

UMA FERRAMENTA GRÁFICA PARA FILTRAGEM DE STRINGS COM LINUX/IPTABLES

Ricardo Kléber Martins Galvão

Departamento de Informática e Matemática Aplicada
Universidade Federal do Rio Grande do Norte
CEP 59072-970 Natal – RN
rk@ufrnet.br

Galileu Batista de Sousa

Departamento de Informática e Matemática Aplicada
Universidade Federal do Rio Grande do Norte
CEP 59072-970 Natal – RN
galileu@dimap.ufrn.br

RESUMO

Este trabalho apresenta uma ferramenta gráfica que, através de um conjunto de módulos, realiza a filtragem de strings no fluxo de dados em um gateway, baseada em assinaturas de ataques conhecidas incorporadas ao sistema manualmente e/ou através de scripts de conversão de assinaturas de outras ferramentas e armazenadas em um banco de dados. Baseada no Sistema Operacional Linux, a aplicação utiliza-se da funcionalidade “match string” do módulo netfilter do kernel do Linux, além da ferramenta iptables e assinaturas do IDS Snort. A interface gráfica é baseada na linguagem de scripts PHP executando em um servidor web Apache.

ABSTRACT

This work presents a graphical tool that, through a set of tools, manually carries through the filtering of strings in the stream of data in one gateway, based in signatures of attacks known incorporated to the system and/or through scripts of conversion of signatures of other stored tools and in a data base. Based in the Operational system Linux, the application uses of the functionality “match string” of the module to netfilter of kernel of the Linux, beyond the tool iptables and signatures the IDS Snort. The graphical interface is based on the language of scripts PHP executing in a Apache webserver.

1 INTRODUÇÃO

O controle de tráfego malicioso em uma rede tornou-se uma preocupação constante dos administradores, sendo várias as soluções empregadas para esta tarefa:

- Utilização de mecanismos de filtragem e/ou monitoração do fluxo entrante da Intranet;
- Utilização de ferramentas de detecção de intrusões (IDS);
- Utilização de anti-vírus nas estações, no servidor de correio interno, ou ainda trabalhando em conjunto com um *firewall*;
- Utilização de *sniffers* de modo a capturar e avaliar o conteúdo dos pacotes;
- Utilização de *proxies* com regras de filtragem (ACLs).

Estas práticas, comuns na atualidade, visam o combate antes, durante e depois da entrada dos pacotes na rede.

Diante deste quadro, mecanismos focados na eficiência e otimização de filtros surgem como uma solução mais adequada a esta situação.

A filtragem de *strings*, provida pelo módulo experimental “match string” do *kernel* 2.4.x do sistema operacional Linux, ao contrário das alternativas comumente em uso, analisa o conteúdo das *strings* e, em caso de casamento de padrão (tráfego x assinatura), registra e bloqueia apenas o pacote analisado, não interferindo no fluxo de pacotes “sadios” por não provocar qualquer ação reativa à origem do pacote, como na solução IDS+*firewall*, onde o IDS detecta o pacote suspeito e interage com o *firewall* bloqueando todos os pacotes subsequentes vindos da mesma origem.

Um exemplo da solução IDS+*firewall* é a ferramenta Pigmeat [11], um *software* de ação

reativa que, após comprovado o casamento de padrões de um pacote com uma das regras do IDS, envia um comando ao *firewall* bloqueando todos os demais pacotes vindos da mesma origem deste pacote.

Outra vantagem apresentada pela filtragem de *strings*, quando comparada à solução IDS+*firewall*, é o foco em portas específicas (recurso provido pelo *firewall* do Linux), diminuindo o retardo de análise no *gateway* onde está instalado, como ilustrado na Figura 1.

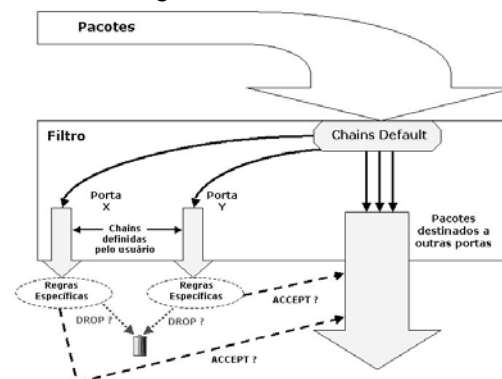


Figura 1 –Filtragem com Foco em Portas Específicas

A utilização de um *script* de conversão de regras do Snort (IDS gratuito amplamente utilizado) para o formato do iptables (*firewall* do Linux) auxilia no processo de atualização de assinaturas de ataque.

Integrando estas soluções em uma interface *web*, a ferramenta apresentada facilita a utilização destes módulos, utilizando-se de um banco de dados (MySQL) para agilizar a consulta a padrões já estabelecidos, inclusão de novas assinaturas e verificação de *logs* do sistema.

2 FILTRAGEM DE PACOTES

2.1 Visão Geral de Filtros de Pacotes

Um filtro de pacotes é um *software* que analisa o *cabeçalho* (*header*) dos pacotes enquanto eles passam, e decide o seu destino [1]. Ele pode decidir entre:

- **descartar (DROP)** o pacote (como se nunca o tivesse recebido);
- **aceitar (ACCEPT)** o pacote (deixando-o seguir ao seu destino);
- **gerar logs (LOG)** ou
- outros alvos (*targets*) especificados pelo usuário.

