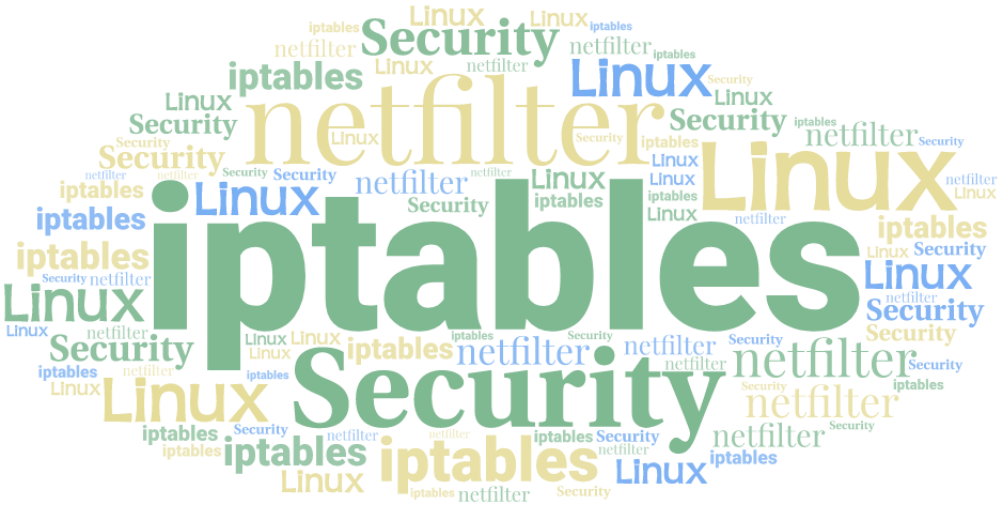


IPTABLES FÁCIL E FUNCIONAL:

GUIA PRÁTICO DO MINI-CURSO DE FIREWALL COM NETFILTER/IPTABLES

LAVRAS – MG

ABR/2019

DIEGO NATIVIDADE

IPTABLES FÁCIL E FUNCIONAL:
GUIA PRÁTICO DO MINI-CURSO DE FIREWALL COM NETFILTER/IPTABLES

Material didático utilizado como guia prático do mini-curso de firewall com netfilter/iptables. Este material é gratuito, não podendo ser vendido e seu conteúdo não pode ser alterado. Quaisquer sugestões, como erros ou críticas, devem ser encaminhadas ao autor.

Contato: natividade@bol.com.br

Versão 1.1

LAVRAS – MG

ABR/2019

APOIO:


CONNECTIVA
REDES DE COMPUTADORES
(35)3822-4271


quantum tecnologia da informação

AGRADECIMENTOS

Agradeço primeiramente a CONNECTIVA REDES DE COMPUTADORES, empresa que fui fundador e que hoje presto consultorias regularmente. Agradeço também a Universidade Federal de Lavras e professores pelos ensinamentos. Em particular, agradeço a minha tia Rita pela ajuda no início de minha vida acadêmica e todo apoio e suporte que tem me dado durante todos esses anos. Agradeço ainda a CAPES, pela bolsa e apoio no mestrado.
Por fim, agradeço ao Universo, que sempre conspira ao nosso favor...

Meus sinceros agradecimentos.

"A desconfiança é a mãe da segurança."

(Madeleine Scudéry)

"Aquele que se eleva nas pontas dos pés não está seguro."

(Lao-Tsé)

"A segurança não é um produto, ela é um processo."

(Bruce Schneier)

"O fator humano é o elo mais fraco da segurança."

(Kevin Mitnick)

"Quis custodiet ipsos custodes?"

(Romano Juvenal)

LISTA DE FIGURAS

LISTA DE TABELAS

LISTA DE SIGLAS

DHCP *Dynamic Host Configuration Protocol*

DNAT *Destination NAT*

DNS *Domain Name System*

HTTP *Hypertext Transfer Protocol*

HTTPS *Hypertext Transfer Protocol Secure*

ICMP *Internet Control Message Protocol*

IDS *Intrusion Detection System*

IP *Internet Protocol*

IPS *Intrusion Prevention System*

MAC address *Medium Access Control address*

MAC *Mandatory Access Control*

NAT *Network Address Translation*

NGFW *Next Generation Firewall*

QoS *Quality of Service*

SNAT *Source NAT*

SPI *Stateful Packet Inspection*

SSH *Secure Shell*

TCP *Transmission Control Protocol*

UTM *Unified Threat Management*

UDP *User Datagram Protocol*

ÍNDICE

1 INTRODUÇÃO

Manter uma rede de computadores 100% SEGURA, é um ideal, inatingível e utópico. *Firewall*, *proxy*, *Intrusion Detection System* (IDS), *Intrusion Prevention System* (IPS) e antivírus, são alguns dos sistemas que compõem a segurança de uma rede de computadores.

A implantação de um *firewall* em uma rede é o básico que se pode fazer para conseguir um mínimo de segurança. Sistemas operacionais GNU/Linux, desde o *Kernel* 2.4, vêm por padrão com um sistema de filtragem de pacotes denominado `netfilter`. A ferramenta utilizada para acessar suas configurações é o `iptables`.

Neste mini-curso, serão apresentados os conceitos básicos de um *firewall*, o funcionamento do `netfilter`/`iptables` e um caso de uso, como prática.

1.1 Objetivo

O objetivo deste mini-curso é dar ao aluno a capacidade de entender o que é um *firewall*, o funcionamento, o fluxo e as regras do `iptables`, além de capacitá-lo a montar suas próprias políticas e dar manutenção em sistemas de *firewall* já em produção.

1.2 Organização do trabalho

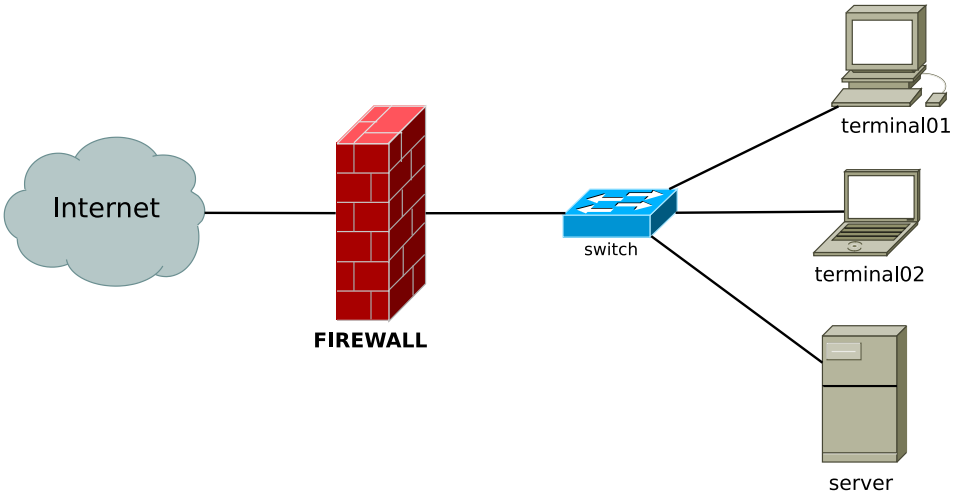
Este trabalho está organizado da seguinte maneira: o Capítulo 2, traz um *background* sobre *firewalls* e seus tipos. No Capítulo 3 é apresentado o `netfilter` e `iptables`, bem como suas principais opções. No Capítulo 4 são apresentados vários exemplos práticos de uso, de forma detalhada e em um sequência lógica para ser implementado em um *firewall* de verdade. No Capítulo 5, uma breve conclusão. Por fim, as referências utilizadas neste trabalho e três apêndices com o *script* de *firewall* utilizado em toda prática deste mini-curso, um exemplo de *firewall* para uso em computador pessoal e um *script* de *firewall* completo e funcional, que pode ser adaptado pelo aluno para uso no dia a dia.

