

Tópicos Especiais em Segurança da Informação

TP10 - Filtro de pacotes (Iptables)

Arthur do Prado Labaki

19-07, 2022

GBC 235

Informações adicionais

Nem todos os exercícios estão com imagem aqui no relatório, mas todas as imagens integradas nesse relatório, quanto códigos, planilhas ou gifs de demonstração estão em meu repositório no GitHub abaixo.

[Link do meu GitHub](#)

Resolução do item 1)

Iptables é um conjunto de ferramentas e medidas que permite o controle e a definição de regras de *firewalls* e *NATs*, permitindo que a máquina tenha uma melhor filtragem de pacotes, deixando passar apenas os seguros. Ele funciona através da comparação de regras para saber se um pacote tem ou não permissão para passar, baseando-se no endereço e porta da origem, endereço e porta do destino, prioridade, e outros fatores.

Além de permitir ou não pacotes que chegam ou saem do sistema, ele também pode ser usado para modificar e monitorar o tráfego da rede, fazer *NAT* (*masquerading*, *source nat*, *destination nat*), redirecionamento de pacotes, marcação de pacotes, modificar a prioridade de pacotes que chegam ou saem do seu sistema, contagem de bytes, dividir tráfego entre máquinas, criar proteções *anti-spoofing*, contra *syn flood*, *DoS*, entre outros.

Resolução do item 2)

Ainda em Iptables, os Chains são locais onde as regras do *firewall* definidas pelo usuário são armazenadas para operação do *firewall*. Existem dois tipos de chains: os embutidos, que já vem predefinidos com o Iptable e os criados pelo usuário. Em relação aos chains embutidos, temos:

- *INPUT* - Nesse *chain* somente os pacotes destinados ao IP do computador em questão são avaliados pelas regras;

- *FORWARD* - Aqui somente os pacotes repassados pela máquina são avaliados, ou seja, pacotes que não provém dela e nem são destinados a ela;
- *OUTPUT* - Os pacotes avaliados dentro desta regra se limitam aos processos locais do computador.

Resolução do item 3)

Foi configurado a nova máquina e foi verificado que existe conexão entre elas, por meio dos comandos ping e ssh.

```

Arquivo  Máquina  Visualizar  Entrada  Dispositivos  Ajuda

inet 127.0.0.1 netmask 255.0.0.0
inet6 ::1 prefixlen 128 scopeid 0x10<host>
loop txqueuelen 1000 (Local Loopback)
RX packets 84 bytes 6368 (6.3 KB)
RX errors 0 dropped 0 overruns 0 frame 0
TX packets 84 bytes 6368 (6.3 KB)
TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

osboxes@osboxes:~$ ssh seed@10.0.2.6
The authenticity of host '10.0.2.6 (10.0.2.6)' can't be established.
ED25519 key fingerprint is SHA256:+273y+861FA+hDukFmEpDvUTn7QfM/3CEyPArE3Av8.
This key is not known by any other names
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '10.0.2.6' (ED25519) to the list of known hosts.
seed@10.0.2.6's password:
Welcome to Ubuntu 20.04.1 LTS (GNU/Linux 5.4.0-54-generic x86_64)

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

379 updates can be installed immediately.
379 of these updates are security updates.
To see these additional updates run: apt list --upgradable

The list of available updates is more than a week old.
To check for new updates run: sudo apt update
Your Hardware Enablement Stack (HWE) is supported until April 2025.
Last login: Wed Jun 22 11:05:25 2022 from 10.0.2.15
[07/22/22]seed@VM:~$ ls
Desktop Documents Downloads GNS3 Music Pictures Public snap Templates Videos
[07/22/22]seed@VM:~$ cd Desktop/
[07/22/22]seed@VM:~/Desktop$ ls
GNS3 IOS L4 Lab1 teste
[07/22/22]seed@VM:~/Desktop$

enp0s3: flags=4163<UP,BR
inet 172.17.0.1
ether 02:42:f8:88
RX packets 0 byt
RX errors 0 drop
TX packets 0 byt
TX errors 0 drop

lo: flags=73<UP,LOOPBACK,
inet 127.0.0.1

```

Figura 1: Demonstrando o acesso entre as duas máquinas

Resolução do item 4)