No Linux, a filtragem de pacotes está implementada diretamente no *kernel*. A principal utilização deste recurso é a realização da análise do cabeçalho dos pacotes e a eventual decisão sobre o seu destino.

2.2 A evolução dos mecanismos de filtragem

Conforme relatado por Russel [2], os *kernels* do Linux têm tido filtros de pacotes desde a série 1.1. A primeira geração, baseada no **ipfw** do BSD, foi portada por Alan Cox no final de 1994. Essa implementação foi melhorada por Jos Vos e outros voluntários para o *kernel* 2.0, com a ferramenta **ipfwadm**, que controlava as regras de filtragem do *kernel*.

Em meados de 1998, Rust Russel e Michael Neuling, voltaram a fazer alterações no *kernel* para implementar novos mecanismos de filtragem. Este esforço culminou no lançamento da ferramenta **ipchains** para o Linux *kernel* 2.2.

Finalmente, a ferramenta da quarta geração, o **iptables**, foi lançada em meados de 1999 para o Linux *kernel* 2.4.

Além das funcionalidades dos seus antecessores, o **iptables** trouxe níveis sofisticados de filtragem, modularização de recursos, inspeção e controle dos estados dos pacotes (*statefull inspection*).

A infraestrutura para funcionamento do **iptables** é implementada pelo **netfilter**, um *framework* adicionado ao novo *kernel* do Linux (2.4) via módulo ou compilado¹ diretamente neste *kernel*.

A ferramenta **iptables** comunica-se diretamente com o *kernel* (via **netfilter**), controlando o destino de cada pacote analisado.

2.3 A Filtragem Tradicional

A filtragem de pacotes implementada desde o *kernel* 2.0 dispõe de três listas de regras na tabela **filter**²; tais listas são denominadas **firewall chains**

(ou simplesmente **chains**). As três *chains* básicas são **INPUT**, **OUTPUT** e **FORWARD**, apresentadas graficamente na Figura 2.

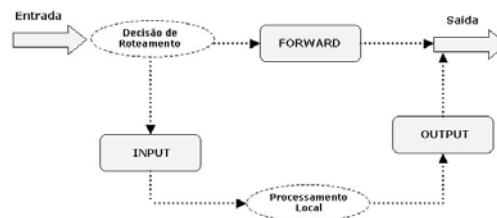


Fig. 2 – Filtragem tradicional do iptables

Os três “balões” em destaque representam as *chains* mencionadas acima. Quando o pacote atinge um balão no diagrama, a *chain* é examinada a fim de que seja decidido o destino do pacote. Se a *chain* aponta para descartar (DROP) o pacote, ele é descartado, mas se a *chain* aponta para aceitar (ACCEPT), ele segue como no diagrama.

Uma *chain* é uma lista de **regras**. Cada regra pode ser interpretada como uma condição do tipo:

“se o cabeçalho do pacote se parece com isso, aqui está o que deve ser feito com o pacote”

Se a regra não se associa com o pacote, então a próxima regra na *chain* é consultada. Não havendo mais regras a consultar, o *kernel* analisa a **política**³ da *chain* para decidir o que fazer.

Conforme ilustrado na Figura 2, quando o pacote chega ao computador, pela placa *ethernet*, por exemplo, o *kernel* analisa o seu destino, num processo denominado roteamento (*routing*). Segue-se então a decisão de roteamento, onde é realizado o repasse (FORWARD), ou, caso o pacote se destine à própria máquina, o encaminhamento à *chain* INPUT. Ao chegar nestas *chains*, o pacote passa por regras específicas que decidirão o seu destino (aceitar ou descartar).

2.4 Novas Implementações do Iptables

Além dos recursos das soluções de filtragem de pacotes que o antecederam, o **iptables** (com o suporte do **netfilter**) é extensível, isto é, dispõe de suporte a módulos adicionais, proporcionando várias outras funcionalidades. Estes módulos podem ser carregados “por demanda”, de acordo com a necessidade de filtragem.

Estas extensões são classificadas em três tipos:

- Extensões de controle de protocolos (-p)
- Novos alvos (-j)
- Novas associações (-m)

¹ Para suportar o módulo **netfilter**, na compilação a opção **CONFIG_NETFILTER** deve ser habilitada.

² módulo específico do *kernel* para filtragem de pacotes.

³ A *chain* pode ter 2 tipos de política: “Tudo o que não for expressamente proibido é permitido” ou “Tudo o que não for expressamente permitido é proibido”.

Extensões TCP

As extensões TCP são automaticamente carregadas quando especificada a opção '**-p tcp**' e dispõe das seguintes opções:

--tcp-flags

Permite que sejam filtradas *flags* TCP específicas.

--source-port

Indica uma porta ou conjunto (*range*) de portas TCP de origem.

--destination-port

Indica uma porta ou conjunto (*range*) de portas TCP de destino.

--tcp-option

Utiliza-se de números específicos para controlar (e eventualmente descartar) pacotes com a opção TCP igual ao do número informado. Um pacote que não tem um cabeçalho TCP completo é automaticamente descartado se há uma tentativa de examinar suas opções TCP.

Extensões UDP

Essas extensões são automaticamente carregadas se a opção '**-p udp**' é especificada.

