Ambiente Gráfico Auxiliar na Configuração de Filtro de Pacotes em um Ambiente Linux

Luiz Fernando Fernandes Fagundes[1], **Fabio Diniz Rossi**[1]

[1]Instituto Federal Farroupilha - Campus Alegrete (IFFar)
RS - 337 - Km 27 - Passo Novo - CEP 97555-000 - Alegrete - RS - Brazil

luiz.fagundes@aluno.iffar.edu.br, fabio.rossi@iffaroupilha.edu.br

Abstract. *Following the expansion of computing networks, security technologies are indispensable in protecting information and hardware resources in private networks. A secure, stable and effective technique available for GNU/Linux environments is the packet filtering system based on the Netfilter toolkit configured by iptables through lines of code, a task that requires technical knowledge and careful analysis. By designing a user-centric environment for configuration, we have identified the possibility of developing a graphical interface that helps to configure the rules by optimizing the time of its implementation and improving its management.*

Resumo. *Acompanhando a expansão das redes computacionais, as tecnologias de segurança são indispensáveis na proteção das informações e recursos de hardware nas redes privadas. Uma técnica segura, estável e eficaz disponível para ambientes GNU/Linux é o sistema de filtro de pacotes baseado no conjunto de ferramentas Netfilter, configurado pelo iptables através de linhas de código, tarefa que requer conhecimento técnico e criteriosa análise. Projetando um ambiente centrado no usuário para configuração, identificou-se a possibilidade de desenvolver uma interface gráfica que auxilie a configuração das regras otimizando o tempo de sua implantação e melhorando seu gerenciamento.*

1. Introdução

Atualmente, na rede mundial de computadores, existem incontáveis processos em andamento que são sigilosos, ou seja, somente a origem e o destino destes dados e informações devem ter conhecimento de seu conteúdo, no entanto, para que tais dados sejam protegidos torna-se imprescindível a aplicação de métodos e tecnologias flexíveis, versáteis e que tornem as redes privadas mais seguras contra ataques de *crackers* oriundos da rede externa.

O componente de filtragem de pacotes tem por função analisar todos os pacotes que trafegam na rede por meio de regras previamente configuradas, corrigindo e/ou minimizando vulnerabilidades existentes, proporciona maior flexibilidade e controle ao administrador, é extremamente útil em situações simples de segurança visto que atua no bloqueio global de tipos específicos de pacotes que trafegam entre redes e no bloqueio de serviços. Sua implantação é de baixo custo, causa pouco impacto no desempenho, fornece controle de acesso e serviços para toda a rede.

Acoplado ao *Kernel* dos Sistemas Operacionais GNU/Linux com versões 2.4 ou superiores o subsistema de filtro de pacotes nativo denominado *Netfilter* permite a

configuração de regras que atuarão no bloqueio ou liberação de tráfego de pacotes na rede, e, para executá-las corretamente, requer conhecimento técnico e experiência de quem as configura, pois podem ocorrer diversos tipos de erros durante a sintaxe das regras que seguem uma lógica e uma padronização.

Tomado o conhecimento destes parâmetros identificou-se a possibilidade de desenvolver uma aplicação *localhost* que auxilie na configuração e na aplicação dos filtros que atuarão no controle automático do tráfego da rede. Por meio de uma uma interface simples, amigável e intuitiva, esta apresentará opções de configuração dinâmica e objetiva, reduzindo assim o tempo de configuração de um *firewall iptables*.

2. Revisão Bibliográfica

Encontra-se dividida em três subseções com a finalidade de conceituar termos importantes para o trabalho, relacionar estudos semelhantes ou que validem a importância da utilização do filtro de pacotes como componente de segurança de redes e relacionar as tecnologias utilizadas na codificação da *interface* do *Check Filter*.

2.1. Fundamentação Teórica

Apresentam conceitos de autores sobre termos essenciais para o entendimento e desenvolvimento do trabalho.

2.1.1. *Firewall*

Um *firewall* para Chapman at al. (1995), consiste em um componente ou um conjunto de componentes que restringem o acesso de uma rede protegida e a internet, ou entre outros conjuntos de rede.

Para Cheswick et al.(2005), *firewall* é qualquer equipamento ou software que limita o acesso à rede.

2.1.2. *IPtables*

O autor Purdy (2004) define o *iptables* como um utilitário de linha de comando fortemente acoplado ao sub-sistema de filtragem de pacotes, disponível no *Kernel* dos sistemas *GNU/Linux* com versões 2.4 e superiores, um conjunto de ferramentas nativo denominado *Netfilter*.