2 FIREWALL

O *firewall* é uma das primeiras linhas de defesa de uma rede ou *host* (CHEN et al., 2016). Consiste basicamente em um filtro de pacotes colocado em um ponto de entrada de uma rede, no qual, cada pacote que passa por ele é verificado a fim de determinar se o pacote será aceito ou não (Acharya; Joshi; Gouda, 2010). Um *firewall* permite que os administradores do sistema criem controles de acesso entre uma rede confiável e uma não confiável (Murthy et al., 1998).

A Figura 2.1 ilustra o funcionamento e posicionamento de um *firewall* na rede. O mesmo está localizado entre a Internet e a rede local, liberando ou bloqueando os acessos, tanto de entrada, quanto de saída.

Figure 2.1 – Exemplo clássico de um *firewall*



Fonte: do autor (2019)

Os *firewalls* são normalmente classificados nas seguintes categorias:

- **Packet filtering firewall**: os pacotes que chegam ou saem são liberados ou bloqueados de acordo com as retrições de *Internet Protocol* (IP), porta, protocolo, etc., sem abrir e inspecionar o conteúdo do pacote;
- **Circuit-level firewall**: opera na camada de sessão, os pacotes *Transmission Control Protocol* (TCP) são abertos e o *handshake* é inspecionado (Compuquip Cyber Security, 2019), (Comodo Antivirus, 2019);
- ***Stateful Packet Inspection* (SPI)**: combinação do *packet filtering* e *circuit-level*, onde os cabeçalhos e o estado dos pacotes são analisados (Compuquip Cyber Security, 2019);

- **Proxy firewall**: trabalham com a camada de aplicação, provendo segurança diretamente para serviços suportados (Comodo Antivirus, 2019);

- ***Unified Threat Management* (UTM)** e ***Next Generation Firewall* (NGFW)**: UTM é uma solução centralizada de *firewall*, *antimaware*, IDS/IPS, entre outras ferramentas. NGFW trata-se de uma solução UTM aprimorada, garantindo mais performance e escalabilidade. Alguns autores classificam ambos como uma só solução.

O `netfilter` é um *firewall* que pode trabalhar desde *stateless* à *statefull*, abrangendo as funcionalidades das três primeiras categorias descritas. A seguir, ver-se-á mais detalhes do `netfilter` e sua principal ferramenta de manipulação, e objeto de estudo deste trabalho, o `iptables`.

3 NETFILTER

`netfilter` é um *framework* composto por vários módulos que permitem fazer *Network Address Translation* (NAT), mascaramento, filtragem de pacotes, marcação entre outras manipulações de pacotes de rede. Ele é utilizado em sistemas GNU/Linux desde o *Kernel* 2.4 (netfilter, 2019b), e funciona da seguinte maneira:

- cada pacote que chega no *Kernel* é verificado se ele "casa"dá (*match*) com alguma das regras criadas;
- as regras são testadas linha a linha, da primeira até a última;
- quando um pacote dá *match*, ele processa a regra e não continua a verificação das demais regras, exceto em situações especiais, como quando utilizado o alvo `LOG` ou `RETURN`, por exemplo;
- caso algum pacote não satisfaça aos critérios de nenhuma das regras, o mesmo será tratado pela "política padrão"(este assunto erá visto mais adiante).

O software utilizado para a manipulação do `netfilter` é o `iptables` (netfilter, 2019b), que será visto em detalhes a seguir.

3.1 iptables

O `iptables` é a principal ferramenta utilizada para a manipulação do `netfilter`. Muitas vezes os dois termos são confundidos. Neste curso, serão utilizados os termos `netfilter` e `iptables` arbitrariamente. A sintaxe padrão para uso do `iptables` é a seguinte:

 # iptables [-t tabela] [opção] [CHAIN] [regra] -j [ALVO/AÇÃO]

A seguir, as tabelas e *chains built-in* do `iptables`.

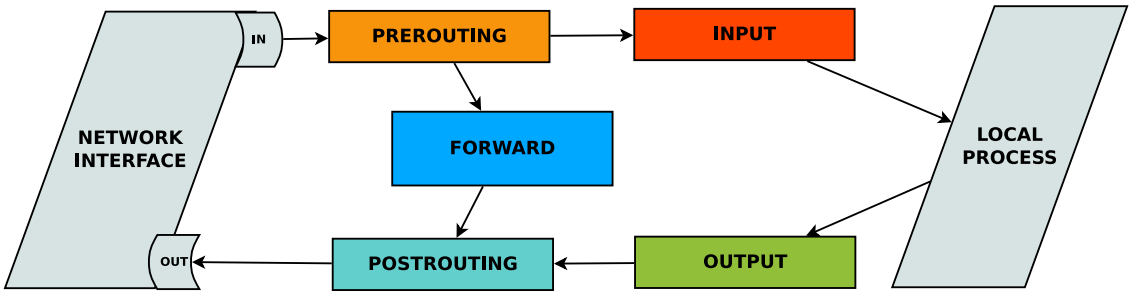
3.1.1 *Chains*

O `iptables` faz uso de *chains* (cadeias), que armazenam as regras e indicam o sentido do fluxo de dados. Existem cinco *chains built-in* (*chains* que já vêm no *Kernel*), mas também é possível a criação de *chains* customizadas, para melhor organização das regras. A seguir as cinco *chains built-in* disponíveis:

- **INPUT**: manipula pacotes que são originados no *Kernel* do *firewall* (na máquina local);
- **OUTPUT**: trata pacotes que são destinados ao *firewall* (destinados a máquina local);
- **FORWARD** lida com pacotes que atravessam o *Kernel* do *firewall* (pacotes que saem de uma maquina para outra na rede, passando pelo *firewall*);
- **PREROUTING**: manipula pacotes que acabaram de chegar ao *Kernel*, antes de qualquer roteamento;
- **POSTROUTING**: manipula pacotes que estão prestes a sair do *Kernel*, após o roteamento.

