Capítulo 8 – Firewalls

Uma *firewall* tem dois objetivos fundamentais:

(i) Proteção por isolamento de máquinas ligadas à rede

(ii) Controlo de interações entre máquinas.

Em ambos os casos as decisões tomadas por uma *firewall* são controladas por um conjunto de regras e aplicação que as interpretam e reagem em função do tráfego que chega à *firewall*.

A proteção por isolamento de uma máquina ligada à rede é atualmente um requisito critico, tanto para máquinas pessoais como organizacionais. É uma vantagem, porque lhe permite usar serviços contactando outras máquinas ligadas direta ou indiretamente a essa rede. É também uma vantagem, porque lhe permite disponibilizar serviços a essas mesmas máquinas. Mas é um risco, pois expõe vulnerabilidades da máquina que podem ser exploradas por atacantes.

Uma *firewall* é um elemento indispensável na ligação de máquinas pessoais e redes privadas a redes privadas a redes alheias potencialmente perigosas.

É um elemento indispensável na ligação de uma rede privada à internet:
- Controlo de acesso
- Controlo de fluxo
- Controlo de conteúdos

Permite concretizar políticas de segurança de uma forma centralizada:
- Minimiza o impacto de vulnerabilidades locais
- Facilita a tomada de posições mais drásticas
- Centraliza a deteção de problemas e o seu tratamento

Uma *firewall* é um elo de ligação entre os sistemas computacionais (conjunto de redes e máquinas) que se pretende proteger, designado por perímetro protegido e as redes a que esse perímetro está ligado através da *firewall*.

A *firewall* é construída por diversas componentes funcionais, quer de hardware – máquinas, redes e equipamentos de interligação como *hubs, switches, gateways, routers*, etc – quer de software – aplicações específicas para filtrar, controlar e modificar fluxos de comunicação.

8.1. Funcionalidade

- Supervisão de toda a comunicação IN ↔ OUT
  - Controlo
    - Do uso dos recursos protegidos por máquinas exteriores
    - Do uso da rede exterior pelas máquinas do perímetro protegido
  - Defesa
    - Contra ataques externos ao perímetro protegido
    - Contra ataques iniciados no interior lançados para o exterior

- Acionamento de mecanismo próprio de gateways

O NAT tem um duplo objetivo:

(i) Simplificar a gestão de endereços das redes internas ligadas à internet através do *gateway*

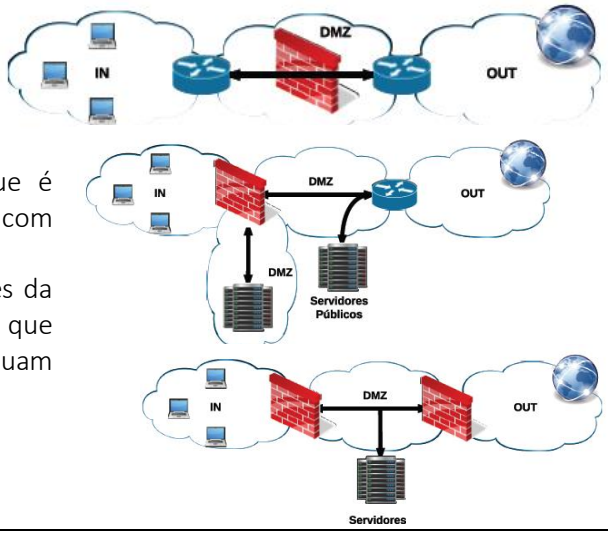
(ii) Impedir um endereçamento *ad hoc* de máquinas internas originado na rede externa.