Provê as opções '**--destination-port**' e '**--source-port**'.

Extensões ICMP

Essas extensões são automaticamente carregadas se a opção '**-p icmp**' é especificada.

Possui uma só opção diferente das demais:

--icmp-type

Controla o tipo de conexão icmp baseado no nome de tipo ou tipo numérico comumente utilizado em conexões deste tipo.

2.6 Novos Alvos

Os alvos (*targets*) indicam “o que fazer” com o pacote caso coincida com as condições da regra.

Os alvos padrões embutidos no iptables são: **DROP** (descartar) e **ACCEPT** (aceitar). Se a regra se associa com o pacote e seu alvo é um desses dois, nenhuma outra regra é consultada: o destino do pacote já foi decidido.

Há dois tipos de alvos diferentes dos descritos acima:

- as *chains* definidas por usuários e
- as extensões

Chains definidas por usuários

Uma funcionalidade que o iptables herdou do ipchains é a possibilidade da criação de novas *chains*, além das três disponíveis (INPUT, FORWARD e OUTPUT).

Quando um pacote associa-se com uma regra cujo alvo é uma *chain* definida pelo usuário, o pacote passa a ser analisado pelas regras dessa nova *chain*.

A Figura 3 mostra graficamente duas *chains*: INPUT (a *chain* padrão) e test (uma *chain* definida pelo usuário).

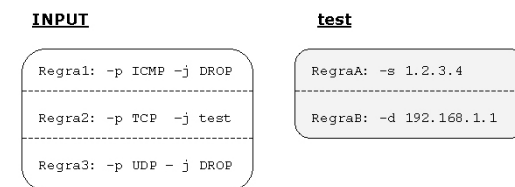


Fig. 3 – Chains definidas pelo usuário.

No modelo da Figura 3, considerando um pacote TCP vindo de 192.168.1.1, com destino a 1.2.3.4, este pacote submete-se à *chain* INPUT e é testado pela Regra1, com a qual não se associa. Porém, quando submetida à Regra2 ocorre a associação, tendo como alvo test. Logo, a próxima regra examinada será a primeira da *chain* test. A RegraA na *chain* test não se associa ao pacote por não coincidir com a origem, passando o pacote à próxima regra (RegraB). Como também não ocorre a associação, já que apresenta um destino diferente, e o pacote chega ao final da *chain* sem se associar a nenhuma regra, a análise retorna à *chain* INPUT. Já que a última regra examinada desta *chain* foi a Regra2, a regra a ser examinada agora é a Regra3, que também não se associa com o pacote.

Graficamente, o caminho percorrido pelo pacote é como ilustrado na Figura 4.

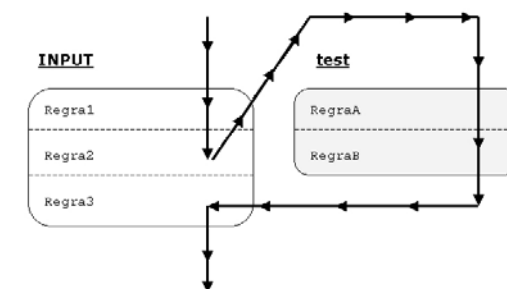


Fig. 4 – Caminho do pacote na nova chain

Extensões ao iptables: Novos alvos (targets)

O outro tipo de alvo é a extensão. Uma extensão-alvo consiste em um módulo do *kernel*, opcional ao iptables, para prover opções de linha de comando. As extensões na distribuição padrão do netfilter são:

LOG

Esse módulo provê o registro em *logs* dos pacotes submetidos, possuindo as seguintes opções adicionais:

--log-level

Seguido de um número de nível ou nome.

--log-prefix

Seguido de uma *string* de até 29 caracteres, que será adicionada no início da linha de registro de log (*syslog*), permitindo melhor identificação da mesma.

REJECT⁴

Esse módulo tem o mesmo efeito do alvo '**DROP**', porém, retorna uma mensagem de erro ao remetente, rejeitando o pacote.

2.7 Novas Associações

Estes tipos de extensões no pacote netfilter (caso instaladas) podem ser habilitadas com a opção '-m'.

Estas funcionalidades adicionais permitem um controle mais detalhado dos pacotes, não se detendo ao cabeçalho, analisando informações em seu conteúdo.

mac

Utilizado para associar a regra com o endereço *Ethernet* (MAC) da máquina de origem do pacote (**--mac-source**).

limit

Utilizado para restringir a taxa de pacotes, e para suprimir mensagens de log.

owner

Associa a regra a várias características do criador do pacote gerado localmente.

state

Interpreta a análise do controle da conexão feita pelo módulo '**ip_conntrack**'.

3 A FILTRAGEM DE STRINGS

3.1 Histórico da filtragem de strings

Até a versão 2.4.9 do *kernel* do Linux, a implementação de um *firewall* baseado no netfilter/iptables oferecia a possibilidade de elaboração de regras envolvendo apenas informações de cabeçalho até o nível de transporte. A filtragem de pacotes baseada em seu conteúdo⁵ não era possível.

Diante dos benefícios já relatados, a equipe de desenvolvimento do netfilter criou, e disponibilizou a partir de então, uma nova associação (extensão) com a capacidade de realizar filtros mais detalhados, incluindo o suporte ao conjunto de informações que compõem o nível de aplicação (cabeçalho de dados) de um pacote. Esta nova característica notadamente funcional e poderosa é responsável pela infraestrutura da ferramenta apresentada neste trabalho.