2.1.3. *Cracker*

Os autores Furmankiewicz e Figueiredo (2000) conceituam um ataque *cracker* todas as ações que tem por objetivo a intenção de prejudicar a vítima.

2.1.4. Filtro de Pacote

Para autor Chapman (1992), um filtro de pacotes é um processo que atua no bloqueio ou liberação de pacotes de dados pela *interface* da rede, utilizando um conjunto de regras de

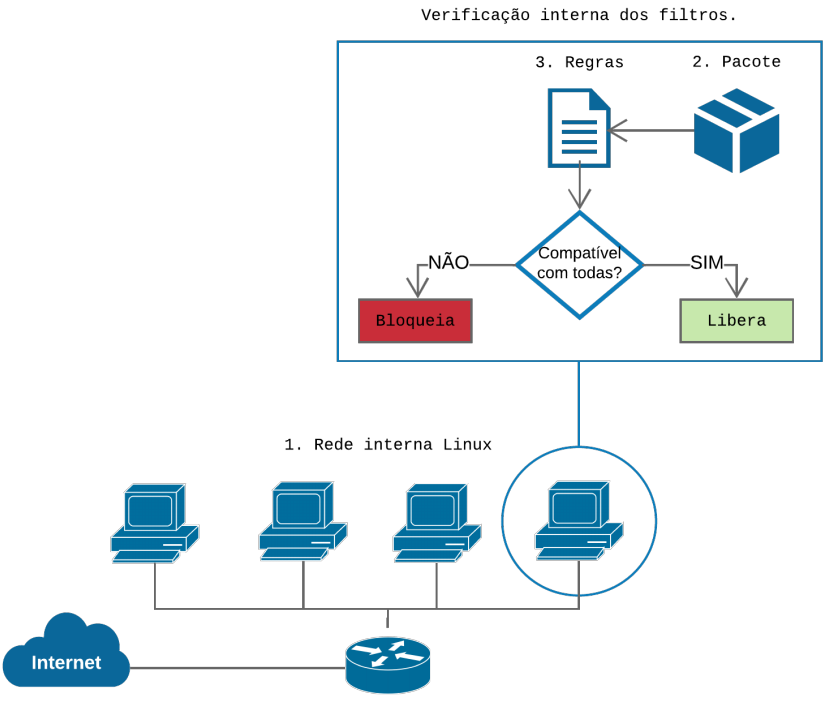
filtragem pré-configuradas e a análise é feita nas informações contidas no cabeçalho de cada pacote individualmente.

Cheswick e Bellovin (2005) ressaltam que a utilização de um filtro de pacotes é uma solução barata e útil na segurança de redes, pois permite filtrar pacotes com base nos endereços de origem ou destino ou nos números de porta.

Todos os pacotes que trafegam na rede são analisados um a um e ações serão aplicadas nos pacotes que atendam às restrições dos filtros, os pacotes que não se encaixem nas restrições seguirão seu fluxo de processamento normalmente para seu destino. À princípio, a filtragem de pacotes, atua nas camadas de rede e transporte, segundo o modelo *OSI*.

A Figura 1 exemplifica, de forma global, como o filtro de pacotes atuará na rede. Todo o tráfego de pacotes que circular na rede interna será verificado pelas *chains* configuradas pelo administrador em um *host*. Os pacotes oriundos da rede externa serão analisados pelos filtros e, de acordo com as regras implementadas, caso atendam às condições de configuração, chegarão ao *IP* de destino. Caso os pacotes não estejam em conformidade com a configuração, serão bloqueados.

Figura 1. Visão global do processo de filtragem de pacotes.



Fonte: elaborado pelo autor.

2.2. Trabalhos Relacionados

Atualmente, existem vários estudos comparativos sobre técnicas e metodologias em segurança de rede que apresentem flexibilidade em sua configuração e compatibilidades com outros recursos que se complementem, objetivando monitorar todo o fluxo de

tráfego na rede, sem que caminhos alternativos fiquem desprotegidos. Tais caminhos normalmente são as rotas usadas por ataques externos em busca da obtenção de informações sigilosas de formas obscuras para prejudicar a vítima.

Trabalhos comparativos entre soluções com *Netfilter* e Cisco ASA (dispositivo físico para segurança de redes) realizado por Silva (2013), que compara o desempenho de ambos em testes de penetração de segurança, avalia custos de investimento e tempo médio de implantação, onde buscam identificar qual é a melhor abordagem a ser tomada na escolha de um *firewall* que atenda as necessidades das organizações, levando em consideração o custo-benefício. Os resultados obtidos neste estudo mostram algumas vantagens do uso do *Netfilter/iptables*:

- Gratuidade no sistema de *firewalling*, pois o recurso é nativo dos sistemas *GNU/Linux*;
- Desempenho no bloqueio de pacotes de entrada e saída tão eficiente quanto ao Cisco ASA;
- Mantém arquivos de *log* detalhados de cada ataque;
- Por ser nativo do SO *GNU/Linux*, tem a capacidade de trabalhar com outros módulos de segurança como *VPN*, sem ter a necessidade de investimento financeiro.

