REVISTA DE CIÊNCIAS EXATAS E TECNOLOGIA

IMPLEMENTAÇÃO MODULAR DE UM SISTEMA DE FIREWALL

Fabiano Sabha Walczak – *Faculdade Anhanguera de Taubaté - unidade I*

RESUMO: O uso de sistemas de segurança é cada dia mais freqüente nas empresas de todos os portes. O nível de segurança que esses sistemas são capazes de prover ao ambiente operacional em que são instalados depende muito da maneira como são implementados e administrados. Neste artigo aborda-se a implementação modular de um sistema de *firewall*, demonstrando-se os principais conceitos e vantagens envolvidas na sua utilização. Apresentam-se também alguns exemplos de implementação prática desta tecnologia.

PALAVRAS-CHAVE:
Segurança, Filtro, Sistemas de segurança, Linux

ABSTRACT: The security systems is more used in the companies. The level of security that these systems are capable to provide in the operational environment where is installed, it's depends so much on the way as that are implemented and managed. In this paper is approached modular implementation of a firewall system, which demonstrating to the main concepts and advantages to use it. Some examples of practical implementation of this technology are also showed .

KEYWORDS:
Security, Filter, Security systems, Linux

Artigo Original
Recebido em: 10/11/2009
Avaliado em:21/06/2010
Publicado em: 22/04/2014

Publicação
Anhanguera Educacional Ltda.

Coordenação
Instituto de Pesquisas Aplicadas e Desenvolvimento Educacional - IPADE

Correspondência
rc.ipade@anhanguera.com

v.5 • n.5 • 2010 • p.101-111

1. INTRODUÇÃO

Dentre as várias necessidades de segurança, a mais comum de encontrar nas empresas de pequeno e médio porte é a implementação de um sistema de filtragem de tráfego direcionado para o bloqueio das conexões não autorizadas. Estes sistemas são usualmente denominados como firewalls e podem se tornar um grande aliado dos administradores de rede.

Os firewalls também podem ocasionar vários problemas com relação à segurança. Um dos mais complexos decorre da falsa sensação de segurança que podem proporcionar, fato que em muitos casos, pode ser pior do que não ter segurança de forma sabida e declarada.

Essa linha muito tênue que separa os dois lados da contribuição do firewall no ambiente de segurança depende, da forma com que esse sistema é implementado e administrado.

Neste artigo abordam-se os principais aspectos envolvidos na implementação de um firewall baseado em um sistema operacional Linux, utilizando como base a ferramenta IpTables, que possibilita a manipulação do Netfilter, o firewall de tecnologia state full nativo em sistemas Linux.

Objetivos do Estudo

O objetivo deste artigo é apresentar um modelo de implementação e gerenciamento modular do sistema de firewall, através do tratamento individual de cada serviço ou conjunto de serviços, de forma a simplificar sua manutenção e gerenciamento.

Isto é feito através da aplicação de regras permissivas ou restritivas criadas através de diversos shell scripts que podem ser desenvolvidos especificamente para cada serviço ou porta cujo tráfego se pretende controlar. Esses scripts são carregados na inicialização dos sistemas de produção, de forma a implementar todas as regras definidas pelo administrador do ambiente.

O fato das regras serem implementadas em scripts independentes permite simplificar as operações necessárias para testar, revisar, ou implementar novas regras, bastando para isso executar o shell script correspondente ao serviço ou porta que se deseja gerenciar.

2. FIREWALL

De acordo com Kurose (2006, p. 541) um firewall é a combinação de software e hardware que isola uma rede local de uma empresa da internet, controlando os pacotes que podem ou não trafegar entre as duas redes. O firewall permite que o administrador da rede controle o acesso entre a rede que administra e o mundo externo.

No contexto da segurança de rede, um firewall executa a função de proteger as conexões entre redes, são dispositivos que permitem que os administradores restrinjam o acesso a componentes da rede. (GALLO, 2003).

Gallo (2003, p. 535) afirma ainda que "em resumo, um firewall é freqüentemente uma soma de muitos componentes diferentes que trabalham juntos para bloquear transmissão e recepção de tráfego".