Para esconder a estrutura do perímetro protegido (NAT – *Network Address Translation*)	
IP Masquerading	*Port Forwarding*
A tradução de endereços IP, designada por IP *masquerading* ou DNAT (*Dynamic NAT*), visa esconder uma rede privada atrás de endereços públicos da sua *gateway*. A técnica de IP *masquerading* é útil para a segurança fornecida por uma *firewall*, porque permite que máquinas interiores à mesma iniciem contactos com o exterior mas impede o contrário. No entanto, o que interessa é que as máquinas interiores não estão visíveis nem contactáveis a partir do exterior, a menos que iniciem algum contato com serviços localizados em máquinas exteriores.	O mecanismo de *porr forwarding* ou *Static* NAT (SNAT), é complementar ao anterior: permite que acessos originados no exterior possam chegar até um serviço localizado numa máquina interior cujo IP não é público. A rede interna pode disponibilizar serviços públicos, uma vez que os mesmos podem ser acedidos do exterior através de mapeamentos fixos feitos através de *port forwarding*.
Para estender o perímetro de segurança	
Encapsulamento (*tunneling*)	
O encapsulamento consiste em colocar datagramas de um protocolo como corpo de datagramas de outros protocolos, em conjunto com um cabeçalho descritivo do encapsulamento. Esta técnica permite transportar datagramas através de redes onde normalmente não circulariam. As *firewalls*, por serem o elo de ligação de uma rede organizacional à Internet, estão num local privilegiado para efetuarem o encapsulamento de vários protocolos, quer sejam ou não suportados pela internet. Quando o encapsulamento se mistura com a comunicação cifrada, dá origem a redes virtualmente privadas (VPN).	

8.2. Estrutura genérica e variações

DMZ (zona desmilitarizada ou rede de Perímetro)

A DMZ é a rede inerente à *firewall*, ou seja, a rede que estabelece a ligação entre os filtros e a *gateway*.
A DMZ é uma "zona de ninguém" – não pode ser considerada uma rede do perímetro protegido porque parte das suas componentes podem ser comprometidas; e não é uma rede exterior porque é controlada pela organização que se pretende defender com a *firewall*.
A denominação de DMZ é atribuída a quaisquer redes da organização onde se colocam servidores da mesma que tenham de ser contactados do exterior e não possuam sistemas ou aplicações confiáveis.
Pode possuir uma única *firewall* chamada de Bastião.



8.3. Tipos (Modos de intervenção)

No livro de Cheswick e Bellovin são identificados três tipos de *firewalls* segundo um modelo de intervenção: filtro de datagramas, filtro aplicacional e filtro de circuitos.

Packet Filters (Filtro de datagramas)

Um filtro de datagramas é um filtro que atua fundamentalmente ao nível rede, nomeadamente ao nível da troca de datagramas IP. Estes filtros normalmente limitam-se a aceitar ou rejeitar a passagem de um datagrama pela *firewall*, no âmbito do seu encaminhamento através da mesma. Estas *firewalls* não lidam com protocolos aplicacionais e portanto não podem ter qualquer controlo sobre os dados trocados no âmbito dos mesmos.

- Exemplos de filtragem

Rejeitam interações não autorizadas segundo o conteúdo dos pacotes IP

- Endereços IP (de origem ou destino): o controlo dos endereços de origem permite autorizar ou negar fluxos de informação entre máquinas, independentemente dos protocolos de transporte usados.
- Protocolos e portos de transporte (de origem ou destino): o controlo dos protocolos de transporte permite autorizar ou negar completamente certos tipos de protocolos de transporte (UDP, TCP). O controlo destina-se fundamentalmente a ajudar a tomar decisões ao nível da autorização ou negação de interações envolvendo certos tipos de portos de transporte.
- Operação ICMP e dados anexos: as operações ICMP destinam-se a auxiliar as tarefas de administração e exploração da pilha de protocolos IP, mas podem ser indesejadas ou revelar aspetos da organização que interesse esconder como a estrutura de rede interna.
- Sentido de criação de circuitos virtuais: o sentido de criação de circuitos virtuais é importante para controlar de que modo os circuitos virtuais podem ser estabelecidos, uma vez que a sua génese é assimétrica, ou seja, existe normalmente um agente passivo (servidor) e um ativo (cliente).

- Limitações

Uma *firewall* deste tipo é relativamente simples de configurar e gerir quando os protocolos aplicacionais usam políticas simples de gestão dos canais de comunicação.