3.2 Ferramentas Gráficas

O conjunto das funcionalidades do iptables é tão eficiente quanto as soluções comerciais disponíveis. O diferencial destas soluções, porém, é o modo de interação com o usuário (interfaces) que as tornam mais efetivas e fáceis de manipular, apresentando seus recursos de forma mais amigável.

Para suprir esta carência, várias interfaces surgiram dando maior flexibilidade de uso aos recursos do iptables. Porém, durante a elaboração deste artigo, não foi encontrada nenhuma ferramenta gráfica que explorasse os recursos de filtragem de *strings*.

Proporcionar um ambiente gráfico para a manipulação das regras do iptables relacionadas a este tipo de filtragem é o objetivo principal deste artigo.

4 PREPARAÇÃO DO AMBIENTE

4.1 Aplicação de patches e compilação do kernel

Dentre as melhorias mais significativas proporcionadas pelo *kernel* 2.4 do Linux, estão o *firewalling statefull*, que implementa a filtragem dos pacotes baseada em combinações de *flags* TCP e MAC *address*, e a filtragem baseada em *strings* (*match string*).

Este último módulo, por ainda encontrar-se em fase experimental, não vem habilitado por padrão nos *kernels* pré-compilados das distribuições atuais, necessitando, portanto, de uma nova compilação para a sua utilização.

Utilizou-se para este trabalho o *kernel* 2.4.16 (<ftp://ftp.kernel.org>) e o iptables 1.2.4 (<http://netfilter.samba.org>).

A partir dos arquivos fontes do iptables, aplicou-se os *patches* necessários para o suporte pelo *kernel* da funcionalidade de filtragem de *strings*, confirme a Figura 5.

```
make pending-patches KERNEL_DIR=/usr/src/linux-2.4.16
make patch-o-matic KERNEL_DIR=/usr/src/linux-2.4.16

Estes patches possuem menus interativos, dentre os quais o
essencial para ativar a funcionalidade é o descrito a
seguir:

Testing... string.patch NOT APPLIED ( 2 missing files)
The string patch:

Author: Emmanuel Roger
Status: Working, not with kernel 2.4.9

This patch adds CONFIG_IP_NF_MATCH_STRING which allows you
to
Match a string in a whole packet.

THIS PATCH DOES NOT WORK WITH KERNEL 2.4.9 !

Do you want to apply this patch [N/y/t/f/q/?] y
```

Fig. 5 –Aplicação do *patch* para suporte a *match string*

Em seguida, compilou-se o código do iptables e bibliotecas correspondentes, conforme Figura 6.

⁴ O REJECT era um alvo nativo no ipchains. Porém, pela semelhança com o DROP, (além da pouca utilização), foi disponibilizado no iptables como uma extensão opcional.

⁵ Área de dados do pacote no nível de transporte (incluindo informações de cabeçalho do nível de aplicação).

```
make KERNEL_DIR=/usr/src/linux-2.4.16
make install KERNEL_DIR=/usr/src/linux-2.4.16
```

Fig. 6 – Compilação do iptables

Compilou-se então o *kernel* com a nova funcionalidade. A Figura 7 realça a opção essencial que deve ser marcada durante a configuração do *kernel* a compilar.

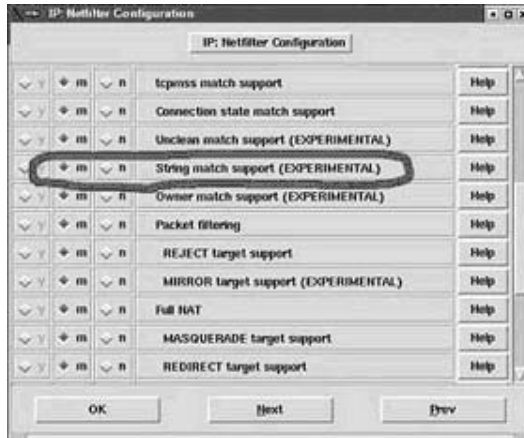


Fig. 7 – Habilitando o *match string* no *kernel*

4.2 Serviços Necessários

Além da instalação do iptables e recompilação do *kernel* do Linux com suporte a *match string*, os seguintes *softwares* devem ser instalados e configurados para o funcionamento da ferramenta sistema:

- Servidor Web Apache
- Linguagem PHP
- Banco de Dados MySQL

5 MÓDULOS DO SISTEMA

O sistema tem uma interface *web*, como apresentada na Figura 8, onde os módulos acessados utilizam-se de scripts PHP para realizar consultas, inclusões e alterações no banco MySQL que armazena as regras de filtragem. O script iptables pode ser gerado a partir da página principal e instalado em seguida ou concatenado a regras de filtragem geradas por outras ferramentas.

Os módulos do sistema são:

- Visualizador de Regras;
- Inserção Manual de Regras;
- Inserção de Regras do Snort;
- Manutenção de Regras;
- Geração de Script; e
- Instalação de Regras



Fig. 8 – Tela principal da Ferramenta

5.1 Visualizador de Regras

Este módulo apresenta a listagem (ordenada por porta) das regras armazenadas no banco de dados.

