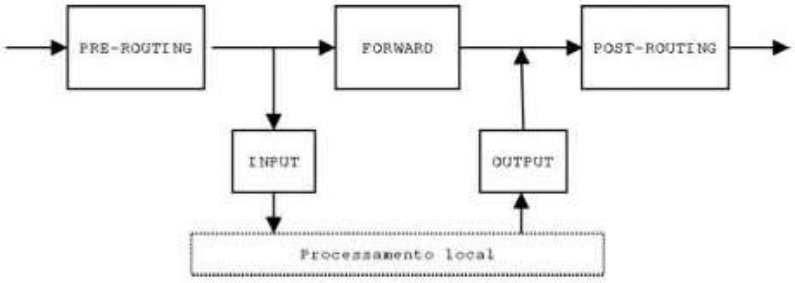
Firewall

Os primeiros firewalls usavam a filtragem de pacote somente para proteger a rede interna de usuários externos. O firewall verificava o cabeçalho de cada pacote que entra na rede interna e tomava a decisão de permitir ou bloquear o pacote baseado no ip usado e o numero da porta especificado no cabeçalho, na parte de tcp ou udp.

Filtragem de pacotes é o bloqueio ou liberação da passagem de pacotes de dados de maneira seletiva, conforme eles atravessam a interface de rede. Em sistemas linux, por exemplo, a filtragem de pacotes é implementada diretamente no kernel. Esses filtros inspecionam os pacotes com base nos cabeçalhos de transporte, rede ou até mesmo enlace. Os critérios mais utilizados são os endereços ip e portas tcp/udp de origem e destino.

Alguns filtros de pacotes também são capazes de transformar o conteúdo dos pacotes, como é o caso do netfilter do linux. Normalmente as pessoas se referem ao filtro de pacotes do linux pelo nome de iptables. Na verdade, o iptables é um utilitário de linha de comando a partir do qual podem ser configuradas as regras de filtragem do netfilter.

O netfilter é formado por um conjunto de cadeias. Cada cadeia é um ponto no caminho que um pacote ip percorre ao entrar ou sair de uma máquina. A figura mostra as cadeias do netfilter.



Existem dois tipos de políticas aplicáveis aos pacotes filtrados:

Permissivo – aceita tudo que não é expressamente proibido.

Restritivo – nega tudo e só encaminha o que for expressamente permitido.

A filtragem de pacotes ip é normalmente feita usando um roteador de filtragem de pacotes, quando tais pacotes passam pelo roteador. Normalmete, um roteador de filtragem de pacotes pode filtrar pacotes baseados em alguns ou todos os campos abaixo:

Endereço ip de origem;

Endereço ip de destino;

Portas tcp/udp de origem;

Portas tcp/udp de destino.

Estes sistemas analisam individualmente os pacotes à medida em que estes são transmitidos da camada de enlace (camada 2 do modelo iso/osi) para a camada de rede (camada 3 do modelo iso/osi).

A principal desvantagem desse tipo de tecnologia é a falta de controle de estado do pacote, o que permite que agentes maliciosos possam produzir pacotes simulados (ip spoofing) para serem injetados na sessão.

Filtragem por política é uma forma diferente de se escrever um conjunto de regras de filtragem. Uma política é definida, a qual configura as regras para quais tipos de tráfego são permitidos e quais tipos são bloqueados. Os pacotes são então classificados, baseando-se no critério tradicional de endereço ip/porta de origem/destino, protocolo, etc.

Firewall

Firewall é um sistema de proteção de redes internas contra acessos não autorizados originados de uma rede não confiável (internet), ao mesmo tempo que permite o acesso controlado da rede interna à internet. Eles podem ser um hardware e/ou software, tendo diferentes tipos de proteção como: pacotes, e-mail, web, etc.

Apesar de se tratar de um conceito geralmente relacionado a proteção contra invasões, o firewall não possui capacidade de analisar toda a extensão do protocolo, ficando geralmente restrito ao nível 4, de transporte, da camada osi.

