Tema: Tecnologias para Segurança em Redes
TP: Práticas com Firewalls (IPTables)

Introdução
O Kernel do Linux, por defeito, inclui a capacidade para processar e filtrar todo o tráfego que passa na pilha de protocolos. Utilizando esta funcionalidade do Linux, podemos "facilmente" configurar uma máquina como *router* e como *firewall* (neste caso estamos particularmente interessados nesta segunda funcionalidade).

De uma forma simples, no âmbito do funcionamento como *firewall*, em que nos interessa a tabela ***filter*** (ver manual do IPTables[1] - http://linux.die.net/man/8/iptables), todos os pacotes são sujeitos a uma de três **cadeias** primárias de **regras**:

1. **Regras** da cadeia **INPUT** – aplicam-se a todos os pacotes que são endereçados para a máquina local;
2. **Regras** da cadeia **FORWARD** – aplicam-se a todos os pacotes que chegam à máquina local (enquanto *router*) e que se destinam a outras máquinas; e
3. **Regras** da cadeia **OUTPUT** – aplicam-se a todos os pacotes gerados na máquina local e que se destinam a outras máquinas.

Cada uma destas cadeias de regras precisa de ser configurada pelo administrador, segundo alguma política de segurança, para atingir um nível de segurança desejado. Tipicamente, essa configuração é feita por *scripts,* escritos numa linguagem relativamente simples, mas que, pelas dependências e relações possíveis, facilmente se transforma num verdadeiro "pesadelo". O comando em Linux que permite administrar essas cadeias de regras é o **iptables**. Existem diversas aplicações (algumas gráficas) que ajudam a manter essas regras, ou até mesmo serviços do SO, como seja o **firewallD** (por defeito no CentOS 7). No Ubuntu encontra-se normalmente disponível a aplicação **GuFw** (http://gufw.org/) – *Graphical user interface for* **ufw** (https://help.ubuntu.com/community/UFW) – que é uma excelente alternativa. No CentOS (uma popular implementação derivada do projeto Fedora) está disponível uma outra aplicação "muito simples" designada por **system-config-firewall**[2], que para além de uma versão GUI tem também uma versão TUI (**system-config-firewall-tui**) – e cuja utilização é proposta neste trabalho.

Antes de começarmos a utilizar estas ferramentas, há alguns detalhes que convém ainda realçar (ver também o manual acima referido):

1. A forma genérica de um comando do **iptables** para criar uma regra (tabela **filter**) é
   `iptables -A CHAIN -p tcp/udp [options] -j ACTION`
   todos os parâmetros são mais ou menos óbvios, mas convém realçar alguns dos valores que ACTION pode assumir (ACCEPT, DROP e LOG)

   A forma genérica de um comando do **iptables** para eliminar uma regra é
   `iptables -D CHAIN num_regra`

   Mas como veremos há um comando particularmente útil, que é
   `iptables -F [CHAIN]`

[1] Este é o manual oficial do IPTables, mas é possível encontrar na Internet outros manuais mais ilustrativos…
[2] A descrição desta ferramenta, assim como de alguns conceitos fundamentais associados ao **iptables** está disponível em https://fedoraproject.org/wiki/How_to_edit_iptables_rules

que permite apagar todas as regras de todas as cadeias (☺), ou de uma só, em particular.

2. Uma sequência de comandos que poderá querer utilizar diversas vezes (!!!)

`iptables -F`	Elimina todas as regras, em todas as cadeias
`iptables -t nat -F`	Elimina todas as regras da tabela NAT
`iptables -t mangle -F`	Elimina todas as regras da tabela MANGLE
`iptables -X`	Elimina as regras definidas pelo utilizador

   Nota: as tabelas **nat** e **mangle** são utilizadas pelo mecanismo de processamento de pacotes e nunca devem ser utilizadas para construir regras. Por isso, por omissão do *switch* –t, todos os comandos do iptables se referem à tabela **filter**. Sendo assim, na maioria das vezes não será necessário eliminar daí regras (mas enganos podem acontecer).
3. É muito útil configurar o kernel para registar os *logs* gerados pela firewall (o que já deve estar ativado, por defeito). Em complemento, o comando `iptables-save [-c] [-t table]` permite obter (e preservar) os totais relativamente aos pacotes e bytes que foram processados em cada cadeia, ao mesmo tempo que salvaguarda o conjunto de regras. Essa informação pode ser reposta pelo comando `iptables-restore` (mas este tem algumas particularidades…); e não se esqueça que aquilo que faz com o **iptables** não é persistente (quando reinicializar o sistema, as alterações perdem-se)!