Os firewalls são apenas uma adaptação moderna de uma antiga forma de segurança medieval: cavar um fosso profundo em torno de um castelo, forçando todos aqueles que desejassem entrar ou sair do castelo, a passar por um único caminho. (TANENBAUM, 2003)

Netfilter

O Netfilter foi introduzido na versão 2.4 do kernel do Linux como um novo mecanismo de manipulação de pacotes, a ferramenta utilizada para controlar o Netfilter é o iptables. (NEMETH, 2007, p. 486). De acordo com o autor "[...] é o irmão mais novo do antigo comando ipchains utilizado nos kernels 2.2 do Linux".

O iptables aplica cadeias ordenadas de regras, um conjunto de cadeias formam as tabelas que são utilizadas para manipular tipos específicos de tráfego de rede.

A tabela padrão é denominada filter e possui três cadeias padrão:

- *FORWARD:* As regras desta cadeia são aplicadas a todos os pacotes que chegam a uma interface e necessitam ser encaminhados a outra.
- *INPUT:* Essa cadeia controla todos os pacotes que são destinado diretamente ao host local, ou seja, pacotes de entrada.

OUTPUT: As regras dessa cadeia refletem em todos os pacotes que possuem como origem o host onde o iptables esta sendo executado.

Ainda segundo Nemeth (2007, p. 486), o iptables possui outras duas tabelas, denominadas Nat e magle. A primeira possui cadeias de regras que controlam a translação de endereços de rede e a outra contém cadeias que modificam ou alteram o conteúdo dos pacotes de rede fora do contexto da filtragem de pacotes e NAT. (NEMETH, 2007).

Apesar da grande utilidade da tabela magle esse artigo não apresenta nenhum modelo de uso específico dessa funcionalidade.

Independente da tabela, cada regra que compõe uma cadeia possui um alvo, que determina qual deve ser o tratamento que o pacote que atende àquela regra deve receber.

Nemeth (2007, p.486) afirma que os alvos disponíveis a essas regras são:

- ACCEPT;
- DROP;
- REJECT;
- LOG;
- MIRROR;
- QUEUE;
- REDIRECT;

- RETURN;e
- ULOG.

A finalidade deste artigo não é detalhar as funcionalidades e/ou cada um dos recursos contidos no Netfilter, portanto não serão abordados com maior ênfase, pois o objetivo é demonstrar uma nova maneira de implementar esses recursos.

Tipos De Firewall

Existem dois tipos principais de Firewall, que podem ser classificados: servidores proxy de aplicativos e filtragem de pacotes.(HATCH; LEE; KURTZ, 2002) Kurose (2006, p. 542) classifica os firewalls em filtro de pacote e gateway de aplicação.

É possível encontrar o uso dos dois tipos de firewalls ao mesmo tempo:

> Muitos firewalls, principalmente as versões comerciais, são híbridos dos dois tipos. Esses são freqüentemente chamados de filtros de pacote com estados, porque monitoram os detalhes de estados de alguns aplicativos para suportar protocolos como o FTP, e ainda são baseados em filtros de pacotes, o que permite que executem o processamento mais rapidamente. (HATCH; LEE; KURTZ, 2002, p. 418)

Atualmente existem centenas de produtos de firewalls no mercado, praticamente todos usam os métodos básicos, filtragem de pacotes, serviços de proxies, para oferecer um serviço de segurança (STREBE; PERKINS, 2000

Servidores Proxy de Aplicativos

Segundo Kurose (2006, p. 544-545) um gateway de aplicação é um servidor de aplicação através do qual todos os dados, de entrada e saída, de uma aplicação devem passar.

Ainda de acordo com Kurose (2006, p. 545) redes internas normalmente possuem vários gateways de aplicativos para vários serviços como telnet, HTTP- Hyper Text Transfer Protocol, FTP – File Transfer Protocol e outros. São usados para monitorar e controlar todo o tráfego de uma rede, o tipo mais comum é aquele onde o usuário deve se conectar a fim de executar alguma atividade na internet. Se um usuário de uma rede local quiser se conectar a um endereço qualquer da internet, primeiro ele deverá conectar-se ao servidor Proxy para, em seguida, a partir dele, poder se conectar com o endereço desejado.