A complexidade de instalação depende do tamanho da rede, da política de segurança, da quantidade de regras que controlam o fluxo de entrada e saída de informações e do grau de segurança desejado.

A análise da arquitetura de implementação de firewalls traz os conceito de bastion host e dmz. Bastion hosts são servidores cuidadosamente implementados e de alta segurança que mantém contato com a rede externa, consequentemente estando expostos aos riscos de ataques.

Algumas de suas características são:

Todo tráfego entre a rede interna e a externa (entrada e saída) deve passar pelo firewall;

Somente o tráfego autorizado passará pelo firewall, todo o resto será bloqueado;

O firewall em si deve ser seguro e impenetrável;

Tipos de controles realizados:

Controle de serviço: determina quais serviços internet (tipos) estarão disponíveis para acesso;

Controle de sentido: determina o sentido de fluxo no qual serviços podem ser iniciados;

Controle de usuário: controla o acesso baseado em qual usuário está requerendo (tipicamente os internos, ou externo via vpn);

Controle de comportamento: controla como cada serviço pode ser usado (ex: anti-spam);

Proxy firewall / gateways de aplicação

Os conceitos de gateways de aplicação (application-level gateways) e “bastion hosts” foram introduzidos por marcus ranum em 1995. Trabalhando como uma espécie de eclusa, o firewall de proxy trabalha recebendo o fluxo de conexão, tratando as requisições como se fossem uma aplicação e originando um novo pedido sob a responsabilidade do mesmo firewall (non-transparent proxy) para o servidor de destino. A resposta para o pedido é recebida pelo firewall e analisada antes de ser entregue para o solicitante original.

Embora os firewalls de filtro de pacotes e statefull apresentem uma diferença em como pacotes de uma determinada conexão são tratados, ambos se baseiam fundamentalmente nas informações do cabeçalho dos protocolos da camada de transporte.

Um application gateway é um firewall stateful capaz de inspecionar os dados da camada de aplicação para tomar decisões mais inteligentes sobre a conexão. Exemplos de controles realizados por um application gateway são autenticação, filtros de url e filtros de conteúdo.

Um application gateway pode ser utilizado para bloquear, por exemplo, aplicações peer to peer (emule, kaaza etc.), ou mensageiros instantâneos (irc, msn etc.) Que tentam se esconder debaixo do protocolo http. No entanto, os application gateway não são capazes de inspecionar dados criptografados via ssl, por exemplo.

Os gateways de aplicações conectam as redes corporativas à internet através de estações seguras (chamadas de bastion hosts) rodando aplicativos especializados para tratar e filtrar os dados (os proxy firewalls). Estes gateways, ao receberem as requisições de acesso dos usuários e realizarem uma segunda conexão externa para receber estes dados, acabam por esconder a identidade dos usuários nestas requisições externas, oferecendo uma proteção adicional contra a ação dos crackers.

Firewall statefull

Um firewall de filtro de pacotes é dito um firewall sem estado. Isso porque ele trata cada um dos pacotes que atravessam a interface de forma independente.

As conexões tcp são caracterizadas por uma série de atributos como ip's de origem e destino, portas de origem de destino, números de sequência etc. Em conjunto, esses atributos determinam o estado de uma conexão tcp.

Ao contrário dos firewalls de filtro de pacotes, um firewal stateful não filtra pacotes de forma isolada, mas sim com base em informações sobre o estado de conexões pré-estabelecidas. Para um firewall stateful, a comunicação bi-direcional é implícita, de forma que não há necessidade de se escrever regras de filtragem para cada um dos sentidos. A seguir são mostrados alguns exemplos de ragras para um firewall stateful utilizando a sintaxe do iptables.