A Figura 3.1 apresenta as *chains built-in* do `iptables`, bem como o sentido de fluxo dos pacotes que chegam no *Kernel*. As *chains* `PREROUTING` e `POSTROUTING`, são sempre executadas, respectivamente, antes e depois de qualquer outra *(*chain) ou processamento. Todo tráfego destinado ao próprio *firewall* (vindo da rede ou da Internet) é manipulado pela *chain* `INPUT`. Todo tráfego que parte do *firewall*, em direção a um *host* na rede ou a Internet é tratado pela *chain* `OUTPUT`. Já o trafego que sai da rede para Internet, da Internet para rede, ou entre as sub-redes (ou seja, todo o tráfego que **passa** pelo firewall, mas não o tem como origem ou destino) são manipulados pela *chain* `FORWARD`.

Figure 3.1 – *Chains* do `iptables`



Fonte: do autor (2019)

3.1.2 Tabelas

As *chains*, vistas nas sessão anterior, estão contidas em tabelas do iptables. O `iptables` possui atualmente, cinco tabelas: `filter`, `nat`, `mangle`, `raw` e `security`. Destas, as duas últimas são relativamente novas, sendo a tabela security usada somente em ambientes que utilizem o módulo `SELinux`. A seguir, uma breve descrição de cada uma dessas tabelas, retiradas de netfilter (2019a), DigitalOcean (2019) e Boolean World (2019)

- **filter**: é a tabela padrão do `iptables`, quando nenhuma tabela é especificada; é responsável pela filtragem de pacotes, como aceitar ou descartar um pacote, por exemplo;
- **nat**: é consultada toda vez que que criada uma nova conexão; usada para fazer mascaramento e redirecionamentos de IPs e portas;
- **mangle**: é utilizada para marcação de pacotes, para que estas marcações possam ser usadas posteriormente em regras de *Quality of Service* (QoS) ou roteamento, por exemplo;
- **raw**: esta tabela fica antes da *connection tracking*, permitindo utilizar o *firewall* em *stateless*; utilizada para marcar ou mesmo bloquear um pacote, antes que ele passe por qualquer tipo processamento, reduzindo a carga do *firewall*.
- **security**: é utilizada em regras de *Mandatory Access Control* (MAC) em conjunto com módulos de segurança do GNU/Linux, como o `SELinux`; cria marcações nos pacotes ou conexões para serem manipuladas posteriormente pelo `SELinux`;

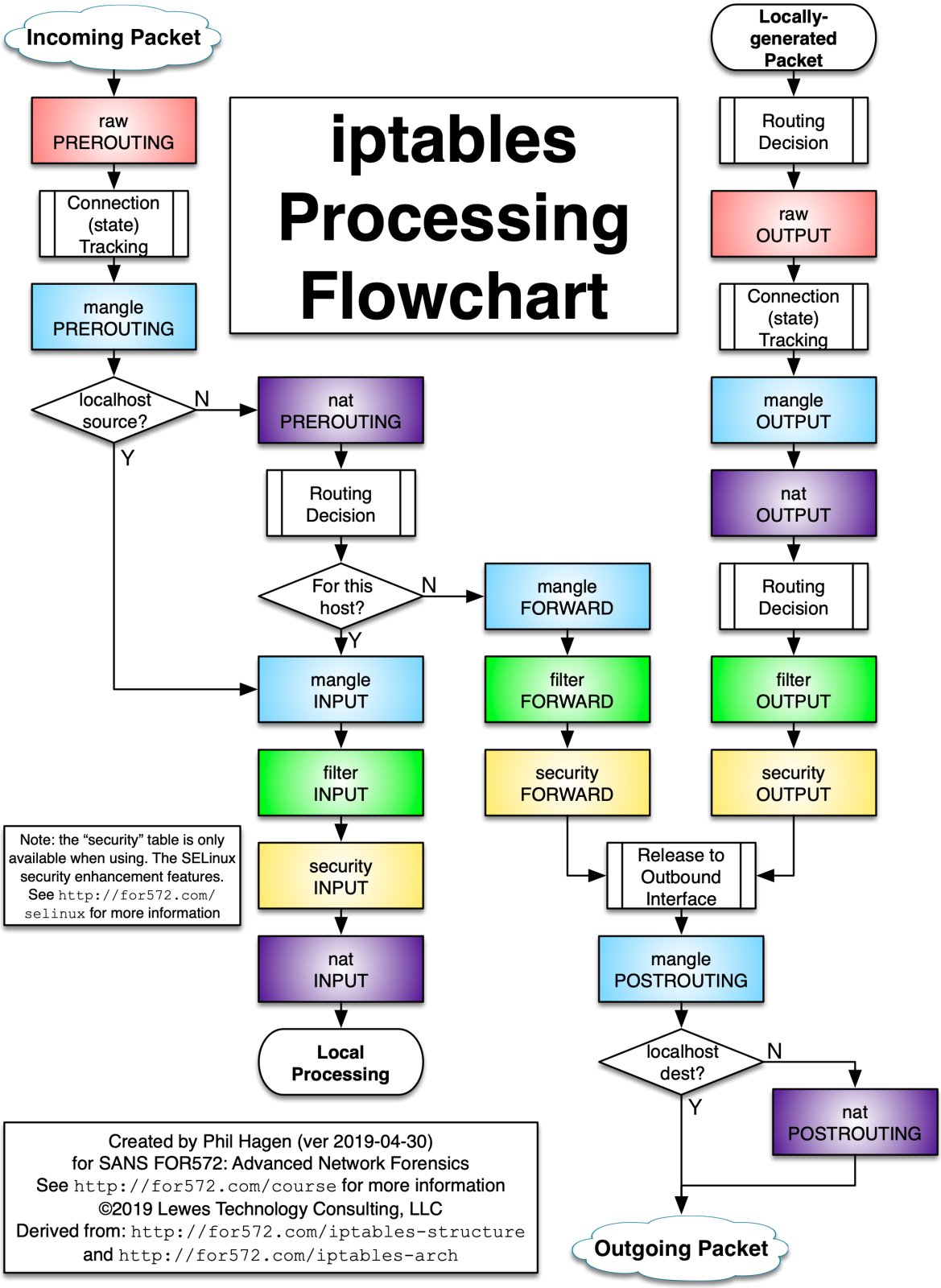
A Tabela 3.1 mostra as tabelas do `iptables` e suas *chains*. Na Figura 3.2, temos o fluxo de dados completo do `iptables`, mostrando todas as tabelas e suas *chains*.

Tabela 3.1 – iptables: tablelas e suas *chains*

Tabela	Descrição	Cadeia
`filter`	Filtragem de pacotes	INPUT, FORWARD, OUTPUT
`nat`	Mascaramento e redirecionamento de IPs/portas	PREROUTING, OUTPUT, POSTROUTING
`mangle`	Marcação de pacotes	Todas as chains built-in
`raw`	Stateless; antes do processamento dos pacotes	PREROUTING, OUTPUT
`security`	Mandatory Access Control (uso com `SELinux`)	INPUT, FORWARD, OUTPUT

Fonte: do autor (2019)

Figure 3.2 – *Datapath* completo do `iptables`



Fonte: Hagen (2019)

3.1.3 Alvos/ações

Os alvos (*target*) ou ações, tratam do que será feito com o pacote, caso o mesmo dê *match* na regra. Usando a opção `-j`, pode-se especificar uma ação diretamente, ou saltar para uma *chain* customizada. Neste mini-curso, não abordaremos *chains* customizadas, estando ete assunto fora de nosso escopo. A seguir veremos os principais alvos do `iptables`, que são: `ACCEPT`, `DROP`, `REJECT`, `LOG`, `DNAT`, `SNAT` e `MASQUERADE`.

