Um Linux Reduzido para Sistema de Firewall

José de Ribamar Braga Pinheiro Júnior e Sidi Ould Ehmety
Universidade Federal do Maranhão
Departamento de Engenharia de Eletricidade
Centro Tecnológico
Campus Universitário do Bacanga
São Luís 65080-400
MA- Brasil
E-mail: jrbraga@deinf.ufma.br *e* Sidi.Ehmety@cl.cam.ac.uk

Tel: +55 98 217 82 42
Fax: +55 98 217 82 41

RESUMO

A troca de informações sobre a Internet representa uma questão de sobrevivência para os negócios nos dias atuais. Pequenas e médias empresas estão cada vez mais utilizando a Internet como uma forma de expansão. O sistema de Firewall é uma das possibilidades para conseguir proteger as redes destas corporações conectadas, de ataques. O Linux é um sistema operacional gratuito que possibilita a configuração de um firewall. Apresenta-se um sistema de firewall baseado no Linux com o kernel reduzido. Uma metodologia de desenvolvimento, com o formalismo de um modelo de referência e um projeto detalhado baseado em rede de Petri, é utilizada para justificar o sistema definido.

Palavras chaves: Linux, Segurança, Sistema de Firewall, Kernel, Sistema Operacional.

1 Introdução

Esta é a era da informação. A informação passou a ser o bem maior de uma sociedade; quem a possui, detém o poder. A comunicação entre computadores, permitindo a troca de informações, está inclusa no cotidiano de empresas e pessoas. Proteger este bem, constitui-se em uma tarefa árdua, sendo necessárias técnicas especiais que garantam que o sistema computacional esteja seguro. Um computador é seguro se você puder depender dele, de seu software e ele se comportar como você espera [1].

Existem diversas maneiras de proteger um sistema computacional de invasores, dentre elas, os sistemas de monitoramento/detecção e o firewall. O firewall é o mais amplamente usado atualmente [2]. Seu princípio de funcionamento, consiste em colocar um sistema computacional entre uma rede a ser protegida e uma outra considerada não confiável, implementando assim, técnicas que proíbam a intrusão de usuários indesejados e a utilização de serviços de rede não permitidos. Ele é composto de hardware e software que separa uma rede interna de outra considerada não confiável. Para tanto, analisa todo e qualquer pacote que trafega sobre ele e a partir de regras preestabelecidas tomam decisões sobre os mesmos, vide Figura 1.

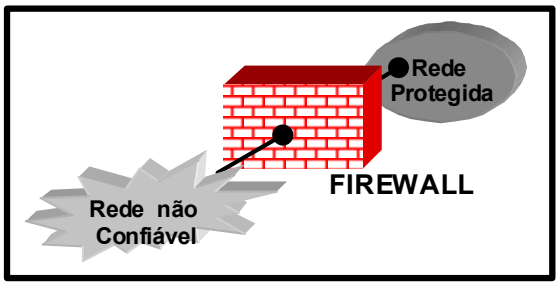

Figura 1 – Sistema de Firewall

O firewall agrupa um conjunto de técnicas como tradução de endereços, serviços de Proxy, Filtragem de pacotes e redes virtuais privadas (VPN - *Virtual Private Network*) que somadas constróem um sistema definido através de uma política de segurança da rede [3]. Existem diversas formas de juntar estes componentes formando o que denominamos arquiteturas de firewall. Elas podem estar presentes em um só equipamento formando as arquiteturas *Single-Box* ou separadas em mais de um equipamento, denominada *Screened Host* [3].

No caso da *Single-box* , uma opção simples e de baixo custo, é a utilização de um computador com mais de uma placa de rede, descrito como *Dual-Homed Host*, que separa duas ou mais redes intermediando as comunicações entre elas. Este tipo de arquitetura possibilita a definição de filtragem de pacotes, proxy, tradução de endereços e VPN´s através de softwares instalados no mesmo. Existe um ponto crucial nesta configuração: o grande esforço do Sistema Operacional do host para executar esta tarefa, podendo ocasionar *overhead*.