Application-Level Gateways (Filtro aplicacional)

As *firewalls* do tipo filtro aplicacional operam ao nível do protocolo aplicacional. A sua função é mediar a parte ou a totalidade das interações aplicacionais entre interlocutores remotos, localizados em redes interligadas pela *firewall*, de forma a controlar a execução desse mesmo protocolo, Por isso as *firewalls* deste tipo são normalmente concretizadas usando um conjunto de aplicações designadas como mediadores (*proxies*) que executam em máquinas *firewall*.

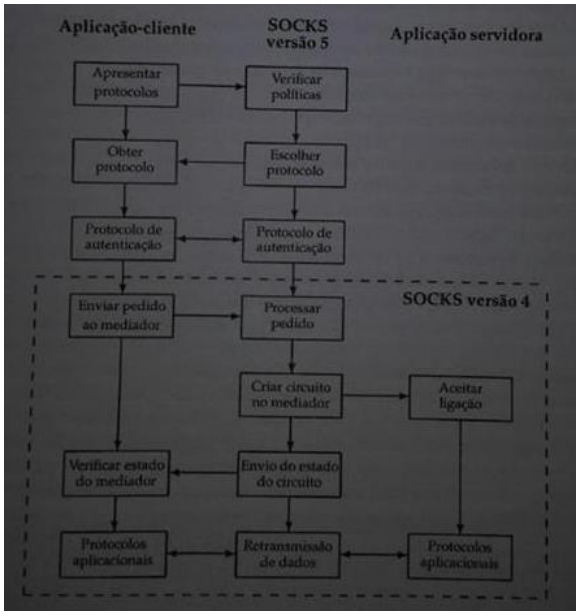
Ao contrario dos filtros dos datagramas, não existem mediadores genéricos. Para cada protocolo aplicacional, é preciso que exista um mediador próprio. Este aspeto faz com que normalmente as *firewalls* usem um filtro de datagramas para controlar a maioria dos fluxos de dados e um conjunto reduzido de mediadores para lidar com alguns fluxos em particular.

A operação de um proxy tem os seguintes aspetos: controlo de aceso por utilizador, análise e alteração de conteúdos, registo (logging) datalhado das operações efetuadas ao nível da aplicação

Circuit Gateway (Filtro de circuitos)

As *firewalls* do tipo filtro de circuitos controlam o estabelecimento de circuitos de formas não acessíveis aos filtros de datagramas, mas sem interferir de forma alguma com o protocolo aplicacional. Um filtro deste tipo, apos autorizar e condicionar o estabelecimento de um circuito virtual, permanece normalmente no meio do mesmo, transferindo octetos entre os dois extremos da comunicação.

Um exemplo de filtros de autorização não transparentes para aplicações-cliente são os baseados no protocolo **SOCKS** (SOCKet Secure). Este protocolo também conhecido como AFT (*Authenticated Firewall Traversal*), permite autorizar e mediar qualquer protocolo aplicacional entre um cliente e um servidor situado em cada extremo da *firewall*. O servidor SOCKS, que medeia interações entre clientes e servidores, é visto no exterior como cliente do serviço, escondendo-lhe o IP de origem do cliente original. O SOCKS pode ser usado de forma inversa, ou seja, servir para permitir determinados acessos do exterior para máquinas do perímetro protegido. As aplicações modificadas para se comportarem como um cliente SOCKS dizem-se "socksificadas". As aplicações deste tipo são os navegadores http.