A Figura 9 apresenta um exemplo de listagem de regras utilizando a ferramenta.

VISUALIZAR REGRAS			
Protocolo	Porta	String	Label
tcp	25	ahaha	virus
tcp	80	/websendmail	Webendmail CGI access attempt
tcp	80	/webqgis	Webqgis CGI access attempt
tcp	80	/php.cgi	PHP CGI access attempt
tcp	80	/pht	PHP CGI access attempt
tcp	80	/faxsurvey	FAXSURVEY probe!
tcp	80	/Count.cgi	COUNT.cgi probe!
tcp	80	/handler	HANDLER probe!
tcp	80	/test.cgi	TEST-CGI probe!

Fig. 9 – Listagem gerada pelo Visualizador de Regras

5.2 Inserção Manual de Regras

Com este módulo da ferramenta o administrador pode inserir novas assinaturas de ataque informando os seguintes dados:

- Protocolo
- Porta
- *String*
- Label de Identificação da regra para os *logs*

A Figura 10 ilustra a tela de um exemplo deste módulo com o formulário para preenchimento de informações da ferramenta.

Um clique no botão inserir acrescenta a regra ao banco de dados.



INSERIR REGRAS MANUALMENTE

Protocolo :

Porta :

String :

Label :

Fig. 10 – Formulário de inserção manual de regras

5.3 Inserção de Regras do Snort

O **snort** (www.snort.org) é hoje um dos IDS mais utilizados, principalmente no ambiente GNU/Linux, contando com várias bases de dados de assinaturas em todo o mundo em constante atualização.

Com o aparecimento a cada dia de novas formas de ataques aos vários serviços Internet (SMTP, POP3, HTTP, etc), a falta de atualização na base de dados de assinaturas faria com que a ferramenta se tornasse, com o tempo, ineficaz, já que não teria em sua base de assinaturas padrões de reconhecimento para estes novos ataques.

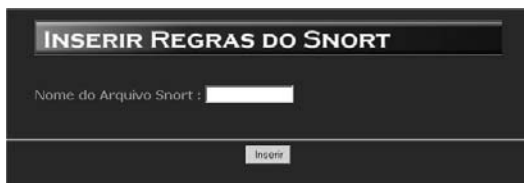
A utilização dos repositórios atualizados de assinaturas do Snort convertidas ao formato do iptables apresenta-se como uma solução eficaz para este problema.

O módulo de inserção de regras do snort utiliza-se de um *script* que converte assinaturas do formato snort aos parâmetros utilizados pelo iptables para a filtragem de *strings*. Bastando então, ao administrador, copiar as novas regras para um diretório específico e utilizar a ferramenta para realizar a adequação das assinaturas ao sistema.

O *script* de inserção de regras do snort lê o arquivo de regras selecionado, considerando os campos referentes ao **protocolo**, **porta destino**, **string suspeita** e **texto** para identificação nos *logs* (*label*).

O módulo de inserção de regras do snort da ferramenta deste artigo apresenta uma tela inicial, como na Figura 11, onde o administrador deve informar o nome do arquivo de regras a converter.

Este arquivo deve ter sido previamente copiado para um diretório acessível pelo usuário *nobody* (já que o sistema é acessado via *web*).



INSERIR REGRAS DO SNORT

Nome do Arquivo Snort :

Fig. 11 –Módulo de inserção de regras do Snort

O sistema então processa o arquivo, convertendo cada linha válida para o formato do iptables. Após a conversão, é apresentada uma tabela contendo os registros inseridos no banco de dados de regras, como na Figura 12.



INSERIR REGRAS DO SNORT

Protocolo	Porta	String	Label
tcp	80	/test.cgi	TEST-CGI probe!
tcp	80	/handler	HANDLER probe!
tcp	80	/Count.cgi	COUNT.cgi probe!
tcp	80	/faxsurvey	FAXSURVEY probe!
tcp	80	/phf	PHF CGI access attempt
tcp	80	/php.cgi	PHP CGI access attempt
tcp	80	/webgnis	Webgnis CGI access attempt
tcp	80	/websndmail	Websndmail CGI access attempt

Figura 12 – Regras inseridas do Snort

5.4. Manutenção de Regras

Com este módulo, o administrador tem a opção de excluir ou alterar regras pré-existent selecionadas a partir do banco de dados. As alterações, contudo, são realizadas apenas no banco de dados, sendo necessária uma nova geração e instalação de *script* para que produzam o resultado desejado.

Acionando o módulo a partir da página principal, a tela apresentada lista todas as regras existentes no banco e solicita o número de índice (listado ao lado de cada regra) da regra a alterar, como ilustra a Figura 13.



ALTERAR REGRAS

Numero de Índice da Regra a Alterar :

Índice	Protocolo	Porta	String	Label
1	tcp	80	/websndmail	Websndmail CGI access attempt
2	tcp	80	/webgnis	Webgnis CGI access attempt
3	tcp	80	/php.cgi	PHP CGI access attempt
4	tcp	80	/phf	PHF CGI access attempt
5	tcp	80	/faxsurvey	FAXSURVEY probe!
6	tcp	80	/Count.cgi	COUNT.cgi probe!
7	tcp	80	/handler	HANDLER probe!
8	tcp	80	/test.cgi	TEST-CGI probe!