Material

Para executar este exercício irá necessitar de duas máquinas. Um servidor, onde vai ativar e configurar a *firewall* (e onde deve ter, pelo menos, os serviços HTTP, FTP e SSH) e um cliente que vai servir apenas para teste. Como sugestão, instale no seu computador um *software* de virtualização (e.g VMPlayer, VirtualBox, ou QEMU/Kvm) e instale imagens já prontas a utilizar, como seja: o **CentOS** (recomenda-se a versão 6.7 ou posterior, com Gnome, mas não a versão 7), uma excelente alternativa como servidor (imagens disponíveis em http://www.osboxes.org)[3]; e o **Kali**, ou qualquer outra compilação de ferramentas de segurança com que esteja familiarizado, uma vez que apenas será utilizado como cliente – não obstante, o Kali terá mais funcionalidades para testes, nomeadamente o Wireshark ☺.

Sumário das tarefas

1. Arrancar com ambas as máquinas, de preferência configurando os respetivos interfaces de rede no modo NAT (particularmente se usar virtualização) – isto permite-lhe uma utilização mais protegida. Este processo é considerado trivial, não sendo considerado necessário fornecer mais detalhes.
2. Instalar (se necessário) e configurar no servidor os serviços desejados e os programas **iptables** e **system-config-firewall** (no CentOS recomendado o esforço será mínimo, mas se escolher outro SO terá que identificar comandos equivalentes).
   NOTA: Se optar pelo CentOS 7 é possível que venha por defeito configurado para usar o serviço **firewallD** – em produção, este serviço é bastante mais eficiente na gestão da *firewall* e na utilização do **iptables**, com o custo adicional de tornar muito mais obscura a

[3] O CentOS aparece como uma versão mínima de um Linux construído com a segurança e a fiabilidade como requisitos centrais. Por esse motivo implementa um número de serviços mínimo bastante fechado, nem sequer incluindo um interface gráfico. Não obstante aparecem algumas imagens que incluem um interface gráfico (tipicamente o Gnome) e que facilitam a administração, muito embora comprometendo os requisitos iniciais. É uma dessas imagens (com Gnome) que se recomenda neste exercício.

utilização do **iptables**, o que não se pretende neste exercício; por esse motivo, poderá ter que desativar o serviço **firewallD** para dispor do system-config-firewall[4].

3. Testar a segurança e a funcionalidade do servidor (versão 1).
4. Afinar a configuração através dos mesmos programas, no servidor (versão 2) e testar novamente.

Tarefa 1

Nota prévia: por questões de segurança e desempenho, deve sempre manter o *software* atualizado. Esta observação é válida também para MVs, a não ser que exista alguma indicação específica para não o fazer.

No servidor e partir de uma consola:

1. Assumindo que está a utilizar o CentOS 6.x, precisará de instalar o serviço FTP (**Sytem → Administration → Add/Remove Software** e escolher, na categoria de servidores, o FTP seguro – *Very Secure FTP Daemon*); seguidamente precisará de ativar e iniciar os serviços desejados, que são o "httpd", o "sshd" e o "vsftpd" (**Sytem → Administration → Services**, selecionando cada um dos serviços desejados e, primeiro ativar – **enable** –, para depois iniciar – **start**); poderá ainda querer configurar o teclado e outros aspetos do interface…
   No final 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.
2. Execute o comando **system-config-firewall-tui** (tal como referido anteriormente, este programa estará disponível, em princípio, em qualquer implementação Linux derivada do projeto Fedora, que é o caso do CentoOS). Este comando permite configurar de forma muito simples a *firewall*, através do *iptables*.
   Deverá aparecer-lhe um ecrã como o que é mostrado na Figura 1, o qual mostra que a *firewall* está desligada. Nessa janela (com a tecla <Tab>, ou as teclas de movimento do cursor, e a tecla <espaço>) ative a opção **Enable,** selecionando depois o botão **Ok**, que terminará a execução do programa – dependendo da implementação que estiver a usar a *firewall* pode inicialmente estar já ativada, não tendo, nesse caso, que fazer nada.