Um servidor Proxy pode gravar sua atividade em um arquivo de log.

Essa atividade pode incluir todos os arquivos que foram feito download, assim como todos os endereços visitados. (HATCH;LEE;KURTZ, 2002).

Gallo (2003, p. 537) destaca que existem dois tipos de servidores Proxy. No primeiro uma conexão recebida pelo servidor Proxy, seria interceptada, e então, uma nova conexão do Proxy para o destino seria criada.

A segunda variante do servidor Proxy permite que o firewall apareça como o único destino para todas as aplicações de uma rede, desta forma rede interna é protegida

completamente das conexões externas, assim como é apresentado na Figura 1.

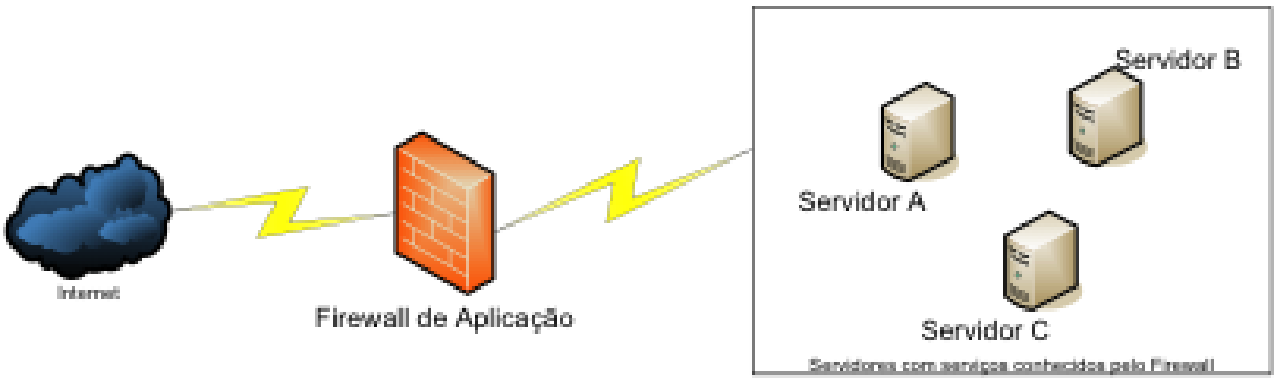

Figura 1: Esquema básico de implementação de proxy de aplicação.

Gateways de aplicação também possuem desvantagens, Kurose (2006, p. 545) cita que é preciso um *gateway* de aplicação diferente para cada aplicação diferente, além da degradação do desempenho, visto que todos os dados serão repassados por meio do *gateway*.

Filtragem de Pacotes

Kurose (2006, p. 542) define que as regras de filtragens especificadas pelo administrador são aplicadas, após a análise, dos cabeçalhos de datagramas determinando se o datagrama será descartado ou não.

Ainda segundo Kurose (2006, p. 542) as decisões de filtragem são baseadas em:

- Endereço IP de origem e de destino;
- Porta TCP ou UDP de origem e de destino;
- Tipo de mensagens ICMP;
- Datagramas de inicialização de conexão usando bits TCP SYN ou ACK.

Uma filtragem de filtragem pode ser baseada na combinação de endereços e número da porta específica. (KUROSE, 2006)

Kurose (2006, p. 543) destaca ainda que embora pareça razoavelmente simples especificar regras de filtragem de pacotes, na verdade há muitas sutilezas e armadilhas potenciais envolvidas.

A Figura 2 apresenta o esquema básico da implementação de um firewall ba-seado em filtragem de pacotes em uma rede local, conectada à internet. Todo o controle de fluxo acontece no firewall que esta na borda da rede local.