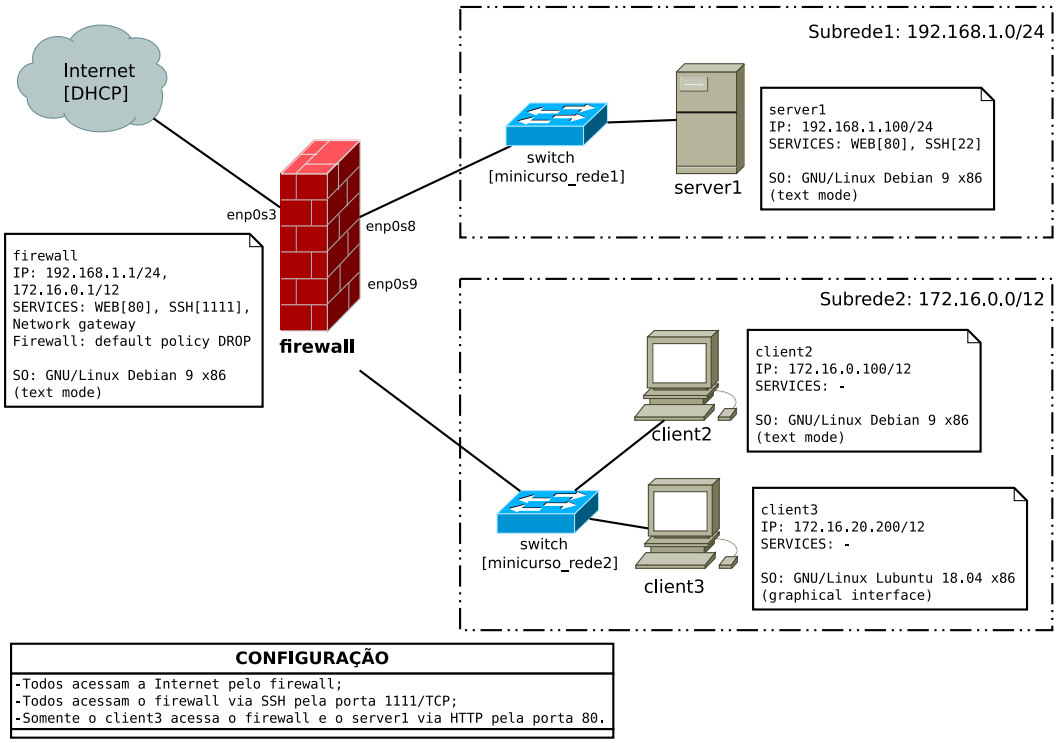
- **ACCEP**: aceita um pacote ou conexão;
- **DROP**: descarta um pacote ou conexão, sem enviar qualquer informação para o remetente;
- **REJECT** rejeita um pacote ou conexão, enviando uma confirmação para o remetente;
- **LOG**: registra um *log* do pacote ou conexão, mas continua percorrendo as próximas regras, ou seja, não termina o processamento;
- **DNAT**: altera o IP/porta de destino de um pacote, com IP/porta especificados por `-to IP:PORT` (para fazer um redirecionamento de portas, por exemplo);
- **SNAT**: altera o IP/porta de origem de um pacote, com IP/porta especificados por `-to IP:PORT` (para fazer o NAT para um IP fixo, por exemplo);
- **MASQUERADE**: caso particular de *Source NAT* (SNAT), no qual altera o IP/porta de origem de um pacote, com IP/porta do *firewall*, afim de fazer NAT quando não se tem um IP fixo.

4 CASO DE USO COM `IPTABLES`

Com o `iptables` é possível criar uma grande variedade regras, devido a grande quantidade de módulos e parâmetros que ele aceita. Agora, serão vistos vários exemplos de regras do `iptables`, começando pelas regras padrão, passando por regras de filtragem, mascaramento e por fim regras de redirecionamento de IPs e portas.

A Figura 4.1, exibe a topologia de rede utilizada em nosso teste, na qual são criadas duas sub-redes: `sub-rede1`, endereço `192.168.1.0/24`, com um servidor `server1`, configurado com o IP `192.168.1.100/24`; `sub-rede2`, endereço `172.16.0.0/12`, com duas estações de trabalho, `client2` e `client3`, com IPs `172.16.0.100/12` e `172.16.20.200/12`, respectivamente. O *firewall* é um computador com três *interfaces* de rede: `enp0s3`, recebendo IP dinamicamente (por *Dynamic Host Configuration Protocol* (DHCP)) e voltada para a Internet; `enp0s8`, com IP `192.168.1.1/24` e conectada com o *switch* da `sub-rede1`; `enp0s3`, com IP `172.16.0.1/12` conectada com o *switch* da `sub-rede2`.

Figure 4.1 – Topologia de rede dos testes



Fonte: do autor (2019)

4.1 Inicio das regras

Para os testes, recomenda-se a criação de um arquivo de *script*, contendo todas as regras que iremos utilizar em nosso *firewall* (uma regra por linha do arquivo).

A Figura 4.2 mostra o cabeçalho do arquivo que utilizaremos nos testes. Nas primeiras linhas deste arquivo (linhas 5, 7 e 9), criaremos 3 variáveis `IF_WAN`, `IF_LAN1` e `IF_LAN2`, referentes as interface de Internet, `sub-rede1` e `sub-rede2`, respectivamente. Em seguida, das linhas 14 à 19, estão os comandos que permitem limpar as regras, contadores e *chains* customizadas, a fim de iniciar um *firewall* do zero, garantindo que não exista nenhuma regra previamente carregada.

Figure 4.2 – Cabeçalho padrão para *scripts* de *firewall*

```
1  #!/bin/bash
2
3  #VARIAVEIS
4  #Interface de Internet
5  IF_WAN="enp0s3"
6  #Interface da "Rede Local 1" (192.168.1.0/24)
7  IF_LAN1="enp0s8"
8  #Interface da "Rede Local 2" (172.16.0.0/12)
9  IF_LAN2="enp0s9"
10
11 #Limpa regras (-F),
12 #exclui cadeias customizadas (-X),
13 #zera contadores (-Z)
14 for TABLE in filter nat mangle raw security
15 do
16     iptables -t $TABLE -F
17     iptables -t $TABLE -X
18     iptables -t $TABLE -Z
19 done
```

Fonte: do autor (2019)

4.2 Regra de bloqueio padrão

As regras padrões, são aquelas utilizadas para pacotes que não satisfazem a nenhuma das regras criadas. Caso não seja especificada uma regra padrão, ela será `ACCEPT` para todos os pacotes e conexões daquela *chain*. Neste exemplo vamos criar um *firewall* com a regra padrão para bloquear tudo, como na Figura 4.3.

