Configurando Firewall com Iptables Ubuntu

Fonte: http://wime.com.br/2015/03/04/como-configurar-um-firewall-usando-o-iptables-ubuntu-14-04/
Adaptado por Kelton Costa

A instalação e configuração de um firewall é um requisito de segurança essencial em qualquer sistema operacional. A maioria das distribuições Linux possui ferramentas/interfaces de firewall que podem ser utilizados para configurar os firewalls. O foco deste artigo é o firewall Iptables.

Iptables é um firewall incluído na maioria das distribuições Linux. Na verdade, o Iptables é um front-end para utilizar os recursos do netfilter que roda a nível do kernel e que podem manipular as pilhas de redes Linux. O Iptables é utilizado para controlar quais pacotes podem entrar ou sair pela rede interna ou externa, assim ele poderá bloquear o tráfego indesejado de pacotes.

Este artigo demonstra um exemplo prático para criar uma regra básica definida para um servidor Ubuntu 14.04.

1 - Iptables – Comandos Básicos

Importante ressaltar que os comandos Iptables devem ser executados com privilégios de root, portanto utilizar o comando `su` ou `sudo -su` para acessar o shell como root. Um bom ponto de partida é listar as regras atuais que estão configurados para iptables, para isto será necessário utilizer o parâmetro `-L`:

```
sudo 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
```

É visto portanto três cadeias padrão (INPUT, OUTPUT, e FORWARD) bem como a política padrão de cada regra (cada uma dessas cadeias tem ACCEPT como política padrão), sendo assim, a missão é a de bloquear todo tráfego indesejado.

Podemos ver a saída de como seria os comandos para modificar as cadeias:

```
sudo iptables -S
-P INPUT ACCEPT
-P FORWARD ACCEPT
-P OUTPUT ACCEPT
```

Nota: Se já exitem algumas regras no firewall e deseja desfazer-se, digite o comando "sudo iptables –F". O referido comando exclui todas as regras do firewall, mas a politica padrão permanecerá, ou seja, se a política padrão estiver configurada para bloquear todas as entradas e você entrar com o comando estando conectado remotamente via SSH, certamente você perderá o acesso remoto ao servidor imediatamente. Para evitar problemas, sempre que for necessário limpar as regras do firewall com o Iptables, antes de qualquer coisa, altere a política padrão, e, em seguida, você poderá limpar as regras já existentes, para isso, você poderá utilizar os seguintes comandos:

```
sudo iptables -P INPUT ACCEPT
sudo iptables -P OUTPUT ACCEPT
sudo iptables -F
```

2 - A primeira regra

Iremos iniciar a construção das políticas de firewall. Vamos começar trabalhando com a cadeia INPUT, responsável por todo tráfego de entrada no servidor. Vamos inserir uma regra que permite a conexão SSH. A regra completa necessária é :

```
sudo iptables -A INPUT -m conntrack --ctstate ESTABLISHED,RELATED -j ACCEPT
```

onde:

- **-A INPUT**: O parâmetro `-A` acrescenta uma regra para a extremidade de uma cadeia. Este é o parâmetro do comando que diz ao Iptables para adicionar uma nova regra, e que queremos que regra adicionada fique no final da cadeia, e queremos operar na cadeia INPUT.
- **-m conntrack**: o Iptables tem um conjunto de funcionalidades básicas, mas também tem um conjunto de extensões ou módulos que oferecem recursos extras.

Nesta parte do comando solicitamos ter acesso à funcionalidade fornecida pelo módulo `conntrack`. Este módulo dá acesso a comandos que podem ser usados para tomar decisões com base na relação dos pacotes para ligações anteriores. * **–ctstate**: Este é um dos comandos disponibilizados peloo módulo `conntrack`. Este comando permite corresponder pacotes com base em como eles estão relacionados com os pacotes que

já vimos antes. O valor `ESTABLISHED` permite que os pacotes que fazem parte de uma conexão existente seja utilizado. O valor `RELATED` permite pacotes que estão associados com uma conexão estabelecida. Esta é a parte da regra que corresponde a sessão SSH atual. * **-j ACCEPT**: Isso especifica o alvo de pacotes que combinem, informamos ao Iptables que os pacotes que correspondem aos critérios anteriores, devem permitir sua passagem. Colocamos esta regra no início, porque queremos garantir que as conexões que já estão em uso sejam correspondidas, aceitas, e retiradas da cadeia antes de chegar a qualquer regra de bloqueio (DROP). Podemos também ver as mudanças no Iptables utilizando o comando para listar regras:

```
sudo iptables -L
Chain INPUT (policy ACCEPT)
target     prot opt source               destination
ACCEPT     all  --  anywhere             anywhere             ctstate RELATED,ESTAB
LISHED


Chain FORWARD (policy ACCEPT)
target     prot opt source               destination


Chain OUTPUT (policy ACCEPT)
target     prot opt source               destination
```