O comando *"iptables -L"* mostra as políticas atuais do iptables na máquina. Nesse caso, não existe nenhuma política nas tabelas, então esta aceitando todos os pacotes.

```
osboxes@osboxes:~$ sudo su
[sudo] password for osboxes:
root@osboxes:/home/osboxes# iptables -L
Chain INPUT (policy ACCEPT)
target     prot opt source                destination

Chain FORWARD (policy ACCEPT)
target     prot opt source                destination

Chain OUTPUT (policy ACCEPT)
target     prot opt source                destination
root@osboxes:/home/osboxes# _
```

Figura 2: Listando as políticas do Iptables

Resolução do item 5)

É utilizado os comandos *"apt install openssh-server"* para instalar o ssh e o *"service ssh status"* para verificar se o ssh realmente está ativo. Com ele instalado, é possível acessar-lo de uma outra máquina com sucesso.

Resolução do item 6)

Os comando *"iptables -P INPUT DROP"* e *"iptables -P OUTPUT ACCEPT"* significam que estamos alterando a politica padrão (-P) do *INPUT* para *DROP*, significando que ele ira recusar todos os pacotes recebidos e do *OUTPUT* para *ACCEPT*, aceitando todos os

```
[07/22/22]seed@VM:~$ ssh osboxes@10.0.2.7
osboxes@10.0.2.7's password:
Welcome to Ubuntu 22.04 LTS (GNU/Linux 5.15.0-25-generic x86_64)

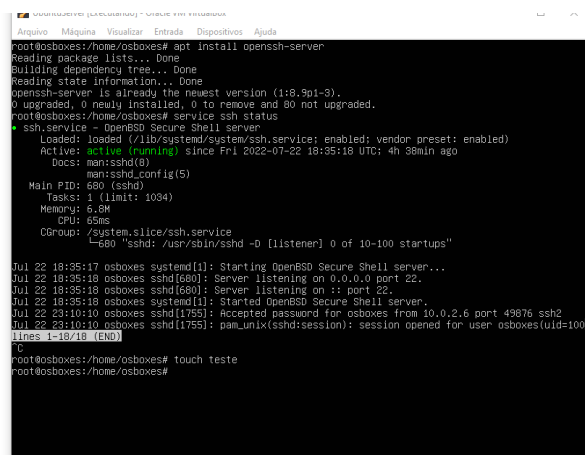
 * Documentation:  https://help.ubuntu.com
 * Management:    https://landscape.canonical.com
 * Support:       https://ubuntu.com/advantage

System information as of Fri Jul 22 11:14:56 PM UTC 2022

System load:  0.01513671875   Processes:    109
Usage of /home: 0.0% of 249.65GB   Users logged in: 1
Memory usage:  25%           IPv4 address for enp0s3: 10.0.2.7
Swap usage:    0%

85 updates can be applied immediately.
61 of these updates are standard security updates.
To see these additional updates run: apt list --upgradable

Last login: Fri Jul 22 23:10:11 2022 from 10.0.2.6
osboxes@osboxes:~$ ls
teste
```



```
root@osboxes:/home/osboxes# apt install openssh-server
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
openssh-server is already the newest version (1:8.9p1-3).
0 upgraded, 0 newly installed, 0 to remove and 80 not upgraded.
root@osboxes:/home/osboxes# service ssh status
ssh.service - OpenSSH Secure Shell server
Loaded: loaded (/lib/systemd/system/ssh.service; enabled; vendor preset: enabled)
Active: active (running) since Fri 2022-07-22 18:35:10 UTC; 4h 30min ago
Docs: man:sshd(8)
      man:sshd_config(5)
Main PID: 680 (sshd)
Tasks: 1 (limit: 1034)
Memory: 6.8M
CPU: 65ms
CGroup: /system.slice/ssh.service
        └─60 "sshd: /usr/sbin/sshd -D [listener] 0 of 10-100 startups"

Jul 22 18:35:17 osboxes systemd[1]: Starting OpenSSH Secure Shell server...
Jul 22 18:35:18 osboxes sshd[680]: Server listening on 0.0.0.0 port 22.
Jul 22 18:35:18 osboxes sshd[680]: Server listening on :: port 22.
Jul 22 18:35:18 osboxes systemd[1]: Started OpenSSH Secure Shell server.
Jul 22 23:10:10 osboxes sshd[1755]: Accepted password for osboxes from 10.0.2.6 port 49876 ssh2
Jul 22 23:10:10 osboxes sshd[1755]: pam_unix(sshd:session): session opened for user osboxes(uid=1000)
lines 1-18/18 (END)
C
root@osboxes:/home/osboxes# touch teste
root@osboxes:/home/osboxes#
```