Figure 4.3 – Regras de bloqueio padrão

```
#Politica padrao: DROP
iptables -P INPUT DROP
iptables -P FORWARD DROP
iptables -P OUTPUT DROP
```

Fonte: do autor (2019)

4.3 Permitir acesso a `lo` e conexões estabelecidas/relatadas

Com as regras padrão definidas como `DROP` para `INPUT`, `OUTPUT` e `FORWARD`, nada entra, sai ou passa pelo *firewall*. Portanto, devemos liberar as portas desejadas, para que a rede funcione corretamente. Mas antes de criar as regras de liberação de portas propriamente ditas, temos que liberar as conexões de entrada e saída da interface de `localhost` (`lo`) e permitir livre acesso as conexão já estabelecidas e relatadas, como mostra a Figura 4.4.

Figure 4.4 – Permitir acesso a `lo`, conexões estabelecidas e relatadas

```
#Libera acesso na interface de localhost (lo)
iptables -A INPUT  -i lo -j ACCEPT
iptables -A OUTPUT -o lo -j ACCEPT

#libera acesso as conexoes ja estabelecidas e relatadas
iptables -A INPUT   -m state --state ESTABLISHED,RELATED -j ACCEPT
iptables -A FORWARD -m state --state ESTABLISHED,RELATED -j ACCEPT
iptables -A OUTPUT  -m state --state NEW,ESTABLISHED,RELATED -j
   ACCEPT
```

Fonte: do autor (2019)

4.4 Permitir acesso ao *firewall* (`INPUT`)

Após realizar as configurações das sessões 4.1, 4.2 e 4.3, nosso *firewall* está pronto para receber as demais regras. **Atente-se que as configurações anteriores são de extrema importância para o correto funcionamento do *firewall* que estamos configurando neste trabalho**.

Com as regras anteriores já aplicadas, vamos liberar o acesso ao *firewall*, para pacotes *Internet Control Message Protocol* (ICMP) tipo 8 e 0, ICMP *request* e *replay* respectivamente, para uso do "ping". Daremos acesso a porta local `22/TCP` (*Secure Shell* (SSH)) para todos e a porta `53/User Datagram Protocol (UDP)` (*Domain Name System* (DNS)) somente para conexões provenientes da sub-rede `172.16.0.0/12` ou `192.168.1.0/24`. Além disso, vamos

permitir o acesso a porta `80/TCP` (*Hypertext Transfer Protocol* (HTTP)) somente para `host` com o IP `172.16.20.200`, *Medium Access Control address* (MAC address) `01:23:45:0A:BC:DE` e proveniente da interface `IF_LAN2`. Para isso, utiliza-se a *chain* `INPUT`, como mostra a Figura 4.5.

Figure 4.5 – Permitir acesso ao *firewall* (`INPUT`)

```
#Libera acesso ICMP request e reply (ping e pong)
iptables -A INPUT -p icmp --icmp-type 8 -j ACCEPT
iptables -A INPUT -p icmp --icmp-type 0 -j ACCEPT

#Libera acesso a porta 22/TCP para todos
iptables -A INPUT -p tcp --dport 22 -j ACCEPT

#Libera acesso a porta 53/UDP somente para as subredes especificadas
iptables -A INPUT -p udp --dport 53 -s 172.16.0.0/16 -j ACCEPT

#Libera acesso a porta 80/TCP somente para IP, MAC e interface
   especificados
iptables -A INPUT -p tcp --dport 80 -s 172.16.20.200 -m mac --mac-
   source 00:12:34:0A:BC:DE -i enp0s3 -j ACCEPT
```

Fonte: do autor (2019)

4.5 Compartilhar Internet para a rede local

A partir de agora, faremos as configurações do *firewall* para a rede local. Neste exemplo, o *firewall* atuará como o *gateway* das duas sub-redes. Portanto, precisamos compartilhar a Internet para a rede local, e para isso, precisamos habilitar o encaminhamento de pacotes no *Kernel* e fazer o NAT com mascaramento para as sub-redes.

Por padrão, os sistemas GNU/Linux vêm com o encaminhamento de pacotes desabilitado, para habilitar, precismos setar em "1" o arquivo `/proc/sys/net/ipv4/ip_forward`. Para isso, basta digitar a seguinte linha de comando no shell:

```
echo 1 > /proc/sys/net/ipv4/ip_forward
```

Agora, podemos fazer o NAT com mascaramento para as sub-redes, como mostra a Figura 4.6.

Figure 4.6 – Fazer NAT para a rede local

```
#Faz o NAT com mascaramento para rede local
iptables -t nat -A POSTROUTING -o $IF_WAN -j MASQUERADE
```

Fonte: do autor (2019)

Neste momento, mesmo com o NAT habilitado para a rede local, a Internet nos terminais não funcionará, pois nenhuma porta ainda foi liberada para acesso da rede local, que é o que veremos na sessão seguinte.

4.6 Permitir acessos da rede local

Faremos agora, algumas liberações da rede local para a Internet. Primeiramente, liberaremos o tráfego de pacotes ICMP `request` e *replay* para rede local (usado pelo "ping") e também o acesso às portas `80/TCP` e `443/TCP`, HTTP e *Hypertext Transfer Protocol Secure* (HTTPS) respectivamente (FIGURA 4.7).

Figure 4.7 – Permitir acesso da rede local para a Internet

```
#Libera acesso ao ICMP request e reply (ping e pong) para a Internet
iptables -A FORWARD -o $IF_WAN -p icmp --icmp-type 8 -j ACCEPT
iptables -A FORWARD -o $IF_WAN -p icmp --icmp-type 0 -j ACCEPT

#Libera acesso a porta 80/TCP e 443/TCP para a Internet
iptables -A FORWARD -o $IF_WAN -p tcp --dport 80  -j ACCEPT
iptables -A FORWARD -o $IF_WAN -p tcp --dport 80  -j ACCEPT
iptables -A FORWARD -o $IF_WAN -p tcp --dport 443 -j ACCEPT
iptables -A FORWARD -o $IF_WAN -p tcp --dport 443 -j ACCEPT
```

Fonte: do autor (2019)

Vamos configurar o acesso entre as duas sub-redes, liberando o acesso a `1111/TCP` (SSH) do `server1` para todos da rede local e a porta `80/TCP` (HTTP) somente para conexões provenientes do IP `172.16.20.200/16`, conforme mostra a Figura 4.8.