O principal problema encontrado neste comparativo tem relação à configuração do *iptables*, que é realizada por linha de comando e requer conhecimentos técnicos da linguagem pelo administrador da rede.

Estudos realizados por Basile e Lioy (2015) adaptam versões conhecidas de modelo de filtros de pacotes atuantes em camadas de rede e transporte para que atuem a nível da camada de aplicação na análise comparativa em pares de regras em *firewalls*, com o objetivo de reduzir configurações conflitantes decorrente de uma configuração imprecisa. O modelo testado obteve resultados eficazes no gerenciamento de filtragem de conteúdos de expressões regulares em conexões *proxy HTTP*, podendo ser usada devido à semântica da política. Utilizando métodos desenvolvidos pelos autores do trabalho que consiste na modificação dos algoritmos que realizam o tratamento de tomada de decisões, o princípio básico do filtro de pacotes obteve um desempenho satisfatório nos testes em que foi submetido, mostrando-se uma ferramenta flexível e segura quando aplicada de forma correta. Este trabalho ainda propõe estendê-lo para outros contextos como o de *VPN*.

Este estudo visa explorar essa eficaz ferramenta para análise de fluxo de pacotes na rede interna, incorporando um método de configuração gráfico que seja qualitativamente efetivo na implementação dos filtros e oferecer uma *interface* focada no usuário.

2.3. Tecnologias

Nesta seção são conceituadas as tecnologias escolhidas para a codificação da *interface Check Filter* que impacta diretamente em sua arquitetura e componentes gráficos empregados, pois devem dar suporte para configuração de regras que foram selecionadas para configuração no ambiente gráfico.

2.3.1. *Netfilter*

O *Kernel* do Linux é o núcleo do sistema operacional GNU/Linux, que serve de base para o desenvolvimento, execução de outros programas e controle de *hardware*. Portanto, trabalha entre a camada de usuário e periféricos de *hardware*. Em sua estrutura, o *kernel* possui um sub-sistema que permite a configuração das tabelas de *firewall*, denominado *Netfilter*.

Dentro de sua estrutura, existem três tabelas que são importantes para seu funcionamento: *Filter*, *NAT* e *Meangle*. Estas tabelas organizam as cadeias de regras de acordo com sua estrutura e determinam a quais pacotes as regras serão aplicadas.

1. **Tabela *Filter***
   As regras contidas nesta tabela determinam a aceitação ou bloqueio de um pacote. Dentro desta tabela existem três cadeias:
   - *INPUT* - somente serão avaliados pelas regras previamente configuradas os pacotes destinados ao IP do computador de destino;
   - *OUTPUT* - pacotes avaliados dentro desta regra se limitam aos processos locais do computador;
   - *FORWARD* - somente são avaliados pacotes repassados pela máquina.

   As opções que são aplicáveis pela tabela *filter* são:
   - *REJECT* - quando esta regra é aplicada, todas as outras regras são ignoradas e o pacote é descartado;
   - *ACCEPT* - aceita o pacote, desde que o mesmo não seja avaliado posteriormente dentro da mesma tabela;
   - *DROP* - assemelha-se ao *REJECT*, mas não envia mensagem de erro ao remetente.
   - *LOG* - restringe-se em apenas criar registros sobre um pacote.

2. **Tabela *NAT***
   A tabela *NAT (Network Address Translation)*, realiza a tradução dos endereços que passam pelo roteador no qual essa opera, e divide-se em três cadeias:
   - *PREROUTING* - aplica as regras aos pacotes que entram no *firewall*, independente do seu destino.
   - *POSTROUTING* - são inseridas as regras capazes de modificar o pacote após o roteamento.
   - *OUTPUT* - similar ao *PREROUTING*, a diferença é que atua em pacotes oriundos de processos locais.

   As ações aplicadas por esta tabela são:
   - *SNAT* - realiza a troca dos endereços *IP* de origem;
   - *DNAT* - altera os endereços de *IP* de destino;
   - *MASQUAREDE* - faz o mascaramento de *IP*;
   - *REDIRECT* - redireciona o pacote para uma porta local.