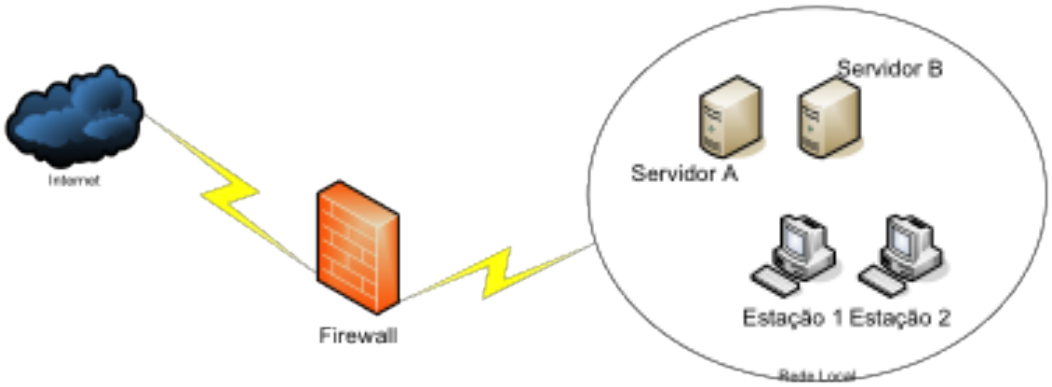

Figura 2: Esquema básico de implementação de filtragem de pacote.

Do ponto de vista de funcionalidade o *firewall* de filtragem de pacotes é oposto de um *firewall gateway* de aplicação, sendo o primeiro de propósito geral e o segundo específico. (GALLO, 2003)

Tanenbaum (2003, p. 825) afirma que em geral, os filtros de pacotes são baseados em tabelas configuradas pelo administrador e são essas tabelas que listam os destinos e as origens aceitáveis e/ou rejeitados, assim como as regras que orientam que ação tomar com os pacotes recebidos de outras máquinas ou destinados a elas.

Segundo Nemeth (2007, p. 484), " Um *firewall* para filtragem de pacotes limita os tipos de trafego que podem passar pelo seu gateway de internet [...]".

3. ESTRATÉGIAS PARA SEGURANÇA

Cheswick; Bellovin; Rubin (2007, p.29-31) destacam algumas estratégias para manter uma rede de computadores segura, dentre as quais esse artigo aborda a segurança baseada em *host* e a segurança de perímetro.

A segurança baseada em *host* é baseada na segurança local do equipamento a ser protegido, para os autores "Na maioria das situações, a rede não é o recurso em risco; em vez disso são os pontos finais da rede que são ameaçados". (CHESWICK; BELLOVIN; RUBIN, 2007)

Considerando que o alvo dos atacantes são os computadores, entende-se que os mesmos deveriam ser configurados de forma a resistir a um ataque, mas normalmente não é isso o que acontece, pois sempre existirão falhas nos programas e/ou sistemas de redes ou na administração dos sistemas computacionais e até mesmo no ambiente como um todo. É isso que ocorre com a segurança do host, o atacante precisa vencer apenas uma vez. (CHESWICK; BELLOVIN; RUBIN, 2007).

Por definição, equipamentos em rede não estão isolados, normalmente as máquinas pertencentes a uma mesma rede irão confiar umas nas outras, portanto se o atacante conseguir comprometer um sistema (computador) ele terá a possibilidade de comprometer

os demais sistemas. (CHESWICK; BELLOVIN; RUBIN,2007)

A proteção de um *host*, contra violações originadas na rede, cabe a si pró-prio. Como dito, o problema consiste em brechas óbvias de segurança que os sistemas comerciais possuem. Mas ainda de acordo com os autores é possível aprimorar a segurança de um *host* até um grau relativamente alto fazendo com que o atacantes procurem outros alvos.

Uma estratégia de segurança bastante conhecida é a segurança de periméro, é comum encontrar definições sobre esse tipo de segurança, fazendo uma analogia simples, se o grau de dificuldade para manter uma casa segura, convém que a vizinhança se reúna e construa um muro em torno das casas, aumentando assim a segurança para todas as casas.

Desta forma, os moradores precisam temer somente a duas situações um ataque interno ou e um ataque externo, com forças para romper o tal muro de proteção. Essa abordagem é chamada de segurança de perímetro. (CHESWICK; BELLOVIN; RUBIN,2007)

Transportando essa analogia para o universo da segurança de redes, temos o muro como sendo o *firewall* e as casas, protegidas, os *hosts.*

Para Cheswick; Bellovin; Rubin (2007, p.32), a maior razão para que um *firewall* de perímetro seja provavelmente mais seguro do que um *host*, consiste simplesmente no fato de que não é um host de uso geral. Desta forma recursos com segurança duvidosa que são significativos para o usuário, são desnecessários.