Uma opção razoável para a arquitetura *Dual-Homed Host* é o Linux, sendo este um sistema operacional gratuito e de código aberto. O Linux pelo fato de seu código ser distribuído livremente, beneficia-se de características como revisões permanentes, inexistência de funções ocultas, a referência do autor do código e uma grande equipe de programadores[4]. Sistemas seguros requerem software de qualidade que utilize técnicas de código seguro, implementadas de forma consistente de acordo com as políticas e normas de segurança. O problema encontrado no Linux é que a forma que é distribuído não facilita seu funcionamento como firewall.

Este trabalho propõe a definição de um sistema de firewall baseado no sistema operacional GNU/Linux, na sua versão mais atual 2.4, com o kernel reduzido, resultando em uma nova distribuição gratuita deste produto para pequenas e médias empresas. Utiliza-se técnicas de modelagem de sistemas para firewall para justificar o produto final. O presente artigo é organizado da seguinte maneira, no capitulo 2 descreve-se a metodologia utilizada para justificar o projeto, no capitulo 3 é mostrada o seu desenvolvimento.

2 Projeto de um sistema de Firewall

Os problemas envolvidos no desenvolvimento de um sistema de firewall não diferem muito dos outros tipos de sistemas [5]. Neste procura-se afastar os riscos, custos ou a inconveniência da manipulação direta, utilizando a análise, a simulação e a manipulação de um modelo de referência que leva mais facilmente a um novo conhecimento. Desta forma providencia-se um entendimento maior do sistema a ser modelado, uma vez que não se trata de todos os detalhes ao mesmo tempo e pode-se provar os seus benefícios durante a implementação. Com um modelo de referencia defini-se quais funções são necessárias no sistema de firewall, sua garantia, sua interação a nível conceitual e seu benefícios de forma escalonada. Além de um modelo, necessita-se também, de uma metodologia de desenvolvimento.

2.1 O modelo de referência

No presente trabalho utiliza-se o modelo de referência de Christoph L. Scuba [5] para desenvolver e justificar o sistema. O modelo de Schuba é funcional pois focaliza-se nas funcionalidades requeridas pelo sistema de firewall para garantir a política de segurança. Neste modelo os sistema são, a nível conceitual, compostos de componentes funcionais interagindo separadamente combinados sobre certas condições para montar um sistema de firewall.

Na Figura 2 pode ser visualizado o modelo de referência proposto por Schuba, descrito a seguir. O intervalo saída/entrada, representa as redes intermediárias de interligação entre a rede de “a” e de “b”. O domínio da política de proteção da rede é definido no modelo, pelos elementos entre este intervalo e o “b”. Num sistema de comunicação típico as mensagens são divididas em unidades que são transmitidas sobre a rede, elas são compostas de controle e dados, representadas respectivamente por t.ctl e t.dado, que tem seu caminho conceitual definido pela linha mais grossa. A decisão da passagem ou não das unidades de transmissão depende do losango com uma interrogação. Esta é definida através das chamadas as funções de segurança nos blocos FA (Função de Autenticidade), FI (Função de Integridade), FCA (Função de controle de Acesso) representadas pelas linhas tracejadas. Cada uma resulta em uma resposta ‘‘falha” ou “passa”. Se qualquer uma delas falhar, a unidade de transmissão não será repassada para o seu destino, isto é representado pela porta lógica AND na Figura 2. O Faud (Função de Auditagem), registra uma lista ordenada dos eventos significantes definidos pela política de segurança. Todos os outros blocos estão interligados a este, indicando que os mesmos poderão utilizar suas funções caso isto seja definido em suas regras. O bloco FRA (Função de Reforço de Acesso) garante que as funções anteriormente explanadas sejam chamadas se definidas pela política de segurança do domínio. Os componentes presentes em a e b, FA e FI, representam porções das funções de

autenticidade e integridade executadas por estas entidades. Este modelo é aplicado nos dois sentidos da comunicação.