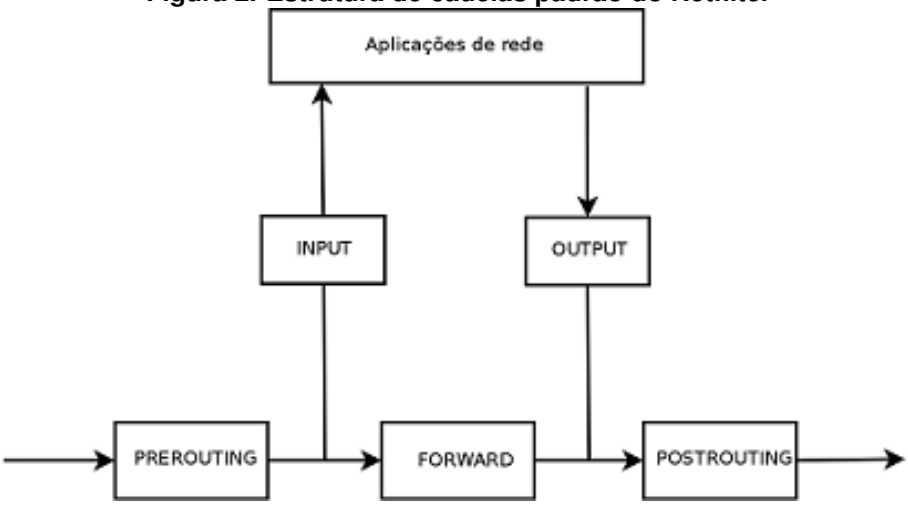
3. **Tabela MANGLE**
   Esta tabela tem por objetivo especificar ações especiais que devem ser aplicadas no tráfego que passa pelas cadeias, que ocorrem anteriormente aos *chains* das tabelas *FILTER* e *NAT*. O *Time of Service (TOS)* tem por objetivo priorizar o tráfego de pacotes na rede por meio da análise dos protocolos, é possível priorizar esse fluxo de pacotes de entrada e pacotes de saída. A prioridade de tráfego está organizado em quatro categorias e seus valores são:

   - Espera Mínima (*Minimize-Delay*), 16 ou 0x10;
   - Máximo Processamento (*Maximize-Throughput*), 8, ou 0x08;
   - Máxima Confiança (*Maximize-Reliability*), 4 ou 0x04;
   - Custo mínimo (*Minimize-Cost*), 2 ou 0x02;
   - Prioridade Normal (*Normal-Service*), 0 ou 0x00.

Por padrão o *TOS* vem ajustado com a prioridade normal que corresponde ao maior tempo de espera o tempo de espera mínima tem maior prioridade de tráfego.

A Figura 2 mostra o conjunto de cadeias padrão. Nessas estruturas, localizam-se as regras configuradas para análise dos pacotes. Cada cadeia é um ponto no caminho de um pacote ao entrar e sair de uma máquina e cada uma realiza o monitoramento dos pacotes de acordo com sua origem e destino, como segue: *PRE-ROUTING* analisa o tráfego ingressante na máquina; *FORWARD* analisa o tráfego que passa pela máquina; *POS-ROUTING* analisa todo o tráfego que sai da máquina; *INPUT* analisa o tráfego que tem como destino a própria máquina; *OUTPUT* analisa o tráfego gerado localmente.

Figura 2. Estrutura de cadeias padrão do *Netfilter*



Fonte: *Debian hand book.*[1]

[1]Disponível em: https://debian-handbook.info/browse/pt-BR/stable/sect.firewall-packet-filtering.html. Acesso em 15 de novembro 2018.

2.4. *Apache*

O *Apache*[2] é um servidor *Web* livre e extremamente difundido entre os usuários *Linux*, e distribuído pela licença *GNU*. Seu código-fonte pode ser estudado e modificado por qualquer pessoa. Atualmente é o servidor *Web* mais utilizado no mundo.

2.5. *PHP*

PHP[3] é acrônimo recursivo para *Hypertext Preprocessor*. É uma linguagem *open source* amplamente utilizada para desenvolvimento *Web*, pois pode ser embutida dentro do *HTML*.

2.6. *HTML*

HTML[4] é a sigla para *HyperText Markup Language*. Consiste em uma linguagem de marcação de texto utilizada na produção de páginas *Web*. Basicamente, trata-se de um conjunto de *tags* que servem para definir a forma que o texto será exibido e outros elementos da página HTML.

2.7. *CSS*

Cascading Style Sheets[5] é uma linguagem utilizada para definir a apresentação de documentos que adotem em seu desenvolvimento, linguagem de marcação como o *HTML*. Sua maior vantagem é separar o formato e o conteúdo de um documento, fazendo uso em conjunto do *HTML* e *CSS*.