Figura 3: Instalando e conectando na máquina com SSH

pacotes vindos da Propriá máquina. Com isso, todos os pacotes recebidos pela maquina serão recusados, mas os pacotes que saem dela serão aceitos.

É possível pensar em casos que essa política seja utilizável, em que uma máquina consegue somente enviar pacotes as outras. Mesmo que essa política inicial de controle de pacotes seja muito bruta, ela ainda é aceitável.

```
root@osboxes:/home/osboxes# iptables -F INPUT DROP
root@osboxes:/home/osboxes# iptables -F OUTPUT ACCEPT
root@osboxes:/home/osboxes# iptables -L
Chain INPUT (policy DROP)
target     prot opt source                destination

Chain FORWARD (policy ACCEPT)
target     prot opt source                destination

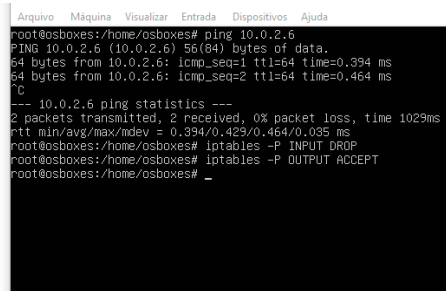
Chain OUTPUT (policy ACCEPT)
target     prot opt source                destination
root@osboxes:/home/osboxes# _
```

Figura 4: Modificando as politicas padrões

Resolução do item 7)

Utilizando o comando ping, é possível verificar que a conexão da segunda máquina com a máquina modificada realmente não ocorre após utilizar o *DROP* no *INPUT*.

```
[07/22/22]seed@VM:~$ ping 10.0.2.7
PING 10.0.2.7 (10.0.2.7) 56(84) bytes of data.
64 bytes from 10.0.2.7: icmp_seq=1 ttl=64 time=0.329 ms
64 bytes from 10.0.2.7: icmp_seq=2 ttl=64 time=0.412 ms
^C
--- 10.0.2.7 ping statistics ---
2 packets transmitted, 2 received, 0% packet loss, time 1027ms
rtt min/avg/max/mdev = 0.329/0.370/0.412/0.041 ms
[07/22/22]seed@VM:~$ ping 10.0.2.7
PING 10.0.2.7 (10.0.2.7) 56(84) bytes of data.
^C
--- 10.0.2.7 ping statistics ---
10 packets transmitted, 0 received, 100% packet loss, time 9194ms
```



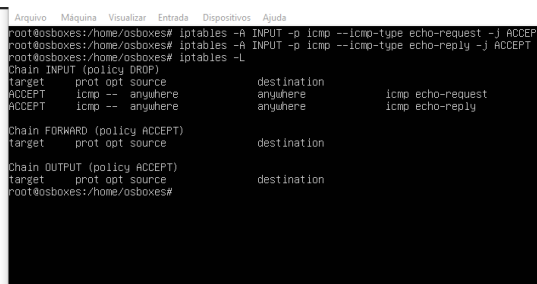
```
root@osboxes:/home/osboxes# ping 10.0.2.6
PING 10.0.2.6 (10.0.2.6) 56(84) bytes of data.
64 bytes from 10.0.2.6: icmp_seq=1 ttl=64 time=0.394 ms
64 bytes from 10.0.2.6: icmp_seq=2 ttl=64 time=0.464 ms
^C
--- 10.0.2.6 ping statistics ---
2 packets transmitted, 2 received, 0% packet loss, time 1029ms
rtt min/avg/max/mdev = 0.394/0.429/0.464/0.035 ms
root@osboxes:/home/osboxes# iptables -P INPUT DROP
root@osboxes:/home/osboxes# iptables -P OUTPUT ACCEPT
root@osboxes:/home/osboxes#
```