3 - Aceitar outras conexões necessárias

Foi informado ao Iptables para manter aberto quaisquer conexões que já estão abertas e permitir novas conexões relacionadas a tais conexões. No entanto, precisamos criar algumas regras para estabelecer quando queremos aceitar novas conexões que não atendam a esses critérios. Vamos adicionar algumas regras para permitir conexões em duas portas especificamente. Desejamos manter a porta SSH aberta (porta 22. Iremos também sugerir que este computador executa um servidor Web na porta padrão 80. As duas linhas que para adicionar essas regras são:

```
sudo iptables -A INPUT -p tcp --dport 22 -j ACCEPT
sudo iptables -A INPUT -p tcp --dport 80 -j ACCEPT
```

Como é possível ver, esses comandos são muito semelhantes à primeira regra, porém mais simples. As novas opções são:

- **-p tcp**: Esta opção é utilizada caso os pacotes sejam do protocolo TCP. Este é um protocolo com base na ligação que irá ser utilizada por maior parte das aplicações, pois permite uma comunicação confiável.

- **–dport**: Esta opção está disponível se o parâmetro `-p tcp` estiver presente. Este requisito corresponde a porta de destino para o pacote correspondente. A primeira regra corresponde aos pacotes TCP com destino a porta 22, enquanto que a segunda regra corresponde ao tráfego TCP apontado para a porta 80.

Existe mais uma regra para adicionar egarantir que o servidor funcione corretamente. Muitas vezes, determinados serviços instalados no computador precisam se comunicar com outros serviços dentro da rede, e fazem isso através da utilização de uma interface de rede chamada `loopback device`, que direciona o tráfego de volta para si mesmo e não para outros computadores. Portanto, se um serviço quer comunicar com outro serviço que está atendendo as conexões através da porta 4555, ele poderá enviar um pacote para a porta 4555 do dispositivo de auto-retorno. Queremos que este tipo de comportamento seja permitido, porque é essencial para o bom funcionamento de muitos programas. A regra que você precisa adicionar é esta:

```
sudo iptables -I INPUT 1 -i lo -j ACCEPT
```

- **-I INPUT 1**: A parâmetro `-I` diz para o Iptables inserir uma regra. Mas, é diferente do parâmetro `-A`, que acrescenta uma regra no final. O parâmetro `-I` insere uma nova regra na posição que você deseja dentro da cadeia de regras.
- *** -i Lo**: A interface “lo” é outro nome para o dispositivo de auto-retorno. Isto significa que qualquer pacote que esteja usando esta interface para se comunicar, deverá ser aceito.

Para ver as regras atuais, devemos usar o parâmetro `-S`. Isso ocorre porque o parâmetro `-L` não inclui algumas informações, como a interface que a regra está vinculada.

```
sudo iptables -S
-P INPUT ACCEPT
-P FORWARD ACCEPT
-P OUTPUT ACCEPT
-A INPUT -i lo -j ACCEPT
-A INPUT -m conntrack --ctstate RELATED,ESTABLISHED -j ACCEPT
-A INPUT -p tcp -m tcp --dport 22 -j ACCEPT
-A INPUT -p tcp -m tcp --dport 80 -j ACCEPT
```

4 - A implementação da regra DROP

Tem-se também quatro regras distintas que aceitam explicitamente pacotes com base em determinados critérios. No entanto, o firewall ainda não está bloqueando nada. Conforme visto anteriormente, o objetivo do Iptables é configurar um firewall que permita somente o tráfego de pacotes confiáveis e em portas que estão sendo utilizadas por serviços que precisamos, por

exemplo, se você quer rodar servidor web, como Apache, provavelmente você precisará da porta 80, então é necessário adicionar ao firewall uma regra que permita isso. Até o momento, a política padrão do firewall é aceitar tudo. Há duas maneiras distintas para tal tarefa:

A primeira maneira é modificar a política padrão da cadeia INPUT.

```
sudo iptables -P INPUT DROP
```

Neste momento o firewall está bloqueando todos os pacotes de entrada, exceto aqueles que permitimos nas regras adicionados no tópico anterior.