2.8. *Bootstrap Studio*

O *Bootstrap Studio*[6] é um editor com ferramentas visuais para criação de páginas e *sites* responsivos. Possui um grande número de componentes internos, de fácil manipulação.

3. Metodologia

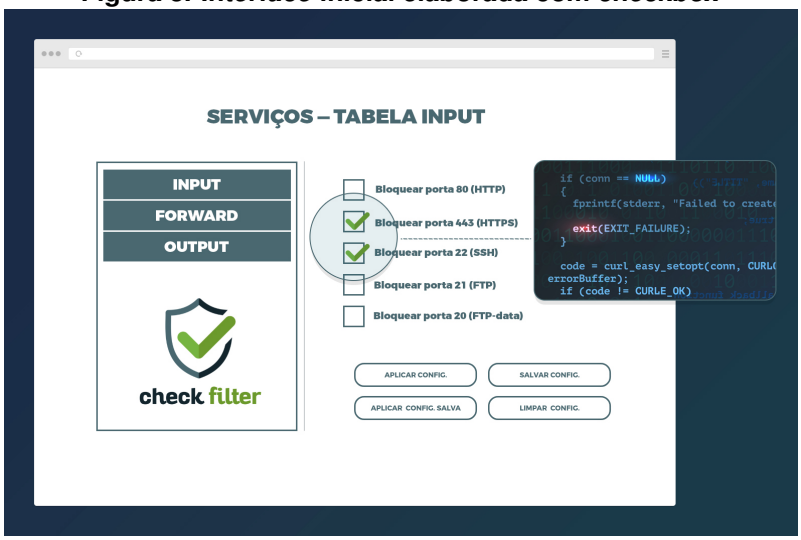
Finalizado a fase de revisão bibliográfica que inclui a análise de trabalhos relacionados, aprendizagem do conhecimento teórico sobre o filtro de pacotes *Netfilter*, sua utilização na prática e por fim a escolha de tecnologias para a criação da *interface*, iniciou-se então a fase de estudo sobre a melhor abordagem para ser empregada na arquitetura gráfica.

A Figura 3 mostra a primeira abordagem da *interface* onde as três tabelas principais do *Netfilter* encontravam-se em uma única tela e por meio do componente gráfico *checkbox* eram feitas as configurações de regras, porém testes iniciais utilizando essa abordagem indicaram que a *interface* era confusa e limitada na suas opções de configuração de regras.

---

[2]https://www.apache.org/foundation/
[3]http://php.net/manual/en/intro-whatis.php
[4]http://tableless.github.io/iniciantes/manual/html/
[5]http://tableless.github.io/iniciantes/manual/css/
[6]https://bootstrapstudio.io/

Figura 3. *Interface* inicial elaborada com *checkbox*



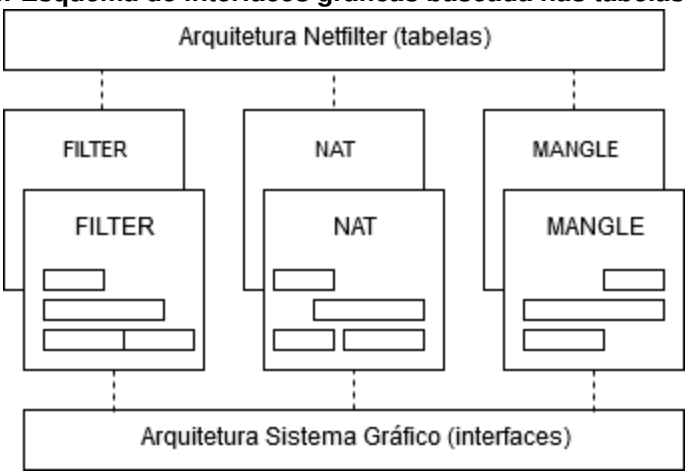
Fonte: elaborado pelo autor.

Diante deste desafio novas análises e *dashboards* foram desenhados a fim de encontrar uma arquitetura de *interface* que cumprisse o objetivo geral e específicos iniciais.

Optou-se então por aplicar um modelo que utilizasse a lógica estrutural da arquitetura *Netfilter* que está organizada em três tabelas principais *FILTER*, *NAT*, e *MANGLE*. Onde cada tabela é responsável por gerenciar regras que são específicas em cada uma.

Utilizando esse modelo, a aplicação foi criada seguindo essas divisões: três interfaces principais que correspondem a cada uma das tabelas *Netfilter* e cada uma destas interfaces disponibilizam meios de configuração que são específicos de determinada tabela.

Figura 4. Esquema de interfaces gráficas baseada nas tabelas *Netfilter*



Fonte: elaborado pelo autor.