Fig. 13 – Seleção da regra a alterar

Selecionada a regra, o administrador terá uma tela, como na Figura 14, onde poderá modificar os campos e em seguida pressionar o botão **alterar**, ou excluir o registro pressionando o botão correspondente.

Fig. 14 – Alterando regras preexistentes

5.5 Geração de Script

Módulo utilizado para gerar o *shell script* (regras.sh) com as regras no formato do iptables prontas para a instalação ou concatenação com regras de filtragem de pacotes e/ou NAT.

A instalação pode ser feita diretamente na ferramenta. Em caso de utilização do iptables em uma mesma máquina para filtragem de pacotes e/ou NAT, o *script* gerado contendo as regras de filtragem de *strings* pode ser concatenado às regras geradas para estes fins por ferramentas como:

- Firewall Builder [12];
- Alfandega [13]; ou
- AGT [14].

O *shell script* gerado tem um cabeçalho padrão com linhas de comando que executam a remoção (*flush*) das regras pré-existentes do iptables.

As demais linhas do *script* gerado são criadas a partir da consulta às regras inseridas no banco de dados.

Cada regra presente no banco corresponde a duas linhas no *script* gerado, sendo a primeira linha responsável pelo registro em *logs* dos pacotes que coincidem com as regras e a segunda linha responsável pelo bloqueio destes pacotes.

5.6 Instalação de Regras

Este módulo é utilizado caso não existam regras adicionais de NAT e/ou filtragem de pacotes para concatenar com o *script* gerado. Basicamente seu funcionamento é executar o *script* gerado pelo módulo anterior, ativando as regras do iptables.

Na prática, a utilização deste módulo não é recomendada em situações reais, onde o *gateway* é administrado remotamente via *web*, já que a execução do *script* a partir do *browser* está condicionada à concessão de permissões de superusuário a qualquer usuário que acesse o sistema. Deste modo, o usuário passaria a ter permissões de manipulação direta do *kernel*.

O uso deste módulo da ferramenta deve ser viabilizado para administração local, sendo a interface *web* utilizada como um mecanismo gráfico de interação com o sistema e não como uma ferramenta de administração remota.

5.6.1 O Script *gerascript.php*

A geração do *script* iptables é o módulo principal da interface *web*. Esta geração é realizada pelo *script* **gerascript.php** que concatena as consultas às regras do banco de assinaturas em uma variável, escrevendo-a em seguida no *script* **regras.sh**.

6 ESTRUTURA DO BANCO DE DADOS

O banco de dados MySQL (www.mysql.com), apesar de não apresentar a mesma variedade de recursos dos bancos mais “robustos”, foi escolhida para fazer parte desta solução pela facilidade de manutenção e rapidez de resposta às consultas realizadas pela ferramenta.

A estrutura do banco de dados **regras** apresenta quatro campos referentes aos parâmetros de cada linha de assinatura do iptables.

- Protocolo – char (4)
- Porta/Serviço – int (11)
- String – char (100)
- Label – char (28)⁶

6 ANÁLISE DE PERFORMANCE

6.1 Testando uma Regra Específica

Verificando-se as regras inseridas no banco de dados, como na Figura 15, selecionou-se uma das strings listadas para os testes:

“/php.cgi” – Log Label: PHP CGI access attempt

Protocolo	Porta	String	Label
tcp	80	/webmail	Webmail CGI access attempt
tcp	80	/webmail	Webmail CGI access attempt
tcp	80	/php.cgi	PHP CGI access attempt
tcp	80	/php	PHP CGI access attempt
tcp	80	/faxsurvey	FAXSURVEY probe
tcp	80	/Count.cgi	COUNT.cgi probe
tcp	80	/handler	HANDLER probe
tcp	80	/test.cgi	TEST-CGI probe

Fig. 15 – Seleção de regra para teste

Interpretando a regra selecionada, temos que o LOG/DROP deve ser ativado para pacotes tcp destinados à porta 80 do servidor e que contenham a *string* **/php.cgi**. Na ocorrência desta *string*, deverá ser gerado um *log* tendo como *label* identificador **PHP CGI access attempt**.

⁶ O tamanho máximo de uma *label* (--log-prefix) no iptables é de 29 caracteres.

A partir de um navegador (*browser*) remoto, solicitou-se ao servidor uma página com a *string* mencionada, como na Figura 16.



Fig. 16 – Teste de *string* suspeita

O servidor, com as regras aplicadas, não permite a consulta pela existência ou não da página (DROP) e registra (LOG) a ocorrência nos *logs* do sistema, como descrito na Figura 17.

```
Apr 6 17:40:02 servidor kernel: PHP CGI access  
attemptIN=eth0 OUT=  
MAC=00:e0:4c:39:02:8a:00:04:75:70:8c:b5:08:00  
SRC=200.19.163.1 DST=150.232.60 LEN=382  
TOS=0x00 PREC=0x00 TTL=108 ID=5838 DF  
PROTO=TCP SPT=13556 DPT=80  
WINDOW=17520 RES=0x00 ACK PSH URGP=0
```

Fig. 17 – Logs do sistema gerado pelo iptables

Assim caracteriza-se uma tentativa de acesso ao servidor utilizando uma *string* definida como suspeita. Em destaque no registro de logs na Figura 17, a *label* específica da regra, e o endereço de origem (SRC).