Com a explosão do comércio eletrônico, percebeu-se que mesmo a última tecnologia em filtragem de pacotes para tcp/ip poderia não ser tão efetiva quanto se esperava. Com todos os investimentos dispendidos em tecnologia de stateful firewalls, os ataques continuavam a prosperar de forma avassaladora. Somente a filtragem dos pacotes de rede não era mais suficiente. Os ataques passaram a se concentrar nas características (e vulnerabilidades) específicas de cada aplicação. Percebeu-se que havia a necessidade de desenvolver um novo método que pudesse analisar as particularidades de cada protocolo e tomar decisões que pudessem evitar ataques maliciosos contra uma rede.

Se comparado com o modelo tradicional de firewall, orientado a redes de dados, o firewall de aplicação é frequentemente instalado junto à plataforma da aplicação, atuando como uma espécie de procurador para o acesso ao servidor (proxy).

Este tipo de firewall conta com o stateful inspection para inspecionar pacotes e tráfego de dados baseado nas características de cada aplicação, nas informações associadas a todas as 7 camadas do modelo osi (e não apenas na camada de rede ou de aplicação) e no estado das conexões e sessões ativas

Um firewall é um portal de filtragem da saída de rede e é efetivo apenas em pacotes que devem passar por ele. Portanto, o firewall só será eficaz quando a única rota para estes pacotes for através dele.

A falta de uma configuração padrão (e do lema "processo, e não produto") explica a falta de uma solução chave. Existem, no entanto, ferramentas que simplificam a configuração do firewall netfilter, com uma representação gráfica das regras de filtragem. Fwbuilder está, sem dúvida, entre os melhores.

Caso específico firewall local

Um firewall pode ser restrito a uma determinada máquina (em oposição a uma rede completa), caso em que seu papel é o de filtrar ou restringir o acesso a alguns serviços, ou possivelmente para evitar que as conexões de saída por softwares maliciosos que um usuário poderia, por vontade própria ou não, ter instalado.

O kernel do linux incorpora o firewall netfilter. Ele pode ser controlado a partir do espaço do usuário com os comandos iptables e ip6tables. A diferença entre estes dois comandos é que o primeiro atua sobre rede ipv4, enquanto que o último sobre o ipv6. Uma vez que ambas pilhas de protocolo de rede provavelmente estarão circulando por muitos anos, ambas as ferramentas serão utilizadas em paralelo.

14.2.1. Funcionamento do netfilter

Netfilter utiliza quatro tabelas distintas que armazenam regras que regulam três tipos de operações sobre pacotes:

Filtro se ocupa das regras de filtragem (aceitando, recusando ou ignorando um pacote);

Nat diz respeito a tradução de endereços e portas de origem ou destino de pacotes;

Mangle diz respeito a outras alterações nos pacotes ip (incluindo campos e opções de tos - tipo de serviço);

Raw permite outras modificações manuais em pacotes antes deles chegarem ao sistema de rastreamento de conexões.

Cada tabela contém listas de regras chamadas cadeias. O firewall usa cadeias padrão para lidar com pacotes com base em circunstâncias pré-definidas. O administrador pode criar outras cadeias, que só serão usadas quando referenciadas por uma das cadeias padrão (tanto direta quanto indiretamente).

A tabela filter (filtro) possui três cadeias padrão:

Input (entrada): lida com os pacotes cujo destino é o próprio firewall;

Output (saída): lida com os pacotes emitidos pelo firewall;

Forward (repassar): lida com os pacotes em trânsito através do firewall (que não é nem a sua origem nem o seu destino).

A tabela nat também tem três cadeias de padrão:

Prerouting (pré roteamento): altera pacotes assim que eles chegam;

Postrouting (pós roteamento): altera pacotes quando eles estão prontos para seguir seu caminho;

Output (saída): altera pacotes gerados pelo próprio firewall.

