhost ~]# iptables -L -v
Chain INPUT (policy ACCEPT 0 packets, 0 bytes)
pkts bytes target      prot opt in     out    source            destination
39 2715 ACCEPT     all  --  any    any    anywhere          anywhere
9 756 REJECT     icmp --  any    any    anywhere          anywhere
state RELATED,ESTABLISHED
icmp echo-request reject-with icmp-host-prohib
h icmp-host-prohibited
0 0 ACCEPT     icmp --  any    any    anywhere          anywhere
6 396 ACCEPT     all  --  lo     any    anywhere          anywhere
2 136 ACCEPT     tcp  --  any    any    anywhere          anywhere
state NEW tcp dpt:ssh
2 104 ACCEPT     tcp  --  any    any    anywhere          anywhere
state NEW tcp dpt:http
3 164 ACCEPT     tcp  --  any    any    anywhere          anywhere
state NEW tcp dpt:ftp
1985 87340 REJECT     all  --  any    any    anywhere          anywhere
reject-with icmp-host-prohib
ited

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-prohib
ited

Chain OUTPUT (policy ACCEPT 61 packets, 9813 bytes)
pkts bytes target      prot opt in     out    source            destination
[root@localhost ~]#
```

Figura 28: Resultado do comando **iptables -L -v**

Comparando este output com o o output no início da tarefa3, podemos ver que houve alterações nos valores *pkts* e *bytes*.

Podemos ver então que o **IPTABLES** está a processar tráfego, aceitando ou rejeitando pacotes, e podemos ver isso através dos valores diferentes de 0 de *bytes* e *pkts*, o que valida que as regras que implementamos estão a funcionar.

3.4 Tarefa 4

Nesta tarefa utilizamos a interface gráfica invocando o comando *system-config-firewall* como diz no enunciado.

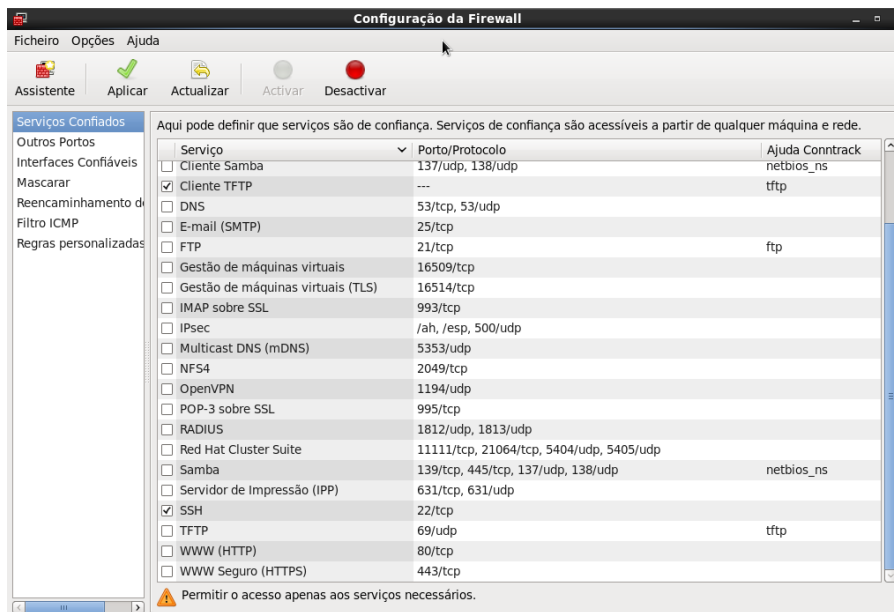


Figura 29: Resultado do comando *system-config-firewall*

Para alargar o número de pacotes, há que alargá-los a vários serviços de confiança. Para isso alargamos a serviços como **OpenVPN**. Achamos este serviço imprescindível. O OpenVPN é um serviço no qual é um software que cria redes privadas virtuais através de tuneis criptografados, portanto como é um serviço seguro achamos bem incluir nos pacotes que podemos receber.

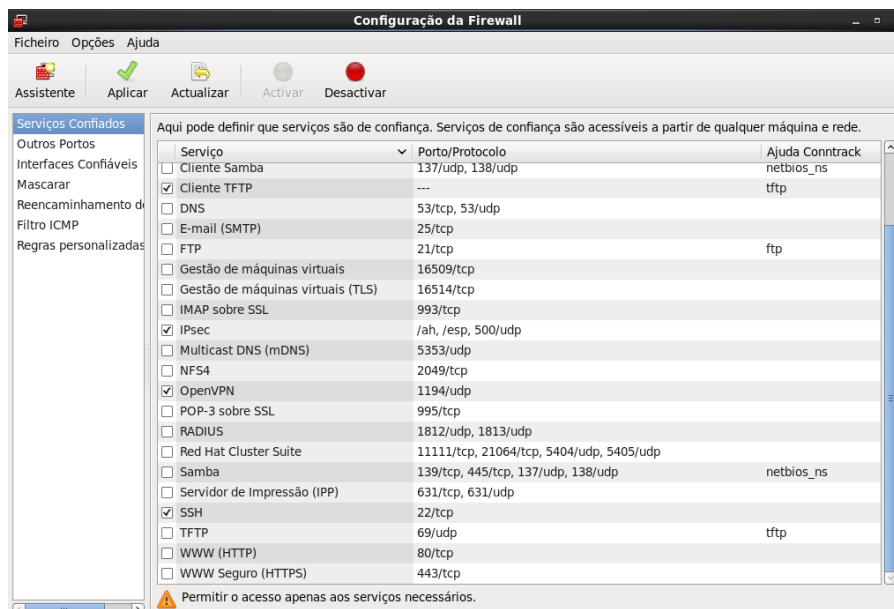


Figura 30: Resultado após seleccionar o serviço OpenVPN

Depois de aplicada estas regras, inserimos o comando `"sudo iptables -L -v"`, dando o seguinte output:

```

root@localhost:~# sudo 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 REJECT      icmp --  any    any     anywhere          anywhere          state RELATED,ESTABLISHED
                                icmp echo-request reject-with icmp-ho
st-prohibited
 0      0 ACCEPT      icmp --  any    any     anywhere          anywhere
 0      0 ACCEPT      all  --  lo     any     anywhere          anywhere
 0      0 ACCEPT      udp  --  any    any     anywhere          anywhere          state NEW udp dpt:openvpn
 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

```

Figura 31: Resultado do comando iptables -L -v

Como podemos ver, o serviço **OpenVPN** já é aceite, tal como era de prever.

Concluindo, com esta funcionalidade gráfica é muito mais acessível "filtrar" os pacotes que entram e não, visto que todo este programa é simples, organizado, não criando qualquer dúvida para qualquer pessoas mesmo um leigo na matéria.

4 Conclusão

Com o projeto efetuado podemos tirar algumas ilações de todo o trabalho. Devido ao conhecimento que adquirimos nas aulas bem como grandes pesquisas na Internet, o capítulo **Contextualização** em que falamos de **IPTABLES**, foi feito sem nenhum problema relevante.

Em relação ao trabalho, inicialmente não estávamos a conseguir realizar os passos do enunciado visto que tínhamos problemas que conseguimos resolver depois de muitas pesquisas na internet. Assim, para as 2 máquinas virtuais inserimos um novo *adapter* com **Internal Network**. Devido a isto conseguimos realizar então as 3 primeiras tarefas sem nenhum problema de relevante.

Já em relação à 4ª tarefa sentimos algumas dificuldades visto que não conseguimos inserir a a função de log que é suportada pelos iptables, no entanto conseguimos realizá-la mesmo não estando completa.

Concluindo, podemos constatar que adquirimos conhecimento nesta área que, devido aos acontecimentos presentes no dia-a-dia, se torna muito importante. Antes deste trabalho tínhamos conhecimentos teóricos do que fazia a firewall mas sem nunca compreender como limitava os pacotes.

Com este trabalho, adquirimos conhecimento que iremos aprofundá-lo visto que é uma área interessante para todos os elementos do grupo.

Referências

- [1] <https://pt.wikipedia.org/wiki/OpenVPN>
- [2] <https://e-tinet.com/linux/tabelas-do-iptables-firewall-linux/>