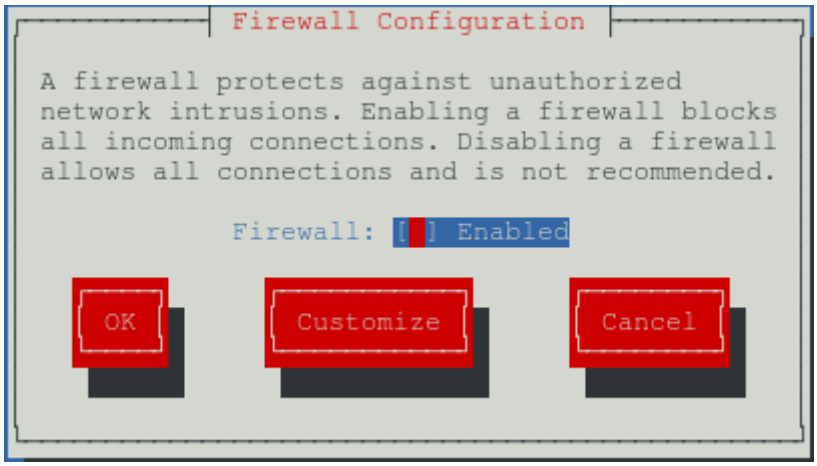

Figura 1 – Ecrã do programa system-config-firewall-tui

[4] Encontra um sumário das instruções para realizar essa alteração da configuração em https://www.digitalocean.com/community/tutorials/how-to-migrate-from-firewalld-to-iptables-on-centos-7

3. Execute **iptables –L –v** (como é habitual, a execução do comando irá necessitar de direitos de "*super user*"); registe o resultado obtido e indique, justificando:
    a. Quais são as políticas por defeito para cada uma das cadeias INPUT, FORWARD e OUTPUT;
    b. Comente o nível de segurança e demais informação que consiga extrair do resultado da execução do comando.
4. 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. Se em qualquer fase do trabalho se sentir perdido(a), pode sempre voltar a este ponto e repor as tabelas com os seus valores iniciais, com o comando **iptables-restore < iptables.dump** (mas antes de o fazer é aconselhável "limpar" todas as tabelas, com o comando **iptables –F**, como foi inicialmente referido).
5. Usando de novo o comando **system-config-firewall-tui**, desative a firewall e execute novamente o comando **iptables –L –v**
    a. Registe e analise as novas regras, procurando verificar se são mais seguras ou não e porquê.
6. Volte a ligar a *firewall*, revertendo a operação realizada no passo anterior.

Tarefa 2

No Cliente e partir de uma consola:

1. Comece por verificar a conetividade, executando o comando **ping <ip_address>** onde <ip_address> representa, naturalmente, o endereço IP do servidor (essa mesma convecção é usada nas restantes fases e tarefas).
2. Execute o comando **nmap –sS <ip_address>**
    a. Que informação lhe forneceu o programa? Pode aproveitar para explorar algumas das funcionalidades adicionais desta ferramenta.
3. Execute o comando **w3m http://<ip_address>**
   Nota: este comando é um *browser* que funciona em modo texto, idêntico ao popular **lynx** (disponível em outras implementações do Linux, como o Ubuntu) e que pode ser utilizado da mesma forma; dependendo do cliente que está a usar, pode ter que instalar um desses comandos!
    a. Conseguiu visualizar uma página? Registe a resposta que obteve.
4. Execute de seguida o comando **ftp <ip_address>**
    a. Conseguiu aceder ao servidor? Registe a resposta que obteve.
5. Execute finalmente o comando **ssh <ip_address>**
   Nota: sendo a primeira vez que se liga ao servidor é natural que lhe apareça uma solicitação relativa à possível aceitação do certificado público do servidor. Se tal acontecer deve aceitar o certificado.
    a. Conseguiu aceder ao servidor? Registe a resposta que obteve.