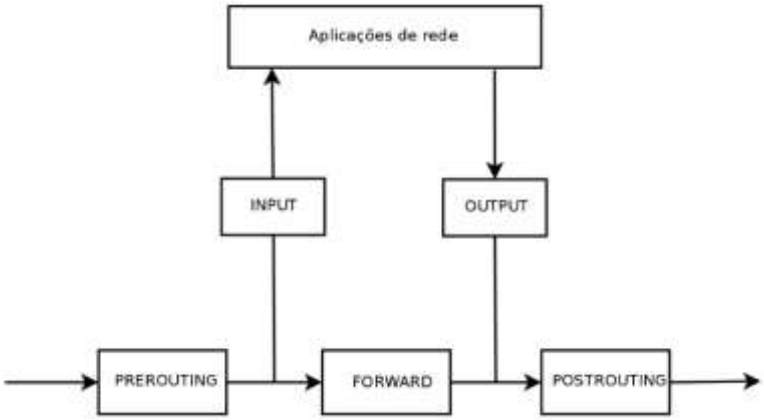


Figura 14.1. Como cadeias netfilter são chamadas

Cada cadeia é uma lista de regras, cada regra é um conjunto de condições e uma ação a ser executada quando as condições forem satisfeitas. Ao processar um pacote, o firewall examina a correspondente, uma regra após a outra; quando as condições para uma regra são satisfeitas, "pula" (daí a opção -j , de jump, nos comandos) para a especificada ação para continuar o processamento. Os comportamentos mais comuns são padronizados, e existem ações específicas para eles. Fazer uma destas ações padrão interrompe o processamento da cadeia, já que o destino do pacote já está selado (salvo uma exceção mencionada a seguir):

De volta ao básico icmp

Icmp (internet control message protocol) é o protocolo usado para transmitir informações complementares sobre as comunicações. Permite testar a conectividade de rede com o comando ping (que envia uma mensagem icmp echo request (solicitação de eco), que o destinatário deve responder com uma mensagem icmp echo reply) (resposta echo). Ele sinaliza quando um firewall está rejeitando um pacote, indica um estouro de memoria no buffer de recebimento, propõe uma melhor rota para os pacotes seguintes na conexão, e assim por diante. Este protocolo é definido por vários documentos rfc, o inicial rfc777 e rfc792 logo foram concluídos e ampliados.

Para referência, um buffer de recepção é uma pequena região de memória para armazenamento de dados entre o tempo que chega na rede e o tempo que o kernel o manipula. Se esta região está cheia, os novos dados não podem ser recebidos, e o icmp sinaliza o problema, de modo que o emissor possa diminuir a sua taxa de transferência (que deve, idealmente, chegar a um equilíbrio após algum tempo).

Observe que, embora uma rede ipv4 possa funcionar sem icmp, icmpv6 é estritamente necessário para uma rede ipv6, uma vez que combina várias funções que eram, no mundo ipv4, espalhados por icmpv4, igmp (internet group membership protocol) e arp (address resolution protocol). Icmpv6 é definido na rfc4443.

Accept: permite que o pacote siga seu caminho;

Reject: rejeita o pacote com um erro icmp (a opção --reject-withtipo para iptables permite seleccionar o tipo de erro);

Drop: apaga (ignora) o pacote;

Log: loga (via syslogd) uma mensagem com uma descrição do pacote; observe que esta ação não interrompe o processamento, e a execução da cadeia continua na próxima regra, razão pela qual logar pacotes recusados exige tanto uma regra log quando uma regra reject/drop;

Ulog: loga uma mensagem via ulogd, que pode ser melhor adaptado e mais eficiente que o syslogd para lidar com um grande número de mensagens; observe que esta ação, como log, também retorna o processamento para a próxima regra na cadeia chamada;

Chain_name: vai para a cadeia dada e processa as suas regras;

Return: interrompe o processamento da cadeia atual, e volta para a cadeia chamada; no caso da cadeia atual ser uma das padrão, não há nenhuma cadeia de chamada, de modo que a ação padrão (definida com a opção -p para o iptables) é executada em vez disto;

Snat (apenas na tabela nat): aplica source nat (opções extras descrevem as alterações exatas para aplicar);

Dnat (apenas na tabela nat): aplica destination nat (opções extras descrevem as alterações exatas para aplicar);

Masquerade (apenas na tabela nat): aplica masquerading (um caso especial de source nat);

Redirect (apenas na tabela nat): redireciona um pacote para uma determinada porta do próprio firewall; isto pode ser usado para configurar um proxy web transparente que funciona sem nenhuma configuração no lado do cliente, uma vez que o cliente pensa que ele se conecta ao destinatário mas na verdade as comunicações passam pelo proxy.