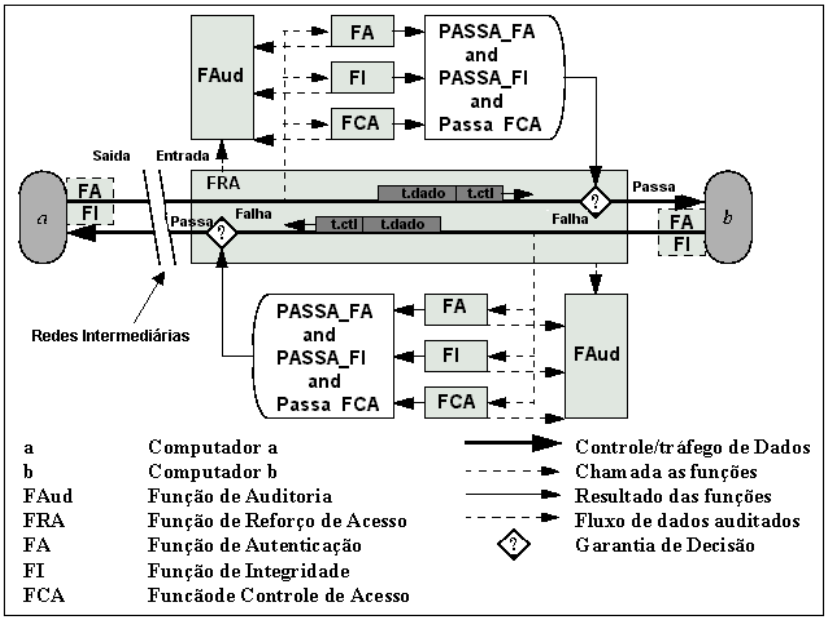

Figura 2 – Modelo de referência de Schuba

2.2 O Ciclo de vida de um projeto

O ciclo de vida proposto por Schuba, possui as seguintes fases: definição da política de segurança, projeto de alto nível, seleção dos componentes de firewall, projeto detalhado e verificação, configuração e implementação, revisão e teste e finalmente, ciclo periódico. A primeira etapa responde as perguntas clássicas para definição de uma política de segurança [3], qual o perímetro da rede a ser protegida, quais os serviços serão disponibilizados para a rede externa, quais os controles necessários para o sistema, entre outras. O projeto de alto nível é a definição dos elementos globais do sistema sem entrar em detalhes, nesta fase aplica-se o modelo de referência. Na fase de seleção dos componentes são escolhidos os elementos que farão parte do sistema de firewall. No Projeto detalhado e verificação, define-se todo o detalhamento do projeto, os procedimentos operacionais, as reconfigurações de rede, etc. Nesta fase utiliza-se o formalismo da rede de petri colorida para definição detalhada do projeto. Durante a fase de configuração e implementação o sistema será feito e configurado podendo ser automatizados por ferramentas. O ciclo periódico é o de revisão permanente do sistema para novas políticas, novos tipos de falhas, etc. A partir de qualquer fase pode-se voltar as suas anteriores .

3 Um Linux Reduzido para Firewall

O Linux é um kernel de um sistema operacional semelhante ao UNIX, desenvolvido pelo finlandês Linux Torvalds, por volta de 1991, até a versão 0.02, quando passou a recrutar esforços pela Internet e após 3 anos atingiu a versão 1.0m [8]. Algumas instituições oferecem o Linux de forma empacotada através de organizações ao redor do mundo, que recebem o nome de distribuições. Uma distribuição é composta pelo kernel (o próprio Linux) e alguns outros softwares, como o shell, por exemplo, que formam um sistema operacional completo adicionado de uma série de aplicativos. Grande parte destes são originados numa versão modificada do sistema GNU que é uma tentativa da OSF (Open Software Fundation) de desenvolver um sistema operacional "unix-like". Daí o nome oficial do Sistema Operacional ser GNU/Linux, que daqui em diante será denominado somente por Linux. Estas distribuições, no entanto, não são geralmente bem adequadas quando são aplicadas a resolver problemas específicos, como um sistema de firewall, por exemplo. Desta forma aplicou-se a metodologia de Schuba para desenvolver um sistema de firewall baseado na arquitetura do sistema operacional Linux, definindo para tanto todos os componentes necessários, e a partir daí, chegar-se a uma definição mínima do mesmo.