Em princípio deverá ter conseguido proteger bastante bem o servidor… ao ponto de não permitir que ele responda (quase) a qualquer pedido de serviço ☹.

Tarefa 3

No servidor:

1. Execute novamente o comando **system-config-firewall-tui**. Desta vez selecione a opção **Customize**. Com as teclas de direção e de espaço selecione os protocolos **FTP**, **SSH** e **WWW**, os quais deseja que fiquem acessíveis. Se pretender abrir mais alguns serviços ou portas pode fazê-lo. Para avançar no processo de configuração escolha a opção **Forward**,

surgindo-lhe uma janela onde terá a possibilidade autorizar portas que não estão diretamente identificadas pelos nomes de serviços, na janela anterior. Não precisa de alterar nada nesta fase. Continue escolhendo a opção **Forward**, a qual o levará às fases seguintes da configuração: (1) seleção de interfaces de rede que pretende que tenham acesso completo (nada a alterar, também); (2) as interfaces de rede que pretende que sejam "mascaradas"; (3) a ativação da função de *port forwarding*; (4) o filtro para o protocolo ICMP, o qual lhe permite selecionar os comandos ICMP que a *firewall* irá filtrar – para efeitos do exercício selecione a opção **Echo Request (ping)**; (5) e, finalmente, o editor de regras customizadas, onde poderia introduzir regras específicas que serão adicionadas no final (aqui também não deve fazer qualquer alteração). Terminado o ciclo de configuração irá regressar à janela inicial, onde irá selecionar o botão Ok e aceitar as alterações, terminando a execução do programa.

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

No Cliente e a partir de uma consola:

3. Execute o comando **ping <ip_address>**. Registe a resposta obtida e comente-a.
4. Execute o comando **w3m http://<ip_address>**
    a. Desta vez conseguiu visualizar uma página? Registe a resposta obtida. Para abortar a aplicação basta pressionar a tecla "q"
5. Execute de seguida o comando **ftp <ip_address>**
    a. Desta vez conseguiu obter uma ligação? Registe a resposta obtida. Se obteve uma ligação mas não conseguir fazer *login*, isso não é estranho, uma vez que não configurou o servidor vsFTPd (que deverá estar a usar). Mas talvez tenha sucesso se tentar o utilizador "anonymous"…
6. Execute o comando **nmap –sS <ip_address>**
    a. Registe a informação lhe forneceu desta vez o programa. Compare-a com a que obteve anteriormente e reflita sobre o nível de segurança atual.

No servidor:

7. Execute novamente o comando **iptables –L -v**
    a. Registe as alterações que observa e procure justificar o que é possível observar, à luz da atividade desta tarefa.

Conclusão:

Este breve exercício não lhe permite ainda construir regras adequadas para uma *firewall* a implementar num cenário real. No entanto serviu para introduzir os conceitos básicos essenciais para, por exemplo, conseguir interpretar e/ou ajustar os diversos conjuntos de regras, mais ou menos *standards*, que pode encontrar na Internet, para diferentes ambientes.

Por outro lado, já estão disponíveis diversos interfaces gráficos que ajudam a configurar de uma forma mais simples e eficiente o iptables. São exemplos:

- system-config-firewall (uma alternativa GUI ao TUI usado no trabalho; atenção que os dois não podem ser usados de forma complementar, pois em particular a versão GUI inicializa completamente a *firewall*, sempre que faz alguma alteração);
- fwbuilder;
- Turtle Firewall Project;
- ISCS (Integrated Secure Communications System);
- IPMenu;

- Easy Firewall Generator; e
- config-firewall, que como foi referido, no CentOS 7 recorre ao serviço firewallD para configurar o iptables de uma forma mais eficiente e "profissional".

Há também um conjunto alargado de produtos comerciais baseados no Linux, iptables e netfilter.

Para finalizar o trabalho, procure instalar uma das interfaces gráficas e explore as suas funcionalidades, reforçando o seu conhecimento relativamente a esta importante ferramenta de segurança em redes, que é o **iptables**. Em particular procure melhorar a proteção implementada na **Tarefa 3**, limitando o acesso a endereços locais e ativando a função de **log** que é suportada pelo **iptables** – naturalmente que deverá registar no seu *logbook* os resultados das experiências realizadas para verificar essas alterações.