Outras ações, particularmente as relativas à tabela mangle, estão fora do escopo deste texto. O iptables(8) e ip6tables(8) tem um lista completa.

Sintaxe do iptables e do ip6tables

Os comandos iptables e ip6tables permitem manipulação de tabelas, cadeias e regras. Sua opção -t tabela indica em qual tabela operar (por padrão, na filter).

Comandos

A opção -n cadeia cria uma nova cadeia. A -x cadeia exclui uma cadeia vazia e sem uso. A -a cadeia regra adiciona uma regra no final da cadeia dada. A opção -i cadeia número_regra regra insere uma regra antes da regra número número_regra. A opção -d cadeia número_regra (ou -d cadeia regra) remove uma regra na cadeia; a primeira sintaxe identifica a regra a ser removida pelo seu número, enquanto a segunda a identifica pelo seu conteúdo. A opção -f cadeia esvazia uma cadeia (remove todas suas regras); se nenhuma cadeia é mencionada, todas as regras da tabela são removidas. A opção -l cadeia lista as regras na cadeia. Finalmente, a opção -p cadeia ação define a ação padrão, ou "política", para uma dada cadeia; observe que apenas as cadeias padrão podem ter essa política.

Regras

Cada regra é expressa como condições -j ação opções_ações. Se várias condições são descritas na mesma regra, então o critério é a conjunção (e lógico) das condições, que é pelo menos tão restritiva quanto cada condição individual.

A condição -p protocolo corresponde ao campo protocolo do pacote ip. Os valores mais comuns são tcp, udp, icmp, e icmpv6. Prefixar a condição com um ponto de exclamação nega a condição, que se transforma numa correspondência para "todos os pacotes com um protocolo diferente do especificado". Este mecanismo de negação não é específico para a opção -p e também pode ser aplicada a todas outras condições.

A condição -s endereço ou -s rede/máscara corresponde ao endereço de origem do pacote. Do mesmo modo, -d endereço ou -d rede/máscara corresponde ao endereço de destino.

A condição -i interface seleciona os pacotes entrando pela dada interface. -o interface seleciona pacotes saindo de uma interface específica.

Existem condições mais específicas, dependendo das condições genéricas acima descritas. Por exemplo, a condição -p tcp pode ser complementada com condições sobre as portas tcp, com cláusulas como --source-port porta (porta de origem) e --destination-port porta (porta de destino).

A condição --state estado corresponde ao estado de um pacote em uma conexão (isto requer o módulo ipt_conntrack do kernel, para rastreamento de conexões). O estado new descreve um pacote iniciando uma nova conexão; o estado established corresponde aos pacotes pertencentes a uma conexão já existente, e relatedcorrespondem aos pacotes iniciando uma nova conexão relacionada a uma já existente (o que é útil para as conexões ftp-data no modo ativo do protocolo ftp).

A seção anterior lista as ações disponíveis, mas não suas respectivas opções. A ação log, por exemplo, tem as seguintes opções:

--log-level, com valor padrão warning (aviso), indica o nível de severidade no syslog;

--log-prefix permite especificar um prefixo de texto para diferenciar mensagens registradas;

--log-tcp-sequence, --log-tcp-options e --log-ip-options indicam dados extras a serem integrados na mensagem: respectivamente, o número de seqüência tcp, opções tcp, e as opções ip.

A ação dnat fornece a opção --to-destination endereço:porta para indicar o novo endereço ip de destino e/ou porta. Da mesma forma, snat fornece --to-source endereço:porta para indicar o novo endereço e/ou porta ip de origem.