Figura 5: Tentando utilizar o comando Ping

Resolução do item 8)

Utilizando os comandos *iptables -A INPUT -p icmp --icmp-type echo-request -j ACCEPT* e *iptables -A INPUT -p icmp --icmp-type echo-reply -j ACCEPT* permitimos os pacotes *icmp* do tipo *echo request* e *echo reply*, que são os utilizados pelo comando ping. Agora é possível a segunda máquina conversar com essa alterada através do comando ping.

```
[07/22/22]seed@VM:~$ ping 10.0.2.7
PING 10.0.2.7 (10.0.2.7) 56(84) bytes of data.
64 bytes from 10.0.2.7: icmp_seq=1 ttl=64 time=0.355 ms
64 bytes from 10.0.2.7: icmp_seq=2 ttl=64 time=0.275 ms
64 bytes from 10.0.2.7: icmp_seq=3 ttl=64 time=0.265 ms
64 bytes from 10.0.2.7: icmp_seq=4 ttl=64 time=0.350 ms
64 bytes from 10.0.2.7: icmp_seq=5 ttl=64 time=0.334 ms
64 bytes from 10.0.2.7: icmp_seq=6 ttl=64 time=0.278 ms
64 bytes from 10.0.2.7: icmp_seq=7 ttl=64 time=0.447 ms
64 bytes from 10.0.2.7: icmp_seq=8 ttl=64 time=0.255 ms
^C
--- 10.0.2.7 ping statistics ---
8 packets transmitted, 8 received, 0% packet loss, time 7155ms
rtt min/avg/max/mdev = 0.255/0.319/0.447/0.060 ms
```



```
root@osboxes:/home/osboxes# iptables -A INPUT -p icmp --icmp-type echo-request -j ACCEPT
root@osboxes:/home/osboxes# iptables -A INPUT -p icmp --icmp-type echo-reply -j ACCEPT
root@osboxes:/home/osboxes# iptables -L
Chain INPUT (policy DROP)
target prot opt source destination
ACCEPT icmp -- anywhere anywhere icmp echo-request
ACCEPT icmp -- anywhere anywhere icmp echo-reply

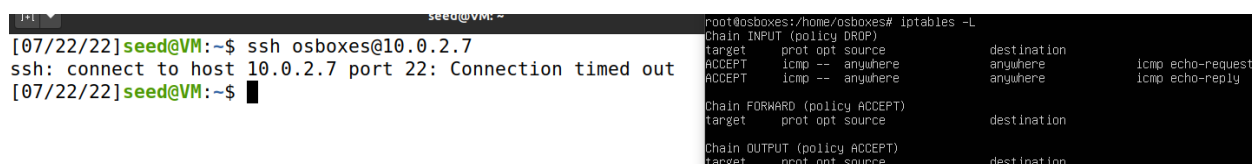
Chain FORWARD (policy ACCEPT)
target prot opt source destination

Chain OUTPUT (policy ACCEPT)
target prot opt source destination
root@osboxes:/home/osboxes#
```

Figura 6: Utilizando o comando Ping com sucesso

Resolução do item 9)

Tentando acessar a máquina principal utilizando o comando `ssh` da segunda máquina, é possível verificar que o acesso não é permitido, demorando um tempo até ocorrer um *time out*. Isso aconteceu pois o *INPUT* da máquina principal está como *DROP*, com exceção dos pacotes *icmp* adicionados no exercício anterior. Como o `ssh` não utiliza esses pacotes, o acesso não será disponível.



```

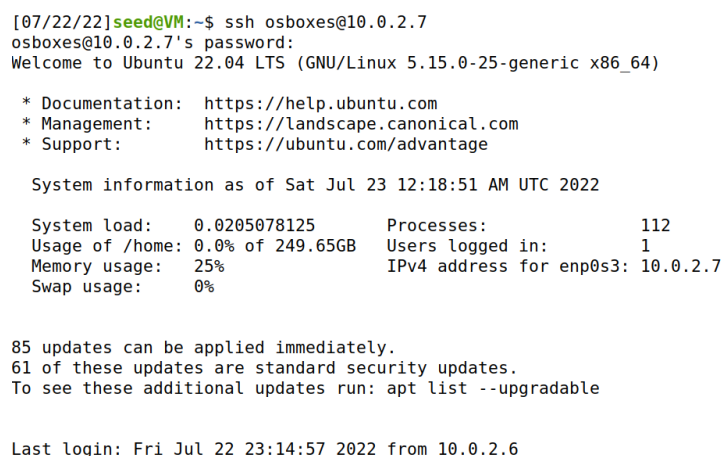
[07/22/22]seed@VM:~$ ssh osboxes@10.0.2.7
ssh: connect to host 10.0.2.7 port 22: Connection timed out
[07/22/22]seed@VM:~$