6.2 Comparações com outras soluções

O resultado da ação desta ferramenta foi o DROP do(s) pacote(s) contendo a *string* suspeita e o registro desta ação nos *logs* do sistema, sem qualquer restrição a eventuais pacotes enviados posteriormente independente da origem.

Segue a avaliação de outras ferramentas/soluções para identificação de ataques baseada na análise de *strings*.

6.2.1 Snort IDS

A utilização de um IDS como o snort (Figura 18), baseia-se na instalação de um *sniffer*⁷ na rede e análise de todos os pacotes que passam pelo segmento.

⁷ Sniffer: Ferramenta que, colocando a interface de rede da máquina em modo promíscuo, analisa todos os pacotes enviados a máquinas da rede (por *broadcast*).

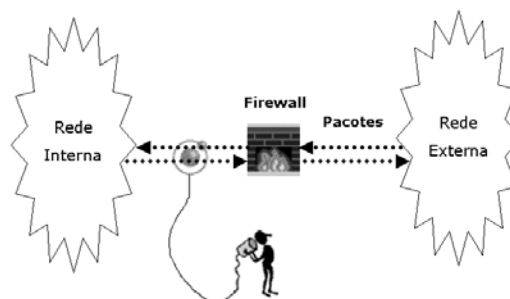


Fig. 18 – Funcionamento do Snort IDS

A ineficiência deste mecanismo comparando-o à ferramenta apresentada reside no fato de que não há ação reativa sobre o(s) pacote(s) que contenha(m) *strings* coincidentes com as regras ativas.

A incidência das *strings* coincidentes é reportada nos *logs* do sistema, mas, o pacote segue seu destino sem qualquer intervenção.

6.2.2 Snort IDS + Firewall

O uso de *scripts* que analisam os *logs* do snort e adicionam regras ao *firewall* sem a intervenção do administrador no caso de ocorrência(s) de casamento de padrões (Figura 19), ignora posteriormente todos os pacotes de cuja origem tenham partido pacotes coincidentes com alguma assinatura do snort.

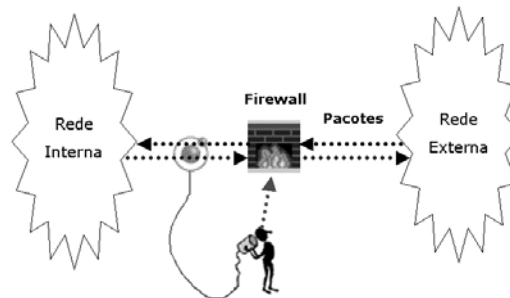


Figura 19 – Solução snort IDS + Firewall

A ineficiência desta solução reside no fato de que a ocorrência de falso-positivos ou ataques de *ip spoofing*⁸ podem bloquear por tempo indeterminado conexões legítimas ao servidor.

Uma alternativa às soluções que inserem regras no *firewall* descartando todos os pacotes de origens reportadas pelo snort, como o **Pigmeat** [11], é o *script* **snort2iptables** [15] – indicado pelo **CHUVAKIN** [7] – que converte regras do snort para o padrão do iptables utilizando, também, o *match string*, como a ferramenta apresentada neste artigo. Porém, não dispõe de interface gráfica nem permite a inserção manual de regras a partir da própria ferramenta.

⁸ Falsificação do endereço IP de origem

6.2.3 Anti-Virus no Servidor de E-mails

A utilização deste tipo de solução (Figura 20) mostra-se eficaz, porém, só contempla pacotes cujas origens/destinos sejam relacionadas a e-mails (portas 25, 110 e 143).

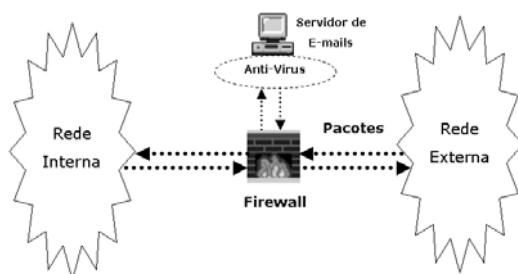


Fig. 20 – Uso de Anti-Virus no Servidor de E-mails

A ferramenta apresentada neste trabalho, inclusive, pode ser utilizada para este fim, bastando para isso, informar somente *strings* relacionadas a vírus, gerar o *script* correspondente, copiá-lo para o servidor de e-mails e executar o *script* localmente.

6.2.4 Soluções baseadas nos servidores e/ou estações

A utilização de soluções específicas para cada servidor somada à utilização de anti-vírus e *firewalls* pessoais nas estações (Figura 21) é uma solução de difícil manutenção. Esta dificuldade aumenta proporcionalmente ao número de servidores e estações da rede.

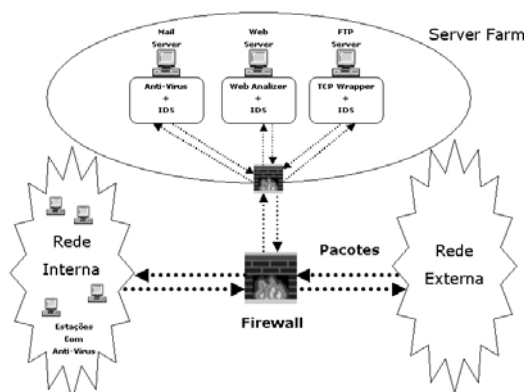


Fig. 21 – Solução Baseada nos Servidores e Estações

7 ANÁLISE DE IMPACTOS DE UTILIZAÇÃO

Embora a utilização da ferramenta apresentada restrinja a pesquisa de conteúdo aos pacotes destinados a portas específicas, a instalação em um *gateway* pode significar um retardo que pode se tornar expressivo de acordo com o fluxo de dados na rede.