A versão utilizada do linux no nosso sistema, possui um pacote de firewall em seu kernel, o netfilter [7]. Este é um aplicativo embutido no kernel para a manipulação de pacotes de rede. Na arquitetura do netfilter existem pontos bem definidos, chamados de hook, a partir dos quais os pacotes atravessam a pilha de protocolos. Em cada um destes pontos o netfilter é chamado, sendo repassado o protocolo e o identificador do hook. Partes do kernel podem ser configuradas a escutar os diferentes hooks para cada protocolo. Quando o pacote é repassado para o netfilter, este verifica se algo foi configurado para aquele e o seu hook, podendo aceitar, destruir, enfileirar, repetir a chamada ou rejeitar enviando a origem, uma mensagem de erro.

3.1 Definição da Política de segurança

Apesar das necessidades das organizações diferirem entre si, a política de segurança foi definida considerando uma instalação típica para empresas conectada a Internet com servidores de FTP, SMTP, POP, HTTP, HTTPS e DNS. Seguem abaixo os requisitos de segurança considerado para o caso em apreço:

- Todos os pacotes serão negados, a não ser que seja explicitamente especificado o contrário
- Bloqueio de todas as conexões iniciadas externamente, exceto os serviços permitidos que serão redirecionados p/ seus respectivos servidores
- Os serviços de FTP, SMTP, POP utilizarão o IP Security para comunicação
- Permissão de tráfego ilimitado a partir da interface de loopback
- Permissão de conexões iniciadas a partir da rede interna
- Pacotes destinados ao firewall serão descartados exceto os da rede interna para o serviço ssh
- A rede interna sofrerá SNAT (Source Network Address Translation) do tipo mascaramento
- Impedirá ataques do tipo:
  - Syn flood
  - IP spoofing
  - Source Routed Options
  - ICMP (redirect e destination unreachable)
  - Entre outros
- Rejeitará pacotes com endereços reservados exceto os definidos em suas interfaces
- Pacotes ICMP terão tamanho máximo limitado quando destinados a rede interna

3.2 Projeto de alto nível

As funções de integridade e autenticidade estão a cargo do IPSec que extende o IP, permitindo uma criptografia fim a fim entre os hosts [3], este provê uma comunicação segura entre os clientes e os diversos servidores no nível de camada de rede. Para tanto é necessário a instalação deste produto nos hosts/roteadores do cliente externo que desejarem uma conexão com os serviços definidos na política de segurança.

As funções de controle de acesso são implementadas através de regras bem definidas nas opções de filtragem e pacotes do sistema de firewall. A lista de controle de acesso será aplicada as camadas físicas (definidas pela tecnologia de rede), de rede e de transporte, indicando o uso das características de cada uma destas camadas para definir as regras. Características como endereço MAC, endereço IP de origem, porta de conexão serão então utilizadas.

3.3 Seleção dos componentes de firewall

O kernel do Linux é modular, por isto foram escolhidos alguns componentes que formam o netfilter para implementar o sistema de firewall, aqui segue o resumo das suas descrições. *Connection tracking* , mantem um registro de quais pacotes tem passado pela máquina para figurar como estão relacionados com suas conexões. *FTP protocol support* ajusta o protocolo FTP para firewall. *IP tables support* , dá suporte ao Iptables que é um *framework* geral e extensivo para implementação de tradução de endereço e filtragem. *Limit match support* permite a definição de controle de uma taxa máxima para as regras que aceitam a passagem dos pacotes. Isto deve ser implementado para resolver alguns tipos de ataque DoS (Deny of Service) [3], indicando que um certo serviço somente aceitará N conexões em um determinado intervalo de tempo. *MAC address match support*, permite definir regras com o endereço do pacote da camada de enlace (sub-Camada de Acesso ao Meio). *Multiple port match support*, consentindo o suporte a regras definidas em uma série de portas de destino ou origem. *Connection state match support,* necessário para eleger pacotes baseado no estado da conexão e seus pacotes anteriores. *Packet filtering,*

define uma tabela (filter) que possui uma série de regras para pacotes locais (entrada e saída) e os que são repassados pelo sistema de firewall. *Full NAT* determina uma tabela (nat) que possui uma série de regras para tradução de endereço. *LOG target support a*diciona um alvo denominado LOG que permite criar regras que registrem o cabeçalho do pacote para o a ferramenta de auditagem syslog [7]