A escolha das tecnologias para criação da *interface* foi um fator determinante para a criação da *interface*, pois deveriam oferecer o suporte necessário para a criação

da arquitetura escolhida, oferecer componentes gráficos que possibilitassem opções de configuração rápida e intuitiva e principalmente que se fosse capaz de se comunicar com a plataforma *Netfilter* por meio do *iptables*.

Optou-se então por desenvolver uma *interface local host* utilizando o servidor *Web Apache* devido as suas funcionalidades e sua compatibilidade com outros recursos e principalmente, por sua capacidade de executar códigos em *PHP* e pelo seu alto nível de confiabilidade em troca de informações entre cliente e servidor.

A linguagem de programação *PHP* foi escolhida, principalmente, devido à função *exec()* que irá executar o código externo ao ambiente gráfico, utilizado pelo administrador da rede. O código correspondente à combinação de componentes da *interface* gráfica será executado no *shell* do Linux por meio do *iptables*, agregando a regra especificada às *chains* das tabelas do *Netfilter*.

O *Bootstrap* possibilitou diferentes testes de layout na fase de codificação de telas sem que houvesse grande perda de tempo, pois esse *framework* possibilita a visualização da *interface* durante o processo de criação e permite exportar todos os componentes e o código do projeto em um arquivo *HTML*.

O *HTML* e o *CSS* permitiram toda a alteração necessária após a importação da estrutura gráfica construída no *Bootstrap* seja na alteração da estrutura ou no estilo da *interface*.

4. Plataforma Proposta

Empregando o uso das ferramentas mencionadas, este trabalho tem por finalidade oferecer um método alternativo na inserção das regras em um ambiente gráfico que realize a comunicação entre usuário-máquina, aplicando conhecimentos da Interação Humano-Computador, tendo como foco o usuário. Portanto, trata-se de um *front-end* externo aos recursos nativos do *Linux*, mas capaz de simplificar o método nativo de implementação de regras de pacotes.

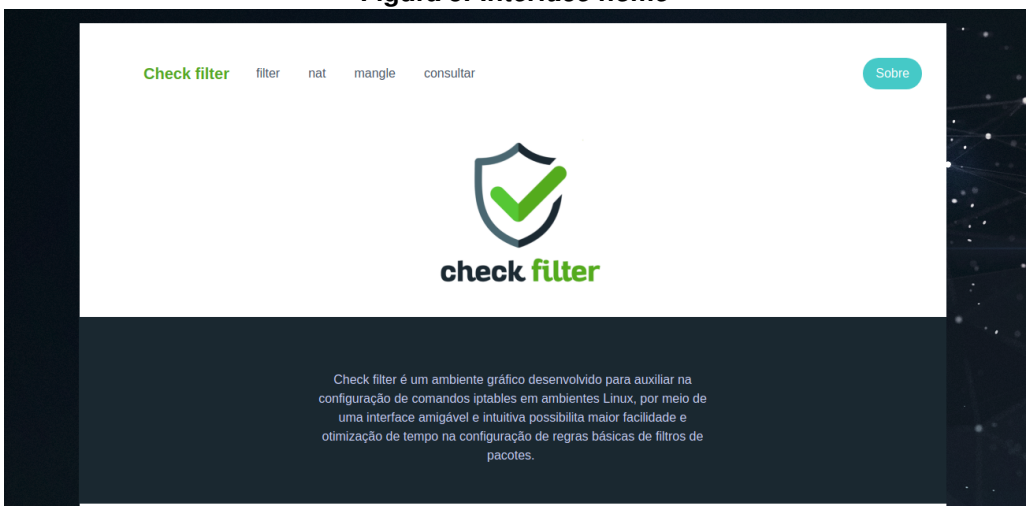
5. Resultados e Discussão

A interface gráfica desenvolvida é o produto da interação de diferentes tecnologias difundidas e utilizadas em diferentes tipos de projetos. O estudo da ferramenta *Netfilter* foi de vital importância, pois o conhecimento sobre sua organização, aplicabilidades e funcionamento são a base fundamental do protótipo que foi projetado.

Seguindo a arquitetura *Netfilter* projetou-se então toda a estrutura da ferramenta *Check Filter*, dividida em *interface home*, filter, nat, mangle e consulta.

Na *interface home*, Figura 5, é possível identificar facilmente essa organização, que de forma intuitiva indica ao administrador de rede onde encontram-se as funcionalidades pertinentes de cada tabela. Na parte superior há um *menu bar* de fácil acesso para navegação entre todas as interfaces da ferramenta *Check Filter*, este componente está presente em todas as *interfaces*.