```

Figura 7: Tentando acesso do SSH com a máquina principal

Resolução do item 10)

Com o comando `"iptables -A INPUT -p tcp --dport 22 -j ACCEPT"`, agora a transmissão de pacotes de outras máquinas via `ssh` (porta tcp 22) está permitida, sendo possível estabelecer essa conexão entre elas.



```

[07/22/22]seed@VM:~$ ssh osboxes@10.0.2.7
osboxes@10.0.2.7's password:
Welcome to Ubuntu 22.04 LTS (GNU/Linux 5.15.0-25-generic x86_64)

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

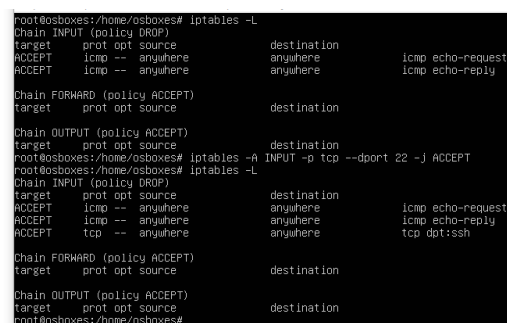
System information as of Sat Jul 23 12:18:51 AM UTC 2022

System load:  0.0205078125   Processes:    112
Usage of /home: 0.0% of 249.65GB   Users logged in: 1
Memory usage:   25%          IPv4 address for enp0s3: 10.0.2.7
Swap usage:     0%

85 updates can be applied immediately.
61 of these updates are standard security updates.
To see these additional updates run: apt list --upgradable

Last login: Fri Jul 22 23:14:57 2022 from 10.0.2.6

```



```

root@osboxes:/home/osboxes# iptables -L
Chain INPUT (policy DROP)
target     prot opt source                destination           icmp echo-request
ACCEPT     icmp -- anywhere             anywhere              icmp echo-reply

Chain FORWARD (policy ACCEPT)
target     prot opt source                destination

Chain OUTPUT (policy ACCEPT)
target     prot opt source                destination

root@osboxes:/home/osboxes# iptables -A INPUT -p tcp --dport 22 -j ACCEPT
root@osboxes:/home/osboxes# iptables -L
Chain INPUT (policy DROP)
target     prot opt source                destination           icmp echo-request
ACCEPT     icmp -- anywhere             anywhere              icmp echo-reply
ACCEPT     tcp  -- anywhere             anywhere              tcp dpt:ssh

Chain FORWARD (policy ACCEPT)
target     prot opt source                destination

Chain OUTPUT (policy ACCEPT)
target     prot opt source                destination

root@osboxes:/home/osboxes#

```

Figura 8: Estabelecendo conexão SSH com a máquina principal

Resolução do item 11)