Recomenda-se a avaliação do fluxo por porta antes desta instalação. O processamento adicional

gerado por cada novo filtro não deve extrapolar os recursos físicos da máquina (CPU e memória).

Com a utilização da ferramenta no *gateway*, comprovado um retardo no roteamento de pacotes tal que inviabilize o seu uso, pode-se optar pela geração de regras independentes para os servidores, funcionando similarmente à última solução apresentada no tópico anterior.

Pode haver um retardo causado por ataques de negação de serviço (DoS), isto é, alguém ou alguma aplicação pode estar enviando uma grande quantidade de pacotes maliciosos a uma porta específica contemplada pelos filtros *match string*, exigindo um maior processamento para analisar e descartar estes pacotes e consumindo, assim, os recursos da máquina.

Nestes casos, cabe ao administrador a análise dos *logs* dos filtros e a inserção de regras bloqueando momentaneamente o repasse de pacotes enviados pelo(s) endereço(s) de origem dos quais partiram os ataques e, em paralelo, a tomada de medidas de notificação e cobrança de atitudes aos responsáveis pelo *host* ou rede a qual pertence. As regras temporárias passariam a analisar os cabeçalhos dos pacotes partindo da origem suspeita, e não mais o conteúdo, diminuindo o processamento da máquina durante o processo investigativo.

8 CONCLUSÃO

A utilização da filtragem de *strings* diretamente como módulo do *firewall* é consitui-se em uma solução eficaz e apresenta resultados bem mais satisfatórios que as demais soluções citadas neste trabalho.

Com as novas funcionalidades incorporadas aos roteadores (como ACLs, NAT e redirecionamento de portas), um sistema de *firewall* necessita de novos componentes que o tornem robusto o suficiente de modo a justificar sua utilização em detrimento de um roteador. A filtragem de *strings* é um recurso candidato à incorporação definitiva aos componentes obrigatórios das novas gerações de *firewalls*.

O Iptables/netfilter possui vários outros módulos, também em fase de testes, que se enquadram nesta promissora categoria de novos recursos dos *firewalls*. A filtragem baseada em MAC Address, o suporte a características do IPv6, a incorporação de níveis de controle de QoS e a remarcação de campos *Diffserv* são algumas destas características.

Como continuação deste trabalho, uma linha de pesquisa interessante é a incorporação deste ambiente gráfico a uma das ferramentas já disponíveis de filtragem de pacotes (utilizando iptables) como Alfanega [13], o AGT [14], o IPMenu [15] ou o Firewall Builder [12].

REFERÊNCIAS BIBLIOGRÁFICAS

^[1] **RUSSEL**, Rusty. 2.4 Packet Filtering HOWTO
(<http://netfilter.samba.org/documentation/HOWTO/pt/packet-filtering-HOWTO.html>)

^[2] **RUSSEL**, Rusty. Linux Iptables HOWTO
(<http://www.telematik.informatik.uni-karlsruhe.de/lehre/seminare/LinuxSem/downloads/netfilter/iptables-HOWTO.html>)

^[3] **WRESKI**, Dave. Linux 2.4: Next Generation Kernel Security
(http://www.linuxsecurity.com/feature_stories/kernel-24-security-printer.html)

^[4] **WRESKI**, Dave. Linux Kernel 2.4 Firewalling Matures: netfilter
(http://www.linuxsecurity.com/feature_stories/kernel-netfilter.html)

^[5] **SYSCTL**. Filtering packets based on string matching
(<http://articles.linuxguru.net/view/125>)

^[6] **CHUVAKIN**, Anton. A Comparison of iptables Automation Tools
(<http://online.securityfocus.com/infocus/1410>)

^[7] **CHUVAKIN**, Anton. IPTables Linux Firewall with Packet String-Matching Support
(<http://online.securityfocus.com/infocus/1531>)

^[8] **CERT** Coordination Center. Design the firewall system
(<http://www.cert.org/security-improvement/practices/p053.html>)

^[9] **STEARNS**, William. Snort2iptables
(<http://www.stearns.org/snort2iptables/>)

^[10] **LARSEN**, Jason. Hogwash Snort-Based Scrubber (<http://hogwash.sourceforge.net>)

^[11] **RAMONI**. Pigmeat – Bloqueio por Firewall em Tempo Real
(<http://pigmeat.linuxinfo.com.br/>)

^[12] **KURLAND**, Vadim. Firewall Builder
(<http://www.fwbuilder.org/>)

^[13] **TOSTA**, Christian. Alfandega
(<http://alfandega.sourceforge.net/>)

^[14] **ROGERS**, Douglas C. AGT
(<http://sourceforge.net/projects/agt>)

^[15] **STES**, David. IPMENU – Netfilter/Iptables Rule Editor
(<http://users.pandora.be/stes/ipmenu.html>)