A ação redirect (disponível apenas se o nat está disponível) fornece a opção --to-ports porta(s) para indicar a porta, ou intervalo de portas, para onde os pacotes devem ser redirecionados.

Criando regras

Cada criação de regra exige uma invocação de iptables/ip6tables. Digitar estes comandos manualmente pode ser tedioso, por isso as chamadas são normalmente armazenados em um script para que a mesma configuração seja criada automaticamente a cada vez que a máquina inicia. Este script pode ser escrito à mão, mas também pode ser interessante prepará-lo com uma ferramenta de alto nível, como fwbuilder.

apt install fwbuilder

O princípio é simples. Na primeira etapa, é preciso descrever todos os elementos que estarão envolvidos nas regras reais:

O próprio firewall, com suas interfaces de rede;

As redes, com suas faixas de ip correspondentes;

Os servidores;

As portas que pertencem aos serviços hospedados nos servidores.

As regras são então criadas com simples ações de arrastar-e-soltar nos objetos. Alguns menus contextuais podem alterar a condição (negando-a, por exemplo). Em seguida, a ação deve ser escolhida e configurada.

Quando ipv6 está ativo, pode se criar dois conjuntos de regras distintas para ipv4 e ipv6, ou criar uma só e deixar o fwbuilder traduzir as regras de acordo com os endereços atribuídos aos objetos.

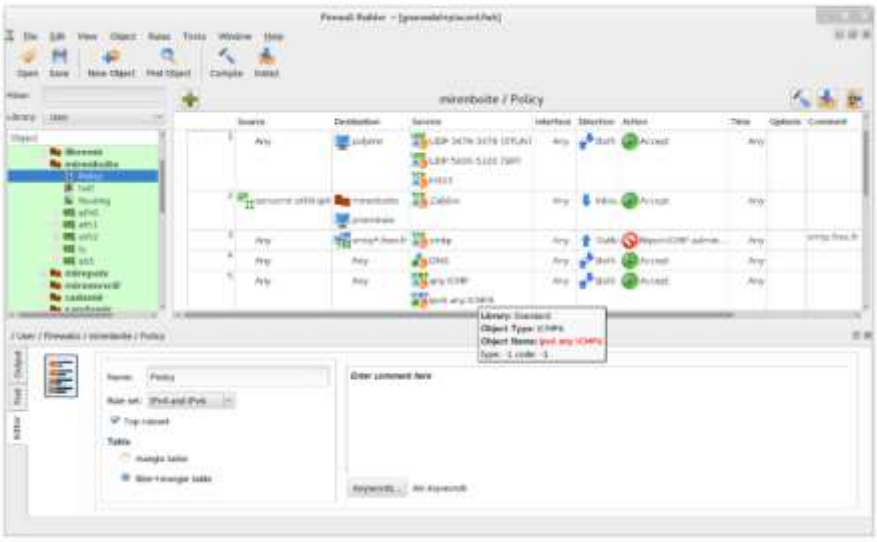


Figura 14.2. Janela principal do fwbuilder

Fwbuilder pode gerar um script de configuração do firewall de acordo com as regras que foram definidas. Sua arquitetura modular lhe confere a capacidade de gerar scripts que visam diferentes sistemas (iptables para linux, ipfpara o freebsd e pf para openbsd).

Instalando as regras em cada inicialização

Em outros casos, a maneira recomendada é registrar o script de configuração em uma directiva up do /etc/network/interfaces. No exemplo a seguir, o script é armazenado em /usr/local/etc/arrakis.fw.

Exemplo 14.1. Arquivo interfaces chamando script firewall

Auto eth0

Iface eth0 inet static

Address 192.168.0.1

Network 192.168.0.0

Netmask 255.255.255.0

Broadcast 192.168.0.255

Up /usr/local/etc/arrakis.fw

Isso obviamente assume que você está usando o ifupdown para configurar as interfaces de rede. Se você está usando algo diferente (como o networkmanager ou systemd-networkd), então consulte suas respectivas documentações para encontrar maneiras de executar um script após a interface ter sido levantada.