Figure 4.8 – Permitir acesso entre as sub-redes

```
#Libera acesso ICMP request e reply (ping e pong) entre as subredes
iptables -A FORWARD -i $IF_LAN1 -o $IF_LAN2 -p icmp --icmp-type 8 -j ACCEPT
iptables -A FORWARD -i $IF_LAN1 -o $IF_LAN2 -p icmp --icmp-type 0 -j ACCEPT
iptables -A FORWARD -i $IF_LAN2 -o $IF_LAN1 -p icmp --icmp-type 8 -j ACCEPT
iptables -A FORWARD -i $IF_LAN2 -o $IF_LAN1 -p icmp --icmp-type 0 -j ACCEPT

#Libera acesso a porta 1111/TCP para a rede local
iptables -A FORWARD -i $IF_LAN2 -o $IF_LAN1 -d 192.168.1.100 -p tcp --dport 1111 -j ACCEPT

#Libera acesso a porta 80/TCP para um IP da rede local
iptables -A FORWARD -i $IF_LAN2 -o $IF_LAN1 -s 172.16.20.200 -d 192.168.1.100 -p tcp --dport 80 -j ACCEPT
```

Fonte: do autor (2019)

Por fim, vamos fazer um *Destination NAT* (DNAT) (redirecionamento de portas), direcionando todo o tráfego que chegar na porta `8081/TCP` do *firewall*, para a porta `80/TCP` do `server1` (FIGURA 4.9).

Figure 4.9 – Fazer DNAT para IP da rede local

```
#Faz DNAT para um IP da rede local
iptables -t nat -A PREROUTING -i $IF_WAN  -p tcp --dport 8888 -j DNAT --to 192.168.1.100:7777
iptables -t nat -A PREROUTING -i $IF_LAN2 -p tcp --dport 8888 -j DNAT --to 192.168.1.100:7777

#Libera acesso a porta 7777/TCP para o IP da regra acima
iptables -A FORWARD -d 192.168.1.100 -p tcp --dport 7777 -j ACCEPT
```

Fonte: do autor (2019)

5 CONCLUSÃO

Manter a segurança da rede local, seja de uma grande organização, pequena empresa ou doméstica é um desafio que deve ser encarado pelo administrador. Seja ela que qual for o tamanho, a rede deve ter um cuidado especial no que se refere a segurança. Os diversos *appliances* de *firewall* que existem no mercado, sejam *open source* ou proprietárias, sem dúvuda facilitam muito na tarefa de administração. Mas conhecer o funcionamento do `iptables`, bem como dar manutenção ou montar regras customizadas, torna o adminstrador de rede mais versátil e apto a corrigir problemas do dia a dia, ou propor soluções que, por algum motivo, o *appliance* não ofereça.

Este trabalho está longe de ser um referencial completo de *firewalls* ou *netfilter/iptables*, mas fornece uma base para se iniciar os estudos e também contribui para o aperfeiçoamento do profissional interessado em iniciar nesta área do conhecimento. Acreditamos que, com os conhecimentos obtidos aqui, o aluno estará apto a dar manutenção em sistemas de *firewall* existentes e configurar sua própria solução, caso necessite. Mas somente com muita prática é que o administrador de rede, ou profissional de segurança, consegue a experiencia necessária para resolver os problemas do dia a dia.

REFERENCIAS

Acharya, H. B.; Joshi, A.; Gouda, M. G. Firewall modules and modular firewalls. In: **The 18th IEEE International Conference on Network Protocols**. [S.l.: s.n.], 2010. p. 174–182. ISSN 1092-1648. 5762766.

Boolean World. **An In-Depth Guide to iptables, the Linux Firewall**. 2019. Disponível em: <https://www.booleanworld.com/depth-guide-iptables-linux-firewall/>.

CHEN, H. et al. Tri-modularization of firewall policies. In: **Proceedings of the 21st ACM on Symposium on Access Control Models and Technologies**. New York, NY, USA: ACM, 2016. (SACMAT '16), p. 37–48. ISBN 978-1-4503-3802-8. Chen:2016:TFP:2914642.2914646. Disponível em: <http://doi.acm.org/10.1145/2914642.2914646>.

Comodo Antivirus. **What is Firewall and Types of Firewall**. 2019. Disponível em: <https://antivirus.comodo.com/blog/comodo-news/types-of-firewall/>.

Compuquip Cyber Security. **The Different Types of Firewall Architectures**. 2019. Disponível em: <https://www.compuquip.com/blog/the-different-types-of-firewall-architectures>.

DigitalOcean. **A Deep Dive into Iptables and Netfilter Architecture**. 2019. Disponível em: <https://www.digitalocean.com/community/tutorials/a-deep-dive-into-iptables-and-netfilter-architecture>.

HAGEN, P. **iptables Processing Flowchart (Updated Often)**. 2019. Disponível em: <https://stuffphilwrites.com/2014/09/iptables-processing-flowchart/>.

Murthy, U. et al. Firewalls for security in wireless networks. In: **Proceedings of the Thirty-First Hawaii International Conference on System Sciences**. [S.l.: s.n.], 1998. v. 7, p. 672–680 vol.7. 649269.

netfilter. **IPTABLES (manual)**. 2019. Disponível em: <http://ipset.netfilter.org/iptables.man.html>.

netfilter. **The netfilter.org project**. 2019. Disponível em: <https://www.netfilter.org/>.