Figura 5. *Interface home*



Fonte: elaborado pelo autor.

Essa organização foi definida e implementada porque evita erros de configuração de parâmetros, de regras que podem ocorrer quando utilizado o método de configuração via *shell script*. Tais erros são minimizados quando sua configuração realiza-se pela interface gráfica desenvolvida.

Exemplificando a comparação: Um administrador de rede deseja implementar um filtro de pacote que descarte a entrada de pacotes *TCP* na rede pela porta 445 (*Netbios*). A Figura 6 mostra o comando que deve ser executado para que a porta seja bloqueada.

Figura 6. Comando *iptables*

```
iptables -A INPUT -p tcp --dport 445 -j DROP
```

Fonte: elaborado pelo autor.

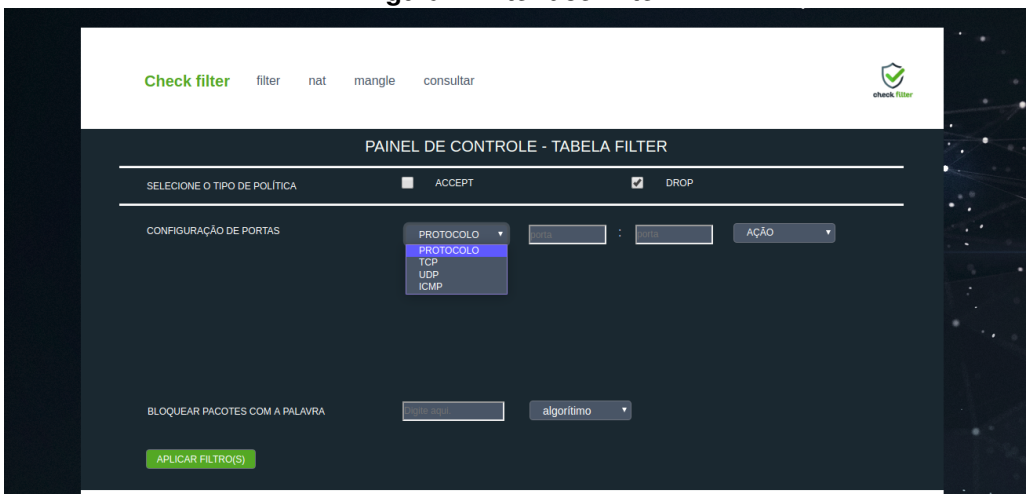
Todavia, em qualquer momento de configuração dessa regra pode acontecer erro de sintaxe e, caso ocorra um erro dessa natureza, o comando não funcionará e uma mensagem de alerta será emitida ao usuário. Um outro caso que pode ocorrer é algum equívoco de troca na especificação de parâmetros de porta ou de tabela. Nesse caso, erro algum será mostrado e a regra será incluída para filtrar pacotes em uma tabela que não encarrega-se por realizar tal operação. Ambos os casos irão fazer com que o *firewall* se comporte de forma diferente da especificada.

No ambiente gráfico, esse tipo de erro, seja de sintaxe ou troca de parâmetros, é minimizado ou anulado devido ao modo de configuração por componentes gráficos, pois apenas são disponibilizados componentes de configuração que são pertinentes à tabela. A Figura 7 demonstra componentes que são aplicados exclusivamente na tabela *FILTER*.

As funcionalidades desta *interface* incluem:

- Alterar a política da tabela;
- Configurar regras restritivas ou permissivas em uma única porta ou em uma faixa de porta especificando o protocolo que se deseja trabalhar;
- Bloquear pacotes específicos que contenham um padrão de sintaxe (palavra);

Figura 7. *Interface Filter*



Fonte: elaborado pelo autor.