Filtragem no pfsense

A necessidade de investir em tecnologias que garantam mais segurança ao sistema que a empresa utiliza bateu à sua porta?

Segurança nunca é demais, sobretudo nos tempos atuais, e é justamente por isso que vamos apresentar, neste artigo, o que é pfsense e como essa poderosa ferramenta mudará o seu jeito de pensar a respeito de firewalls e roteadores.

O que é pfsense?

Essa deve ser a sua maior dúvida no momento, certo? A definição que christopher m. Buechler, um dos idealizadores e criadores do pfsense ao lado de scott ullrich, serve muito bem para responder a esta questão: “pfsense é uma distribuição customizada, livre e open source (código aberto), do projeto freebsd criado para ser utilizado como um firewall ou roteador, inteiramente gerido em uma interface web fácil de usar”.

Em outras palavras, o pfsense é uma robusta solução de firewall e/ou roteador amplamente utilizada hoje por empresas e usuários avançados (mais de 1 milhão de downloads foram feitos desde o seu lançamento). Por ser open source, consolidou-se como uma grande concorrente das principais soluções pagas disponíveis no mercado.

Quais são as principais vantagens e os recursos do pfsense?

Primeiramente, uma das principais vantagens é a sua licença bsd — licença de código aberto, gratuita, utilizada em sistemas baseados em unix. Esse tipo de licença permite que o pfsense seja customizado de acordo com as maiores necessidades da empresa.

Um fator que auxilia na customização é a imensa variedade de pacotes de software, muitos deles criados por especialistas da comunidade de desenvolvedores para acrescentar novas funcionalidades.

Na linguagem dos especialistas em segurança da informação, a disponibilização dos pacotes para as mais diversas funções credencia o pfsense como um utm (unified threat management, ou central unificada de gerenciamento de ameaças, em português), que, em breves palavras, pode ser entendido por um dispositivo com diversas funções, tais como:

Firewall;

Servidor (internet, dhcp, ntp, proxy…);

Antivírus;

Antispyware;

Antispam;

Filtragem de conteúdo;

Detecção de intrusão, entre outros.

Com tantas funções primordiais de segurança reunidas em uma única solução, um utm como o pfsense, apesar de gratuito, pode funcionar com excelência equiparável aos mais diversos produtos do mercado.

Além dessas vantagens o pfsense é considerado muito leve, exigindo baixíssimos requisitos de hardware, é estável, fácil de utilizar (possui até um dashboard e uma interface configurável) e possui excelentes recursos de filtragem.

Entretanto, caso a tarefa de fazer do pfsense a sua solução em firewall/roteador por contra própria seja trabalhosa e não muito condizente com o seu nível de conhecimento técnico, existem várias distribuições de firewall (desenvolvidas diretamente do pfsense) já configuradas que incluem suporte completo e, em alguns casos, um appliance (hardware).

Como baixar, instalar e configurar o pfsense?

Presumindo que você seja um usuário do sistema linux, sem dúvidas você usfruirá completamente do pfsense. Afinal, as próprias distribuições do linux são bastante seguras, estáveis e customizáveis. Essa combinação certamente pode resultar em uma poderosa solução de segurança para sua empresa.

Voltando o foco para o pfsense, o primeiro passo para fazer o download da imagem é acessar este link, selecionar o tipo de arquitetura da sua cpu — intel (i386; 32-bit), amd (amd64; 64-bit) ou netgate adi, escolher o formato (platform) e clicar em download.