Stateful Packet Filter		
Realiza Stateful Packet Inspection (PSI) que analisa pacotes completamente incluindo o seu contexto (interações passadas relacionadas) determinando e caracterizando a aplicação em causa e aplicam regras de filtragem/limitação. - Exemplos de contexto		
Ligações TCP estabelecidas	Interações pedido/resposta sobre UDP	Mensagens de erro ICMP a pacotes TCP/UDP antes enviados
- Os pedidos de estabelecimento de ligações são controlados. - As ligações estabelecidas são permitidos	- Autorização dinâmica de respostas - Exemplo: resolução de nomes DNS	- Para prevenir ataques por inundação com pacotes ICMP
Tabelas NAT dinâmicas	Identificação de protocolos aplicacionais através do fluxo de dados	
- Criação de entradas consoante o trafego observado	- Para lidar com fluxos que usem portos dinâmicos ou "roubados" - Exemplos: FTP, protocolos RCP, protocolos P2P - Utilidade: filtragem, transparente proxying, QoS	

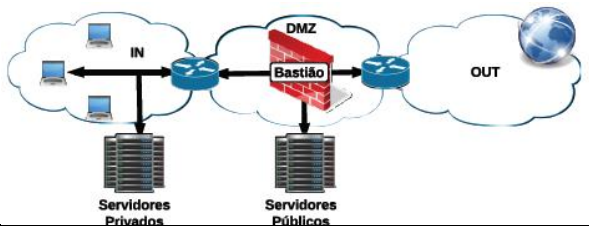
8.4. Bastião

Deve executar versões seguras de sistemas operativos com uma configuração segura tendo instalados apenas os serviços considerados essenciais como *Proxy* de *Telnet*, DNS, FTP, SMTP e autenticação.

Em geral é uma plataforma para *application-level gateways* mas quanto mais proxies houverem no bastião, menor será o seu desempenho. Os proxies podem ser executados em appliances específicas. O bastião apenas encaminha trafego para as appliances apropriadas. Este executa os *application-level gateways* de forma segura, ou seja, independente do comprometimento de um não afeta os restantes e sem privilégios especiais em que o seu comprometimento não permite afetar a máquina.

Os servidores públicos não devem ser colocados num bastião, como por exemplo: DNS, SMTP, HTTP, FTP, SSH, RAS, etc. Devem executar em máquinas desntro de DMZs. Assim, o bastião apenas encaminha trafego para a maquina apropriada dentro de uma DMZ.

8.5. Topologias Elementares

Gateway Simples (*Dual-homed gateway*)	
Arquitetura	**Vantagens**
Uma única máquina – *gateway* bastião  	- Simplicidade - Economia de recursos
	Problemas/Desvantagens
	- O comprometimento da máquina desativa a firewall - A carga de processamento da firewall está toda sobre uma única máquina - Os serviços públicos estão dentro da rede protegida

8.6. Serviços de Segurança

Autorização	Redireccionamento de Tráfefo
- De fluxos de dados (Packet Filtering) -- Nivel transporte ou rede - De utentes (App-Level/Circuit-Level)	- Para máquinas dedicadas (mail, www, ftp, etc) - Proxying (explicito ou transparente)
Processamento de conteúdos	**Comunicação segura**
- Alteração de protocolos de alto nível: -- Aplicação, etc. - Analise de conteúdos	- Virtual Private Networks (VPNs) -- Cifra e controlo de integridade de fluxos de dados sobre redes públicas (inseguras) - Encapsulamento
Defesa contra tentativas de Dos	
- Deteção de ataques - Filtragem de datagramas perigosos - Acionamento de medidas paliativas	

8.7. Limitações das Firewalls

Não resolvem o problema dos atacantes dentro da rede interna a menos que a rede possua firewalls internas. Só são eficazes se controlarem totalmente as ligações entre domínios onde podem ser feitas paralelamente de inúmeras maneiras. São difíceis de administrar em ambientes com interesses heterogéneos como universidades, ISPs.