Utilizando atribuir o DROP como política padrão, você precisará adicionar regras que liberem as portas que você precisá utilizar. Se não fizer isso, o serviço não poderá ser acessado externamente, por exemplo, se o Apache está sendo executado na porta 80 e não foi adicionado as regras para liberá-la, quando tentar acessar o site, o navegador não conseguirá executar.

5 - Modelo de firewall com Iptables

Para a configuração do firewall, existe um modelo básico, porém eficiente para seguir. **Nota**: não esquecer de remover regras de serviços que não estão rodando no seu servidor.

```
### Exclui todas as regras ###
Iptables -t nat -F
Iptables -t mangle -F
Iptables -t filter -F

### Exclui cadeias customizadas ###
Iptables -X

### Zera os contadores das cadeias ###
Iptables -t nat -Z
Iptables -t mangle -Z
Iptables -t filter -Z

### Define a política padrão do firewall ###
Iptables -P INPUT DROP
Iptables -P OUTPUT DROP
Iptables -P FORWARD DROP

### Regras INPUT ###
```

```
### informa os estados que devem ser checados (Conexão estabelecida
# ou Relacionada). Caso o estado da conexão seja uma dessas 2, então
# ele vai aceitar.
Iptables -A INPUT -m state --state ESTABLISHED,RELATED -j ACCEPT

# Libera o INPUT para a interface loopback, ou seja, a própria máquina
Iptables -A INPUT -i lo -j ACCEPT

# Permite icmp 0 (resposta de Echo)
Iptables -A INPUT -p icmp --icmp-type 0 -j ACCEPT

# Permite icmp 8 (Pedido de Echo)
Iptables -A INPUT -p icmp --icmp-type 8 -j ACCEPT

# Permite o acesso ao servidor usando SSH
Iptables -A INPUT -p tcp --dport 22 -j ACCEPT

# Permite acesso do Apache na porta 80
Iptables -A INPUT -p tcp --dport 80 -j ACCEPT

# Permite acesso do Apache na porta 443 (https)
Iptables -A INPUT -p tcp --dport 443 -j ACCEPT

# Permite o acesso ao servidor usando FTP
Iptables -A INPUT -p tcp --dport 21 -j ACCEPT

# Permitir acesso externo ao MySQL Server
Iptables -A INPUT -p tcp --dport 3306 -j ACCEPT

# PassivePorts Proftpd
Iptables -A INPUT -p tcp --dport 49152:65534 -j ACCEPT

# Permitir listagem de diretorios FTP com Proftpd
/sbin/modprobe ip_nat_ftp

# Permitir acesso ao servidor de envio de email
Iptables -A INPUT -p tcp --dport 110 -j ACCEPT
```

```
### Regras OUTPUT ###
Iptables -A OUTPUT -m state --state NEW,ESTABLISHED,RELATED -j ACCEPT
```

6 - Salvando as configuração Iptables

Por padrão, as regras que adicionadas ao Iptables são temporários. Isto significa que quando reiniciar o servidor, as regras serão perdidas. Por um lado pode ser positivo, pois, quando estiver criando seu firewall e, acidentalmente aplicar alguma regra que bloqueie seu acesso, poderá reiniciar o servidor e tudo voltará ao normal. Mas assim que terminar suas configurações, com certeza desejará que as configurações de firewall inicie no start do servidor. O pacote necessário para realizar tal desejo é o `iptables-persistent`.

```
sudo apt-get update
sudo apt-get install iptables-persistent
```

Durante a instalação, deverá informar se deseja salvar as regras atuais para que sejam carregadas automaticamente bem com irá perguntar se deseja salvar as regras IPv6 configuradas.
Nota: as regras são configuradas através de um utilitário chamado **ip6tables** de forma separadas para controlar o fluxo de pacotes IPv6.

Quando a instalação estiver concluída, o sistema terá um novo serviço chamado `iptables-persistent` que está configurado para executar na inicialização. Este serviço irá carregar as regras e aplicá-las.

7 – Considerações

Este estudo é o ponto de partida para o desenvolvimento de um firewall que atenda às necessidades particulares e organizacionais. Há diversos outros utilitários de firewall e alguns que podem ser até mais fácil, mas cabe ressaltar que o Iptables é uma boa ferramenta de aprendizagem, mesmo porque ele expõe parte da estrutura Netfilter e também está presente em muitos outros sistemas.