Foram retirados do Kernel, todos os módulos não relacionados com o sistema de firewall proposto, conseguiu-se então uma diminuição de 45% do tamanho do kernel original, incluindo as opções de firewall. Além do kernel foram escolhidos alguns aplicativos de suporte do sistema operacional, como o programa init para inicialização, scripts do tipo rc, biblioteca de suporte a chamada de sistema, arquivos de configuração e de sistemas de arquivos e outros. Para os serviços de rede foram incluídos os programas shell secure e bind para terminal remoto e serviço de nomes, respectivamente. Todo os componentes escolhidos foram empacotados em um disquete de 1.44 MB.

3.4 Projeto detalhado e verificação

Nesta fase, como já foi dito, utilizou-se um formalismo da rede de PETRI colorida para definir o projeto detalhado. A Rede de Petri é uma ferramenta matemática gráfica de modelagem, empregada para sistemas concorrentes [6]. A Rede de Petri colorida é uma extensão para sistemas em que a comunicação, a sincronização e o recurso compartilhando possui um papel importante. Ela reúne o formalismo da rede de Petri e de uma linguagem de alto nível através de uma representação matemática, formal com sintaxe e semântica clara. O termo colorida refere-se ao tipo de dado que pode ser ocupado em um lugar [5].

A Figura 3 mostra uma parte do projeto detalhado utilizando uma rede de Petri. Esta define a direção e o controle da taxa máxima de conexões permitidas, assim como preestabelecido nas políticas de segurança. O controle da taxa máxima é utilizada para evitar alguns tipos de negação de serviço como o syn flooding. Um datagrama vindo de um local anterior passa por uma transição denominada "verifica conexão" que executa sobre este a função verifica_conexão(). Recebendo como parâmetro um datagrama, retorna *rcv* como "falha" ou "passa". Ela verifica a direção da conexão que está sendo realizada; a resposta "passa" será dada às conexões iniciadas no sentido rede interna para rede externa. O datagrama e a resposta da função será então colocados no local "Conexão verificada". Este dispara duas transições que são condicionadas ao *rvc*,. A transição "falha conexão" entrega o datagrama com informações adicionais a um local de auditagem caso rcv="falha". Caso rcv seja "passa" o datagrama é entregue ao local "conexão aceita". A transição "verifica limite" executa a função verifica_limite() que recebe como parâmetro o datagrama e retorna novamente "falha" ou "passa". Ela é parametrizada, internamente, por dois valores: taxa_limite, que indica a taxa média máxima de pacotes por segundos que deverão passar, e limite_máximo, que mede o valor máximo de pacotes que poderão passar antes de taxa_limite estourar. A cada "taxa_limite" intervalo de tempo, uma unidade será retirada do limite_máximo. Sempre que os dois limites estourarem juntos a função retorna "falha". O local "limite verificado" recebe o dado datagrama e a resposta da função anterior iniciando duas transações condicionais que podem repassar o datagrama em diante ou auditá-lo como já foi visto.