As regras do iptables funcionam de maneira ordenada, em que as regras são verificadas de maneira ascendente, ou seja, o pacote será analisado de acordo com a regra 1, depois com a 2, até terminar todas as regras ou encontrar a primeira que se encaixa com o pacote analisado. Sendo assim, mesmo com uma nova regra bloqueando o ssh, a regra antecessora permite ele, logo o acesso será permitido na máquina.

```
[07/22/22]seed@VM:~$ ssh osboxes@10.0.2.7
osboxes@10.0.2.7's password:
Welcome to Ubuntu 22.04 LTS (GNU/Linux 5.15.0-25-generic x86_64)

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

System information as of Sat Jul 23 12:25:55 AM UTC 2022

System load:  0.0           Processes:      112
Usage of /home: 0.0% of 249.65GB   Users logged in: 1
Memory usage:  25%          IPv4 address for enp0s3: 10.0.2.7
Swap usage:    0%

85 updates can be applied immediately.
61 of these updates are standard security updates.
To see these additional updates run: apt list --upgradable

Failed to connect to https://changelogs.ubuntu.com/meta-release-lts. Check your
Internet connection or proxy settings

Last login: Sat Jul 23 00:18:52 2022 from 10.0.2.6
```

```
root@osboxes:/home/osboxes# iptables -L --line-numbers
Chain INPUT (policy DROP)
num target prot opt source destination
1 ACCEPT icmp -- anywhere anywhere icmp echo-request
2 ACCEPT icmp -- anywhere anywhere icmp echo-reply
3 ACCEPT tcp -- anywhere anywhere tcp dpt:ssh

Chain FORWARD (policy ACCEPT)
num target prot opt source destination

Chain OUTPUT (policy ACCEPT)
num target prot opt source destination
root@osboxes:/home/osboxes# iptables -A INPUT -p tcp --dport 22 -J REJECT
root@osboxes:/home/osboxes# iptables -L --line-numbers
Chain INPUT (policy DROP)
num target prot opt source destination
1 ACCEPT icmp -- anywhere anywhere icmp echo-request
2 ACCEPT icmp -- anywhere anywhere icmp echo-reply
3 ACCEPT tcp -- anywhere anywhere tcp dpt:ssh
4 REJECT tcp -- anywhere anywhere tcp dpt:ssh reject-with icmp-port-unreachable

Chain FORWARD (policy ACCEPT)
num target prot opt source destination

Chain OUTPUT (policy ACCEPT)
num target prot opt source destination
root@osboxes:/home/osboxes#
```

Figura 9: Demonstrando o acesso do SSH mesmo com seu REJECT

Resolução do item 12)

Agora, como não existe uma regra de ssh anterior à que realiza o *REJECT*, a conexão ssh será rejeitada pela máquina principal. Também é possível notar uma diferença entre *DROP* e *REJECT*, em que o *DROP* barra um pacote silenciosamente, em que nenhuma resposta ou mensagem de erro é devolvida ao remetente. Já o *REJECT* barra um pacote e devolve um erro ao remetente informando que o pacote foi barrado. É possível notar no exercício, em que o ssh instantaneamente foi devolvido com um erro de conexão recusada, já com o *DROP* era somente cancelado pelo *time out*.


```

seed@VM: ~
22/22]seed@VM:~$ ssh osboxes@10.0.2.7
connect to host 10.0.2.7 port 22: Connection refused
22/22]seed@VM:~$

root@osboxes:/home/osboxes# iptables -D INPUT 4
root@osboxes:/home/osboxes# iptables -L --line-numbers
Chain INPUT (policy DROP)
num target prot opt source destination
1 ACCEPT icmp -- anywhere anywhere icmp echo-request
2 ACCEPT icmp -- anywhere anywhere icmp echo-reply
3 REJECT tcp -- anywhere anywhere tcp dpt:ssh reject-with icmp-port-unreachable
Chain FORWARD (policy ACCEPT)
num target prot opt source destination
Chain OUTPUT (policy ACCEPT)
num target prot opt source destination
root@osboxes:/home/osboxes#

```

Figura 10: Nova resposta do comando SSH

Resolução do item 13)

Para criar regras específicas a um determinado endereço IP, é necessário utilizar os argumentos *-d endereçoIP*". Como o endereço da minha máquina secundária é *10.0.2.6*, temos:

- *iptables -A INPUT -d 10.0.2.6 -p icmp --icmp-type echo-request -j ACCEPT*
- *iptables -A INPUT -d 10.0.2.6 -p icmp --icmp-type echo-reply -j ACCEPT*
- *iptables -A INPUT -d 10.0.2.6 -p tcp --dport 22 -j ACCEPT*
- *iptables -A INPUT -d 10.0.2.6 -p tcp --dport 22 -j REJECT*

```

root@osboxes:/home/osboxes# iptables -F
root@osboxes:/home/osboxes# iptables -A INPUT -d 10.0.2.6 -p icmp --icmp-type echo-request -j ACCEPT
root@osboxes:/home/osboxes# iptables -A INPUT -d 10.0.2.6 -p icmp --icmp-type echo-reply -j ACCEPT
root@osboxes:/home/osboxes# iptables -A INPUT -d 10.0.2.6 -p tcp --dport 22 -j ACCEPT
root@osboxes:/home/osboxes# iptables -A INPUT -d 10.0.2.6 -p tcp --dport 22 -j REJECT
root@osboxes:/home/osboxes# iptables -L
Chain INPUT (policy DROP)
target prot opt source destination
ACCEPT icmp -- anywhere 10.0.2.6 icmp echo-request
ACCEPT icmp -- anywhere 10.0.2.6 icmp echo-reply
ACCEPT tcp -- anywhere 10.0.2.6 tcp dpt:ssh
REJECT tcp -- anywhere 10.0.2.6 tcp dpt:ssh reject-with icmp-port-unre
achable

Chain FORWARD (policy ACCEPT)
target prot opt source destination

Chain OUTPUT (policy ACCEPT)
target prot opt source destination
root@osboxes:/home/osboxes#

```

Figura 11: Refazendo comandos em um IP específico

Resolução do item 14)

Para configurar esse servidor Web, primeiramente deve-se bloquear todas as políticas padrões, impedindo qualquer passagem de pacotes. Após isso, como o servidor é Web, será necessário abrir as portas que hospedam a aplicação para todos os *hosts*, abrindo as portas 80 para HTTP e 443 para HTTPS.

Como essas portas sustentam uma aplicação Web, ela deve ser do tipo cliente-servidor, sendo necessário o *GET* e *RESPONSE*. Dado isso, será necessário permitir essa portas nas *chains INPUT* e *OUTPUT*, pois esse tipo de requisição necessita de ambas as partes compartilharem pacotes. Por fim, também será aberto a porta 22 de ssh para um IP específico, sendo ele o *10.0.2.7*. Como o acesso do ssh será somente feito desse IP no servidor, somente será necessário permiti-lo na *chain INPUT*, permitindo esse acesso.

Com isso, temos os comandos:

- iptables -P INPUT DROP
- iptables -P OUTPUT DROP
- iptables -P FORWARD DROP
- iptables -A INPUT -p tcp --dport 80 -j ACCEPT
- iptables -A OUTPUT -p tcp --dport 80 -j ACCEPT
- iptables -A INPUT -p tcp --dport 443 -j ACCEPT
- iptables -A OUTPUT -p tcp --dport 443 -j ACCEPT
- iptables -A INPUT -d 10.0.2.7 -p tcp --dport 22 -j ACCEPT

```

root@osboxes:/home/osboxes# iptables -P INPUT DROP
root@osboxes:/home/osboxes# iptables -P OUTPUT DROP
root@osboxes:/home/osboxes# iptables -P FORWARD DROP
root@osboxes:/home/osboxes# iptables -A INPUT -p tcp --dport 80 -j ACCEPT
root@osboxes:/home/osboxes# iptables -A OUTPUT -p tcp --dport 80 -j ACCEPT
root@osboxes:/home/osboxes# iptables -A INPUT -p tcp --dport 443 -j ACCEPT
root@osboxes:/home/osboxes# iptables -A OUTPUT -p tcp --dport 443 -j ACCEPT
root@osboxes:/home/osboxes# iptables -A INPUT -d 10.0.2.7 -p tcp --dport 22 -j ACCEPT
root@osboxes:/home/osboxes# iptables -L
Chain INPUT (policy DROP)
target      prot opt source                destination           tcp dpt:http
ACCEPT      tcp  --  anywhere               anywhere              tcp dpt:https
ACCEPT      tcp  --  anywhere               anywhere              tcp dpt:ssh
Chain FORWARD (policy DROP)
target      prot opt source                destination
Chain OUTPUT (policy DROP)
target      prot opt source                destination           tcp dpt:http
ACCEPT      tcp  --  anywhere               anywhere              tcp dpt:https

```

Figura 12: Tabela de regras do Iptables do servidor Web