Um aspecto relativo a segurança do *firewall* de perímetro é que normalmente não se tem um usuário de uso normal e constante, isso ajuda na segurança pois usuários podem causar muitos problemas, como o uso de senhas ruins, conhecidas como senhas fracas. (CHESWICK; BELLOVIN; RUBIN,2007)

Outro aspecto importante, segundo os autores, consiste no fato da administração do *firewall* ser realizada por um profissional específico e responsável pela administração de tais máquinas, eles podem estar mais conscientes da segurança.

4. CONCEITO DE FIREWALL MODULAR

Atualmente uma das grandes dificuldades dos profissionais da área da segurança é entender uma implementação de firewall, quando esse esta todo em linha de texto, ou seja, sem que se tenha uma ferramenta de administração com interface gráfica que normalmente é proprietária e dependendo do tamanho da empresa seus custos inviabilizam a implantação.

Essa dificuldade advem do grande número de linhas de código que um *script* de firewall possui.

A implementação modular de um sistema de *firewall* consiste basicamente em agrupar serviços relacionados em um mesmo *script shell.*

Esses *scripts* podem implementar as regras desejadas uma-a-uma ou de forma agrupada segundo o critério do administrador, criando assim diversos *scripts* independentes,

denominados módulos, que serão chamados em um *script shell* principal.

Por exemplo, o administrador pode criar duas regras de controle de e-mail, uma para entrada e outra para saída dentro de um mesmo *script shell*, compondo assim um módulo que pode ser nomeado de "libera_mail_on".

Caso o administrador deseje ativar esse controle basta executar o modulo "libera_mail_on", no caso de problemas nos e-mail´s o administrador deverá verificar primeiramente esse mesmo módulo.

Uso Do Firewall Modular

O uso do *firewall* modular surgiu pela necessidade de dois momentos que fazem parte do ciclo da administração do sistema de segurança: Implementação e Gerenci-amento.

O fácil entendimento das regras a serem implementadas, assim como o agru-pamento por portas ou serviços relacionados facilitam o gerenciamento do *firewall,* considerando a implementação normal de um *firewall* onde as regras são todas listadas uma abaixo da outra, na implementação modular o administrador carrega regra por regra ou grupo por grupo, conforme o padrão adotado.

Cada regra ou grupo de regras a ser implementada possui dois *shell* scripts distintos cujas nomenclaturas se utilizam dos sufixos "*on*" (ligado) e "*off*" (desligado), em uma evidente referência a carregar e descarregar a regra respectivamente. Caso o administrador deseje carregar uma determinada regra deve utilizar o *script* que possui o sufixo "*on*" caso a regra deva ser desabilitada, utiliza-se o sufixo "*off*

O administrador de um sistema de segurança é a todo momento obrigado à promover mudanças nas regras do *firewall* , pois novas necessidades surgem com o desenvolvimento normal das atividades empresariais, em alguns casos essa demanda é gerada inclusive por órgão governamentais que exigem que a empresa envie ou receba dados on-line com a interligação ou integração de sistemas

Esse cenário exige muita atenção no constante processo de gerenciamento de um sistema de firewall . com o a implementação modular o administrador irá simples-mente modificar um módulo já existente ou então criar um novo módulo, sem modificar nenhuma das regras já existentes.

Vantages No Uso Do Firewall Modular

As vantagens no uso do *firewall* modular podem ser percebidas no momento da implementação do sistema quando o administrador inicia o processo de criação das regras a serem carregadas, uma vez que essas regras são implementadas gradativamente o administrador pode testar a regra utilizando o *script* correspondente.

Com esse método a identificação de problemas, eficiência ou não da regra é facilmente

percebida pelo administrador, outra vantagem considerável é a ausência de regras em duplicidades ou regras contraditórias, pois o gerenciamento dos *scripts* e o próprio processo de criação e testes permitem a clara separação das regras, minimizando este problema comum em *firewall* desse nível.