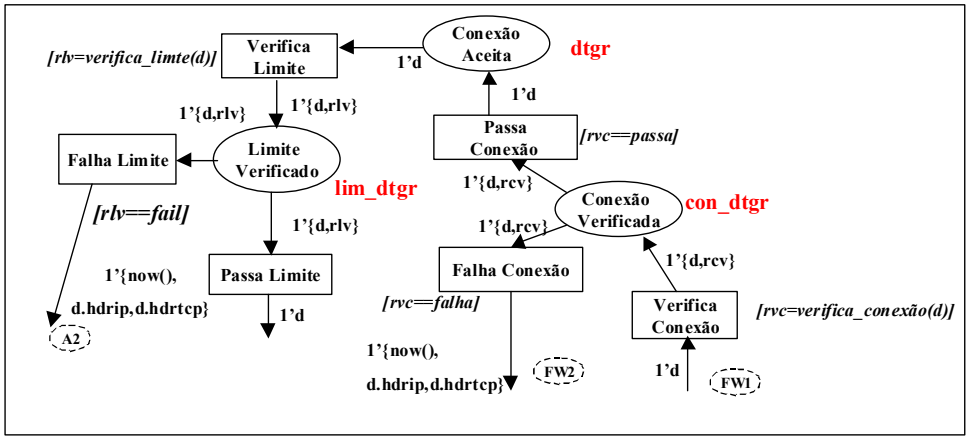

Figura 3 - Parte do projeto detalhado referente ao controle de conexões

3.5 Configuração e implementação

A configuração do firewall do Linux é feita através de um aplicativo do espaço de usuário denominado iptables. O iptables implementa vetores de regras na memória utilizadas para verificar o que cada hook deve fazer com os pacotes que passam por eles. Este gerencia tabelas que indicam quais regras devem ser consideradas quando os pacotes ultrapassam cada um dos pontos de checagem. As tabelas filter, nat e mangle são utilizadas respectivamente para filtragem de pacotes, tradução de endereços de rede (Network Address Translation) e manipulação dos pacotes.

Para a configuração do caso da verificação da conexão ter-se-ia:

Criando um ponto de verificação

/sbin/iptables -N verifica_conexão

Definindo que todos os pacotes tcp com os estados ESTABLISHED,RELATED serão aceitos

/sbin/iptables -A verifica_conexão -m state --state ESTABLISHED,RELATED -j ACCEPT

Aceitando todas os início de conexões exceto aqueles que estiverem entrando pela interface de rede conectada a rede insegura

/sbin/iptables -A verifica_conexão -m state --state NEW -i ! <interface externa> -j ACCEPT

3.6 Revisão e teste

A fase de revisão é, certamente, uma fase importante em qualquer projeto. A partir deste ponto foram revisados alguns tópicos e adicionadas alguma características que não foram contempladas. Dentre as revisões importantes podemos destacar a introdução de uma ferramenta de verificação da integridade dos arquivos denominada AIDE.

O sistema foi aplicado na rede de um departamento da UFMA (Universidade Federal do Maranhão) protegendo cerca de 15 computadores. Para teste utilizou-se algumas ferramentas de auditoria como nmap, retina e SAFESuite e algumas técnicas conhecidas de invasão que não surtiram efeitos.

4 Conclusão

Este artigo apresentou um desenvolvimento de um Linux reduzido para sistema de Firewall. Para justificá-lo utilizou-se uma metodologia de desenvolvimento que pudesse ser utilizada para escolha dos componentes do referido sistema operacional. O modelo de referência utilizado identificou as funções para a feitura do sistema e a metodologia diminuiu os riscos de uma manipulação direta providenciando um entendimento maior do sistema modelado e permitindo uma definição mais formal do problema.

O sistema Linux atualmente representa uma boa opção para pequenas e médias empresas que pretendem integrar-se a rede Internet. O documento presente descreveu o desenvolvimento de um firewall de livre distribuição, baseado neste sistema operacional, mostrando as vantagens da utilização de um formalismo, garantindo a sua transparência e a facilidade de revisão em caso necessidade de melhorias.

Referências

[1] GARFINKEL, Simpson. SPAFFORD, and Gene / **Pratical Unix e Internet Security.** O'Reilly & Associates, Inc. Second Edition, 1996

[2] M. J. Ranum, **An Internet Firewall**, proceedings of World Conference on Systems Management and Security, 1992. ftp://decuac.dec.com/pub/docs/firewall.ps

[3] ZWICK, Elizabeth D. **Bulding Internet Firewalls**, O'Reilly & Associates, Inc., Second Edition, 2000

[4] Warfield, Michael H. **Security and the Open Source Model .** Atlanta Linux Showcase, 1999. http://www.wittsend.com/mhw/1999/oss_security/

[5] SCHUBA, Christoph Ludwig, **Design, and Implementation of Firewall Technology**. Purdue University, 1997.

[6] Peterson, James L. Petri Net Theory and Modeling of System. Prentice-Hall, 1981.

[7] RUSSEL, Rusty. **Linux netfilter Hacking**. Disponível para download em http://netfilter.samba.org, 2000

[8] Maxwell, Scott. **Linux Core Kernel commentary**. Coriolis Open Press, 1999.