APENDIX A – *Script* de *firewall* completo usado no mini-curso

```
#!/bin/sh

### BEGIN INIT INFO
# Provides:          f2.sh
# Required-Start:    $local_fs $remote_fs $network $syslog
# Required-Stop:     $local_fs $remote_fs $network $syslog
# Default-Start:     2 3 4 5
# Default-Stop:
# Short-Description: Start firewall at boot time
# Description:       Enable service provided by f2.sh.
### END INIT INFO

# Serial: 2019050301 ## Mini-curso ##


#######################################################
#Autor:      Diego Natividade #dn@at
#E-mail:     diego@connectivaredes.com
#Github:     /dnatividade
#
#Desc.:      Script de firewall usado no mini-curso
#######################################################

#1 ####################################################
#VARIAVEIS
#Interface de Internet
IF_WAN="enp0s3"
#Interface da "Rede Local 1" (192.168.1.0/24)
IF_LAN1="enp0s8"
#Interface da "Rede Local 2" (172.16.0.0/12)
IF_LAN2="enp0s9"

#Limpa regras (-F),
```

```
34 #exclui cadeias customizadas (-X),
35 #zera contadores (-Z)
36 for TABLE in filter nat mangle raw security
37 do
38     iptables -t $TABLE -F
39     iptables -t $TABLE -X
40     iptables -t $TABLE -Z
41 done
42 ########################################################
43
44 #2 ######################################################
45 #Politica padrao: DROP
46 iptables -P INPUT DROP
47 iptables -P FORWARD DROP
48 iptables -P OUTPUT DROP
49 ########################################################
50
51 #3 ######################################################
52 #Libera acesso na interface de localhost (lo)
53 iptables -A INPUT  -i lo -j ACCEPT
54 iptables -A OUTPUT -o lo -j ACCEPT
55 #libera acesso as conexoes ja estabelecidas e relatadas
56 iptables -A INPUT   -m state --state ESTABLISHED,RELATED -j ACCEPT
57 iptables -A FORWARD -m state --state ESTABLISHED,RELATED -j ACCEPT
58 iptables -A OUTPUT  -m state --state NEW,ESTABLISHED,RELATED -j
    ACCEPT
59 ########################################################
60
61 #4 ######################################################
62 #Libera acesso ICMP request e reply (ping e pong)
63 iptables -A INPUT -p icmp --icmp-type 8 -j ACCEPT
64 iptables -A INPUT -p icmp --icmp-type 0 -j ACCEPT
65 #Libera acesso a porta 22/TCP para todos
66 iptables -A INPUT -p tcp --dport 22 -j ACCEPT
```

```
67 #Libera acesso a porta 53/UDP somente para as subredes especificadas
68 iptables -A INPUT -p udp --dport 53 -s 172.16.0.0/12 -j ACCEPT
69 iptables -A INPUT -p udp --dport 53 -s 192.168.1.0/24 -j ACCEPT
70 #Libera acesso a porta 80/TCP somente para IP, MAC e interface
       especificados
71 iptables -A INPUT -p tcp --dport 80 -s 172.16.20.200 -m mac --mac-
       source 00:12:34:0A:BC:DE -i enp0s3 -j ACCEPT
72 ######################################################
73
74 #5 ####################################################
75 #Habilita o encaminhamento de pacotes no Kernel do Linux
76 echo 1 > /proc/sys/net/ipv4/ip_forward
77 #Faz o NAT com mascaramento para rede local
78 iptables -t nat -A POSTROUTING -o $IF_WAN -j MASQUERADE
79 ######################################################
80
81 #6 ####################################################
82 #Libera acesso ao ICMP request e reply (ping e pong) para a Internet
83 iptables -A FORWARD -o $IF_WAN -p icmp --icmp-type 8 -j ACCEPT
84 iptables -A FORWARD -o $IF_WAN -p icmp --icmp-type 0 -j ACCEPT
85 #Libera acesso a porta 80/TCP e 443/TCP para a Internet
86 iptables -A FORWARD -o $IF_WAN -p tcp --dport 80  -j ACCEPT
87 iptables -A FORWARD -o $IF_WAN -p tcp --dport 80  -j ACCEPT
88 iptables -A FORWARD -o $IF_WAN -p tcp --dport 443 -j ACCEPT
89 iptables -A FORWARD -o $IF_WAN -p tcp --dport 443 -j ACCEPT
90 ######################################################
91
92 #7 ####################################################
93 #Libera acesso ICMP request e reply (ping e pong) entre as subredes
94 iptables -A FORWARD -i $IF_LAN1 -o $IF_LAN2 -p icmp --icmp-type 8 -j
       ACCEPT
95 iptables -A FORWARD -i $IF_LAN1 -o $IF_LAN2 -p icmp --icmp-type 0 -j
       ACCEPT
```

```
96 iptables -A FORWARD -i $IF_LAN2 -o $IF_LAN1 -p icmp --icmp-type 8 -j
   ACCEPT
97 iptables -A FORWARD -i $IF_LAN2 -o $IF_LAN1 -p icmp --icmp-type 0 -j
   ACCEPT
98 #Libera acesso a porta 1111/TCP para a rede local
99 iptables -A FORWARD -i $IF_LAN2 -o $IF_LAN1 -d 192.168.1.100 -p tcp
   --dport 1111 -j ACCEPT
100 #Libera acesso a porta 80/TCP para um IP da rede local
101 iptables -A FORWARD -i $IF_LAN2 -o $IF_LAN1 -s 172.16.20.200 -d
    192.168.1.100 -p tcp --dport 80 -j ACCEPT
102 ######################################################
103
104 #8 ####################################################
105 #Faz DNAT para um IP da rede local
106 iptables -t nat -A PREROUTING -i $IF_WAN  -p tcp --dport 8888 -j DNAT
    --to 192.168.1.100:7777
107 iptables -t nat -A PREROUTING -i $IF_LAN2 -p tcp --dport 8888 -j DNAT
    --to 192.168.1.100:7777
108 #Libera acesso a porta 7777/TCP para o IP da regra acima
109 iptables -A FORWARD -d 192.168.1.100 -p tcp --dport 7777 -j ACCEPT
110 ######################################################
111
112
113 #dn@at
```

APENDIX B – *Script* de um *firewall* para uso pessoal

```
#!/bin/sh

### BEGIN INIT INFO
# Provides:          f2.sh
# Required-Start:    $local_fs $remote_fs $network $syslog
# Required-Stop:     $local_fs $remote_fs $network $syslog
# Default-Start:     2 3 4 5
# Default-Stop:
# Short-Description: Start firewall at boot time
# Description:       Enable service provided by f2.sh.
### END INIT INFO

# Serial: 2019050301 ## Mini-curso ##


#######################################################
#Autor:      Diego Natividade #dn@at
#E-mail:     diego@connectivaredes.com
#Github:     /dnatividade
#
#Desc.:      Script de firewall pessoal - DROP ALL
#######################################################


### Variaveis ###############
IF_INT="eth0"      # Colocar aqui o nome da placa de rede interna

### Mensagem de inicializa  o do Firewall ###
echo "Ativando Regras do Personal Firewall -- #dnatividade"
#####################################################

for TABLE in filter nat mangle raw security
do
```

```
34        iptables -t $TABLE -F
35        iptables -t $TABLE -X
36        iptables -t $TABLE -Z
37 done
38 ########################################################
39
40 iptables -P INPUT DROP
41 iptables -P FORWARD DROP
42 iptables -P OUTPUT DROP
43 ########################################################
44
45 iptables -A INPUT  -i lo -j ACCEPT
46 iptables -A OUTPUT -o lo -j ACCEPT
47 iptables -A INPUT  -i $IF_INT -m state --state ESTABLISHED,RELATED -j
       ACCEPT
48 iptables -A OUTPUT -o $IF_INT -m state --state NEW,ESTABLISHED,
     RELATED -j ACCEPT
49 ########################################################
```

APENDIX C – Exemplo de um *script* de *firewall* completo (para ser adaptado)

```
#!/bin/sh

### BEGIN INIT INFO
# Provides:          f2.sh
# Required-Start:    $local_fs $remote_fs $network $syslog
# Required-Stop:     $local_fs $remote_fs $network $syslog
# Default-Start:     2 3 4 5
# Default-Stop:
# Short-Description: Start firewall at boot time
# Description:       Enable service provided by f2.sh.
### END INIT INFO

# Serial: 2019050301 ## Mini-curso ##


#######################################################
#Autor:      Diego Natividade #dn@at
#E-mail:     diego@connectivaredes.com
#Github:     /dnatividade
#
#Desc.:      Script de firewall completo para estudos
#######################################################

### Variaveis ################
#interfaces de rede
IF_WAN=eth0
IF_LAN=eth1
IF_DMZ=eth2

#redes
REDE_LAN=192.168.0.0/24
#REDE_DMZ=10.10.10.0/24