Outro ponto facilmente percebido é no momento do gerenciamento do siste-ma de segurança, pois a facilidade em administrar as regras se torna mais simples e clara através dos módulos. Pelo fato de trabalhar de forma modular o administrador pode, por exemplo, em caso de manutenção, em uma regra que utiliza a porta 25, desabilitar essa regra sem comprometer o funcionamento do *firewall* como um todo, basta executar o *script* da regra com o sufixo "*off*". Com isso, somente a regra corres-pondente ao serviço que utiliza a porta 25 será interrompido.

Essa prática é muito útil em caso de testes rápidos e solução de problemas relacionados ao uso do *firewall.* Principalmente nos casos de manutenção ou acesso administrativo remoto via SSH – *Secure Shel*, sem interface gráfica.

É importante ressaltar que, com essa metodologia aplicada a um ambiente de produção, o mesmo, terá mais efetividade e clareza adotando-se uma política permissiva de *Firewall*, ou seja, por padrão bloqueia-se todas as conexões e apenas os serviços e portas desejadas são liberadas, tratados e monitorados.

Exemplo De Implementação

A seguir apresenta-se um modelo de implementação de um módulo de serviço, bem como em seguida um modelo do módulo principal que chama os diversos módulos criados.

Todos os *scripts* modulares para liberar ou excluir uma liberação, possuem a mesma estrutura, conforme pode ser observado na Tabela 1.

Tabela 1: Estrutura dos módulos.

#!/bin/bash	Informa que o script esta escrito em linguagem para Shell do tipo bash.
PF=$(which iptables)	Localiza o comando iptables e atribui o comando e o caminho à variável PF.
Servico_on( )	Nome do bloco de comandos. Sempre com os sufixo on ou off.
{	Indica o início do bloco “servico_on”.
$PF –A INPUT	Uso do comando iptables para atribuir uma regra de entrada para Chain INPUT. Com o parâmetro –A adiciona uma regra, com o parâmetro –D exclui a regra.
-p <protocolo>	Indica o protocolo a ser utilizado.
-d <IP>	Endereço de destino.
-s <IP>	Endereço de origem.
-j <Ação>	Indica que ação a regra tomará, ACCEPT para aceitar ou DROP para negar, o tráfego.
$PF –A OUTPUT	Uso do comando iptables para atribuir uma regra de saída para a Chain OUTPUT. Com o parâmetro –A adiciona uma regra, com o parâmetro –D exclui a regra.
}	Indica o final do bloco “servico_on”.
Servico_on	Faz a chamada do bloco “servico_on”.

Modelo de módulo de liberação: Neste módulo estão agrupadas as regras para permitir que um servidor, com o endereço IP 192.168.0.1, aceite e responda a uma solicitação de ping de qualquer endereço IP. Nesse exemplo, esse script dever ser gravado no diretório /root/scripts com o nome de ping_on.sh.

```
#!/bin/bash
PF=$(which iptables)
ping_on( )
{
$PF –A INPUT –p icmp --icmp type 11 –d 192.168.0.1 –s 0/0 –j ACCEPT
$PF –A OUTPUT –p icmp --icmp type 8 –d 0/0 –s 192.168.0.1 –j ACCEPT
}
ping_on
```

Modelo de módulo de exclusão de liberação: Como rege o conceito do *firewall* modular, para todo *script* que possui o sufixo “on” deve existir o seu correspondente com o sufixo “off”. Sendo que o primeiro fez a liberação do tráfego específico e o segundo anula essa liberação.

A idéia do sufixo “off” é de fato a exclusão da regra criada pelo *script* que possui o sufixo “on” e não a troca de ação, como por exemplo de ACCEPT para

DROP. Para tanto basta apenas alterar um parâmetro da regra, conforme já apresentado na tabela 1, a linha de código com o parâmetro –A adiciona uma regra para excluir a regra basta repetir a linha de código no entanto com o parâmetro –D ao invés do –A.