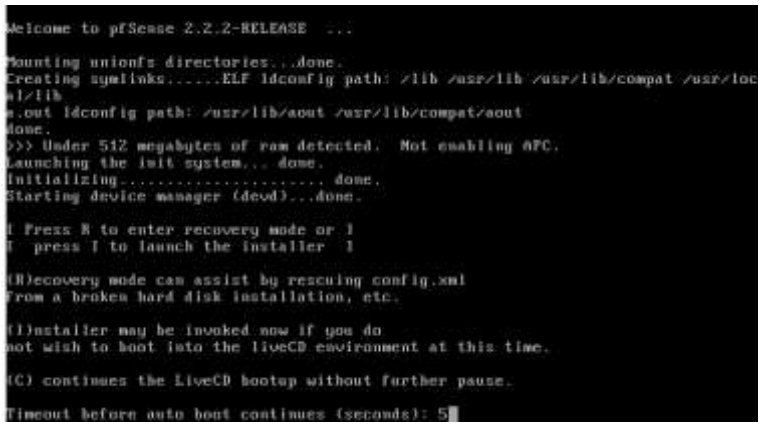
Vale lembrar que o arquivo virá compactado em .gz. Portanto, será necessário ter instalado um programa capaz de fazer a descompactação, como o winrar.

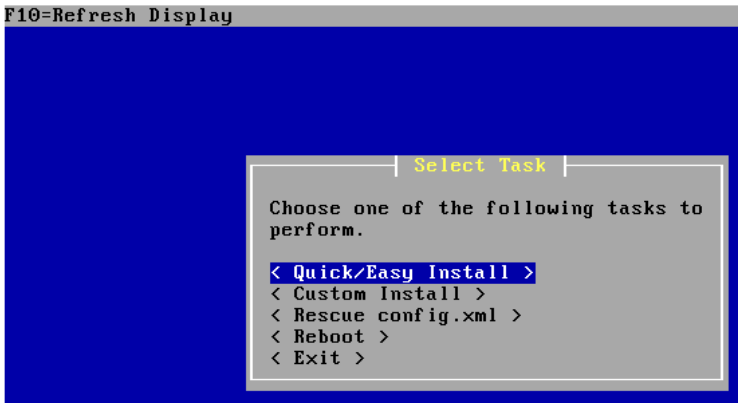
Fazendo a instalação

Quando o arquivo for executado, fique atento à primeira pergunta que será feita logo no boot: “do you want to set up vlan now?”. Traduzindo, o instalador perguntará se você deseja configurar uma vlan (virtual lan). Responda “”não”.

Em seguida, informe as interfaces wan e lan do seu sistema. Caso não saiba como obter essas informações, você poderá optar pela detecção automática do instalador e então os dados serão coletados automaticamente. Feito isso, o instalador questionará sobre a existência de uma interface adicional. Se houver, prossiga registrando-a ou escolha a auto detecção.

Por fim, o sistema pedirá para que essas configurações sejam confirmadas. Ao confirmá-las, as informações básicas de configuração serão exibidas na tela seguidas de um menu de opções.





Configurando o pfsense pela interface web

Por meio de um computador conectado à sua rede lan, você abrirá o browser (navegador) e digitará, na barra de endereço, o ip da interface lan atribuída e exibida anteriormente. Isso fará com que apareça uma tela de login, onde deverão ser digitados os seguintes dados:

Logo após o login será necessário fornecer algumas informações, sendo as mais relevantes:

Hostname: insira o nome do computador no qual o pfsense foi instalado. Isso dispensará futuramente a necessidade de digitar o endereço de ip para acessar a interface web.

Domain: informe o nome de domínio da sua rede (exemplo.com), caso haja.

Dns: nessa tela você poderá registrar o endereços dos servidores dns, podendo, inclusive, determinar qual será o dns primário.

Observação: ao deixar marcada a opção “allow dns server list to be overridden by a dhcp/ppp server on wan…” , os servidores dns serão gerados por meio da porta wan.

General configuration: aqui a sua interface wan será configurada. Para tal, é necessário fornecer informações sobre o provedor de internet (ip estático ou dhcp, autenticação etc.).

Configure lan interface: nessa etapa você apenas deverá informar o endereço da sua rede lan e a máscara de rede.

Set admin webgui password: para finalizar, troque o usuário/senha do administrador.

_____

_____

_____

_____

_____