```

```
34 #IPs de servidores
35 SERVER=192.168.0.11
36 DVR=192.168.0.21
37 CONTABIL=192.168.0.22
38 ###
39
40 LOGDATA=`date +%d/%m/%Y' '%T`
41 ################################
42
43 ### Mensagem de inicializa   o do Firewall ###
44 echo "Ativando Regras do Firewall -- #dnatividade"
45
46 ### Carregando modulos ###
47 modprobe ip_nat_ftp
48 modprobe ip_conntrack
49 modprobe ip_conntrack_ftp
50 #
51 modprobe ip_nat_pptp
52 modprobe pptp
53
54
55 for TABLE in filter nat mangle raw security
56 do
57         iptables -t $TABLE -F #Exclui todas as regras
58         iptables -t $TABLE -X #Exclui cadeias customizadas
59         iptables -t $TABLE -Z #Zera os contadores das cadeias
60 done
61 ######################################################
62
63 ### Define a pol tica padr o do firewall
64 iptables -P INPUT DROP
65 iptables -P OUTPUT DROP
66 iptables -P FORWARD DROP
67 ######################################################
```

```
68
69 #
70 ##
71 ###
72 ### Regras PREROUTING -- Redirecionamento de portas ###
73 iptables -t nat -A PREROUTING -i $IF_WAN -p tcp --dport 5541 -j DNAT
     --to $SERVER
74 iptables -t nat -A PREROUTING -i $IF_WAN -p tcp --dport 9000 -j DNAT
     --to $DVR
75
76 #
77 ##
78 ###
79 ### Regras INPUT ###
80 iptables -A INPUT -m state --state ESTABLISHED,RELATED -j ACCEPT
81 iptables -A INPUT -i lo -j ACCEPT
82 iptables -A INPUT -p icmp --icmp-type 0 -j ACCEPT
83 iptables -A INPUT -p icmp --icmp-type 8 -j ACCEPT
84 iptables -A INPUT -p tcp --dport 22 -j ACCEPT
85 iptables -A INPUT -p tcp --dport 53 -j ACCEPT
86 iptables -A INPUT -p udp --dport 53 -j ACCEPT
87
88 #
89 ##
90 ###
91 ### Regras FORWARD ###
92 iptables -A FORWARD -m state --state ESTABLISHED,RELATED -j ACCEPT
93
94 #da interface "IF_LAN" para "IF_WAN"
95 iptables -A FORWARD -i $IF_LAN -p icmp --icmp-type 0 -j ACCEPT #Ping
96 iptables -A FORWARD -i $IF_LAN -p icmp --icmp-type 8 -j ACCEPT #Ping
97 ###
98 iptables -A FORWARD -i $IF_LAN -m mac --mac-source 01:23:45:AA:BB:CC
     -p tcp --dport 80  -j ACCEPT #libera MAC para acessar porta 80/TCP
```

```
99 iptables -A FORWARD -i $IF_LAN -m mac --mac-source 01:23:45:AA:BB:CC
   -p tcp --dport 443 -j ACCEPT #libera MAC para acessar porta 443/
   TCP
100 ###
101 iptables -A FORWARD -i $IF_LAN -s $SERVER -p tcp --dport 80  -j
   ACCEPT #Libera IP do servidor para acessar porta 80/TCP
102 iptables -A FORWARD -i $IF_LAN -s $SERVER -p tcp --dport 443  -j
   ACCEPT #Libera IP do servidor para acessar porta 443/TCP
103 iptables -A FORWARD -i $IF_LAN -s $SERVER -p udp --dport 5000 -j
   ACCEPT #ACCEPT #Libera IP do servidor para acessar porta 5000/UDP
104 ###
105 iptables -A FORWARD -i $IF_LAN -p tcp --dport 22 -j ACCEPT #Libera
   todos da rede para acessar porta 22/TCP
106 iptables -A FORWARD -i $IF_LAN -p tcp --dport 21 -j ACCEPT #Libera
   todos da rede para acessar porta 21/TCP
107 iptables -A FORWARD -i $IF_LAN -p tcp --dport 20 -j ACCEPT #Libera
   todos da rede para acessar porta 20/TCP
108 ###
109 iptables -A FORWARD -i $IF_LAN -s $CONTABIL -d 1.2.3.4 -p tcp --dport
    80  -j ACCEPT #Permite um IP interno acessar um IP externo na
   porta 80
110 iptables -A FORWARD -i $IF_LAN -s $DVR -d 1.2.3.0/24 -j ACCEPT #
   Permite um IP interno acessar um bloco de IPs /24 externo, em
   qualquer porta/protocolo
111 ###
112 iptables -A FORWARD -i $IF_LAN -i $IF_WAN -s $CONTABIL --sport 8817 -
   d 1.2.3.5 --dport 1393 -p udp -j ACCEPT #Permite um IP/porta
   interno acessar um IP/porta externo
113
114 #da interface "IF_WAN" para "IF_LAN"
115 iptables -A FORWARD -i $IF_WAN -o $IF_LAN -d $SERVER -p tcp --dport
   5541 -j ACCEPT
116 iptables -A FORWARD -i $IF_WAN -o $IF_LAN -d $DVR    -p tcp --dport
   9000 -j ACCEPT
```

```
117
118 #da interface "IF_LAN" para "IF_DMZ"
119 #iptables -A FORWARD -i $IF_LAN -o $DMZ -p icmp --icmp-type 0 -j
    ACCEPT
120 #iptables -A FORWARD -i $IF_LAN -o $DMZ -p icmp --icmp-type 8 -j
    ACCEPT
121 #iptables -A FORWARD -i $IF_LAN -o $DMZ -p tcp --dport 3389 -j ACCEPT
122 #iptables -A FORWARD -i $IF_LAN -o $DMZ -p tcp --dport 139 -j ACCEPT
123
124 #da interface "IF_DMZ" para "IF_LAN"
125 #iptables -A FORWARD -i $IF_DMZ -o $IF_LAN -p icmp --icmp-type 0 -j
    ACCEPT
126 #iptables -A FORWARD -i $IF_DMZ -o $IF_LAN -p icmp --icmp-type 8 -j
    ACCEPT
127 #iptables -A FORWARD -i $IF_DMZ -o $IF_LAN -p tcp --dport 3389 -j
    ACCEPT
128 #iptables -A FORWARD -i $IF_DMZ -o $IF_LAN -p tcp --dport 139 -j
    ACCEPT
129
130 #
131 ##
132 ###
133 ### Regras OUTPUT ###
134 iptables -A OUTPUT -m state --state NEW,ESTABLISHED,RELATED -j ACCEPT
135
136 #
137 ##
138 ###
139 ### Regras POSTROUTING ###
140 echo 1 > /proc/sys/net/ipv4/ip_forward
141
142 #Configurando NAT
143 iptables -t nat -A POSTROUTING -s $REDE_IF_LAN -o $IF_WAN -j
    MASQUERADE #NAT: mascaramento (compartilhar Internet)
```

```
144 #iptables -t nat -A POSTROUTING -o $IF_WAN -j MASQUERADE #NAT:
        mascaramento (compartilhar Internet)
145
146 #dn@at
```