A seguir um exemplo desse *script,* que exclui as regras implementadas atra-vés do *script* ping_on.sh, apresentado anteriormente.

```
#!/bin/bash
PF=$(which iptables)
ping_off( )
{
$PF –D INPUT –p icmp --icmp type 11 –d 192.168.0.1 –s 0/0 –j ACCEPT
$PF –D OUTPUT –p icmp --icmp type 8 –d 0/0 –s 192.168.0.1 –j ACCEPT
}
ping_off
```

Assim como o exemplo anterior, esse *script* dever ser gravado no diretório /root/scripts com o nome de ping_off.sh.

O *script* a seguir é onde todos os demais módulos são chamados e se iniciam. O exemplo é escrito para o modelo "System V" de inicialização, mas pode ser adaptado perfeitamente para o modelo BSD, porém nesse artigo essa adaptação não será abordada.

```
#!/bin/bash
#Firewall Modular - script principal
 PF=$(which iptables)
if [ -z $PF ]; then
                    echo "Comando iptables nao encontrado"
                     exit
fi

DIR="/root/scripts"
case "$1" in
   start)
          $DIR/limpa.sh
          $DIR/ping_on.sh

          ;;
   stop)
           $DIR/limpa.sh
  ;;
   fechado)
          $DIR/drop.sh
      ;;
   restart)
```

```
            $0 stop
            $0 start
    ;;
     status)
             $DIR/status.sh
    ;;
     *)
        echo “Use: $0 {start|stop|restart|status|fechado}”
    exit 1
    ;;
esac
exit 0
```

5. CONCLUSÃO

A implementação de um sistema modular de *firewall* visa sobretudo, otimizar em todos os aspectos a rotina de um administrador de segurança. Desde o desenvolvimento das idéias iniciais, planejamento, implementação, bem como em todo o gerenciamento e monitoramento dos serviços de firewall ativos e inseridos em uma rede.

Através de um modelo simples, objetivo e como o próprio nome diz modular, é possível facilitar o trabalho do administrador em todos os sentidos, tanto em tempo de resposta e/ou implantação quanto eficiência na identificação e solução de problemas.

O processo de criação de regras é feito como em qualquer outra implementa-ção de *firewall,* porém o fato de agrupar essas regras segundo os serviços a quais elas se referem e poder trabalhar esses grupos de forma independentes, trás ao administrador um maior domínio sobre todas as regras, evitando assim um erro comum em implementações, a repetição de regras ou então uma regra anulando a outra, fazendo com que aumente o tempo de processamento das regras e trazendo resultados muitas vezes indesejados.

A identificação dos problemas citados em um sistema modular é praticamente feita de forma imediata, pois se a falha ocorre na utilização de um serviço ou porta específica, o administrador não terá que ler todas as regras de seu *firewall* até localizar a correspondente, basta abrir diretamente o módulo correspondente a porta ou serviço e fazer a verificação.

Como sugestão de trabalhos futuros, é interessante iniciar um estudo para o trabalho deste modelo juntamente com a autenticação do usuário junto ao sistema de *firewall*. Dessa forma teríamos não só um *firewall* modular, mas também com perfil designado para cada usuário da rede.

REFERÊNCIAS

CHESWICK, William R.; BELOVIN, Steven M.; RUBIN, Aviel D. **Firewalls e segurança na Internet**: repelindo o hacker ardiloso, 2ª ed. Porto Alegre: Bookman, 2005

GALLO, Michael A.; HANCOCK, William M. **Comunicação entre computadores e tecnologias de rede**, São Paulo: Pioneira Thomson Learning, 2003.

HATCH, Brian;LEE, James; KURTZ, George **Hackers Expostos – Linux** , São Paulo: Makron Books, 2002

KUROSE, James F.; **Redes de computadores e a internet**: uma bordagem top-down, 3ª ed. São Paulo: Pearson Addison Wesley, 2006.

NEMETH, Evi; SNYDER, Garth; HEIN, Trent R. **Manual Completo do Linux**, São Paulo: Pearson Prentice Hall, 2007

STREBE, Mathew; PERKINS, Charles – **Firewalls**, São Paulo: Makron Books, 2000.

TANENBAUM, Andrew S. **Redes de computadores,** Rio de Janeiro: Elsevier, 2003