8.8. Firewalls pessoais

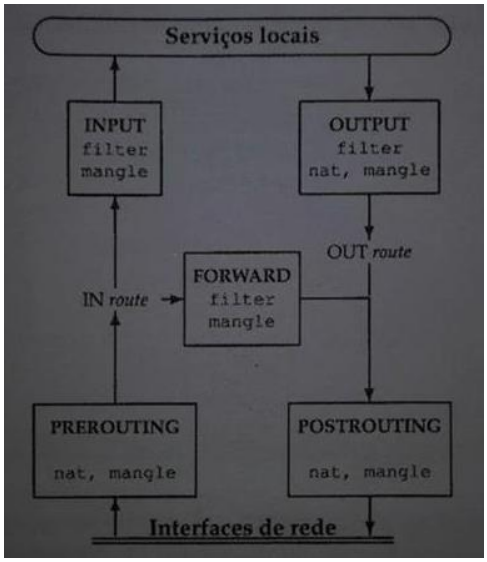
As *firewalls* pessoais não são mais do que *firewalls* que se destinam a proteger uma única maquina e fazem parte do sistema da mesma. Uma *firewalls* pessoal normalmente é um sistema de software que executa na mesma máquina que se quer proteger, ou seja, a *frewall* e o perímetro protegido são exatamente a mesma máquina. Permite controlar aspetos interessantes que são impossíveis para as demais em que as aplicações estão ou não autorizadas a efetuar determinada comunicação. Permite minimizar o comprometido de máquinas alheias no mesmo perímetro de segurança.

- **Problemas**

Nem todos os utentes são especialistas em segurança em redes, pois não sabem nada de protocolos de comunicação e não sabem também como devem forçar um nível de privilégio mínimo. A variedade de interações remotas leva a um grande número de regras onde existes ambientes de trabalho distinto, onde existem diferentes requisitos de segurança, tratamento uniforme versus diferenciado de múltiplas interfaces de rede e onde a confusão e as incoerências são vulnerabilidades difíceis de detetar.

Caso de estudo: *iptables*

O *iptable* é um filtro de pacotes integrado com uma cadeira de processamento de pacotes IP dentro do sistema operativo Linux. Passível a ser estendido de várias maneiras em módulos do sistema operativo e aplicações em modo utilizador.



O *iptables* usa um conceito de cadeias (chains) para analisar datagramas. Uma cadeira é uma sequencia de regras e cada regra possui zero ou mais condições de aplicabilidade e uma decisão. O *iptables* possui cinco cadeias padrão: INPUT (aplica-se a datagramas recebidos pela máquina e que lhe são dirigidos), OUTPUT (aplica-se a datagramas enviados pela máquina e com origem na mesma), FORWARD (aplica-se a datagramas recebidos pela máquina mas que não lhe são dirigidos, ou seja, que passam em transito pela máquina que faz o seu encaminhamento), PREROUTING (aplica-se a todos os datagramas recebidos pela máquina) e POSROUTING (aplica-se a todos os datagramas enviados pela máquina).

O *iptables* usa tabelas para subdividir a aplicação de regras em cada cadeira para agrupar modelos de operação e dependem do modo como o *iptables* foi criado e instalado. Existem três tabelas base: filter(existe sempre por omissão e serve para filtrar datagramas, ou seja, para decidir apenas sobre a sua aceitação ou rejeição), nat (serve para detetar e atuar em situações em que seja necessário fazer NAT) e mangle(serve para efetuar diversos tipos de alterações nos datagramas).

A decisão (target) expressa por cada regra é uma decisão-padrão ou o nome de outra cadeira, porque a decisão deverá ser tomada pelas regras dessa cadeia.

As decisões base são: ACCEPT(indica que o datagrama deve ser aceite), DROP(deve ser descartado), QUEUE(o datagrama deve ser enviado para uma fila de espera destinada a uma aplicação local) e RETURN(indica que a cadeia atual deve ser abandonada e retomada a análise de regras na regra seguinte da cadeia anterior).

Vantagens	Desvantagens
- o facto de ser um produto comparável em eficácia as demais *firewalls* comerciais, - ser apenas o núcleo de uma arquitetura mais complexa e extensível - ser relativamente estável, confiável e escalável - ser económico em termos de recursos computacionais necessários	- o facto de se ter de perceber bem como funciona a interação entre o núcleo LINUX, o *iptables* e diversos outros módulos que interagem com os dois anteriores. - não ser uma solução chave na mão - falta de ferramentas gráficas adequadas aos administradores menos habituados à administração de máquinas LINUX