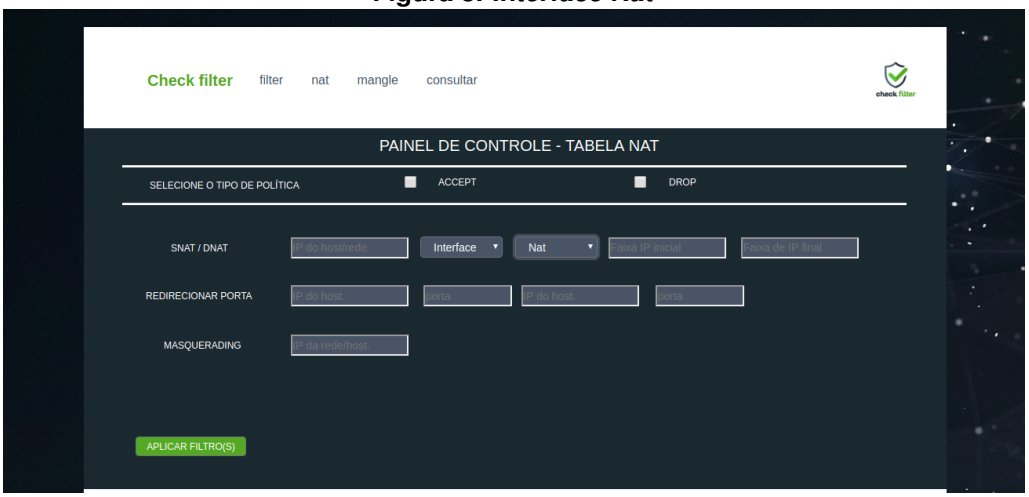
Como é possível identificar na Figura 7 quatro parâmetros (no caso de configuração de uma faixa de portas) ou apenas três (no caso de configuração de apenas uma porta específica), são necessários para que a regra da Figura 6 seja configurada: o protocolo, a(s) porta(s) e a ação.

A Figura 7 demonstra a facilidade de aplicação de regras, o que otimiza seu tempo de configuração em aplicações reais de segurança. Comumente, pode-se conter inúmeras linhas de regras inseridas em diferentes tabelas para diferentes fins, e quanto maior o nível de segurança maior será o número de linhas de regras nas *chains*. Isto também é proporcional em relação ao tempo para configurá-las.

Na *interface nat* Figura 8, que realiza a tradução de endereços de rede são disponibilizados os seguintes modos de configuração:

- Alterar a política da tabela;
- Realizar SNAT, tradução de endereços de origem para *ip's* fixos;
- Realizar DNAT, tradução de endereços de destino;
- Redirecionamento de portas, recurso utilizado para balanço de carga e *proxy* transparente;
- Mascarar *ip* específico ou de uma rede;

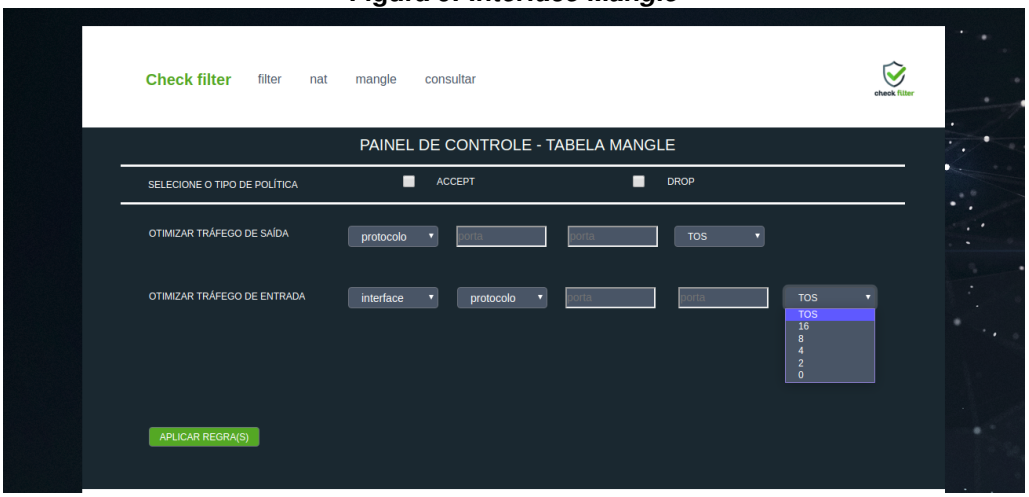
Figura 8. *Interface Nat*



Fonte: elaborado pelo autor.

Para a configuração da *interface mangle*, Figura 9, foi disponibilizado o otimização de tempo de serviços (*TOS*) de entrada e de saída basta a especificação dos parâmetros, como o tipo de protocolo que se deseja priorizar na rede.

Figura 9. *Interface Mangle*



Fonte: elaborado pelo autor.

A identificação da *interface* de consulta, Figura 10, ocorreu na fase de testes da *interface filter* pois após a criação de regras pela plataforma *Check Filter* havia a necessidade de validar a comunicação entre a *interface* gráfica e as tabelas *Netfilter*, essa validação ocorria de forma nativa,ou seja, pelo ambiente *shell script*, logo as seguintes ações foram adicionadas na plataforma:

- Consultar as regras por tabela;
- Excluir regras por tabela, excluir todas as regras ou ainda excluir uma regra específica;
- Salvar regras configuradas;

- Visualizar regras que foram salvas;

Regras salvas são restauradas sempre que o *host* for iniciado, caso contrário serão perdidas pois são alocadas em memória após sua configuração.

Figura 10. *Interface* de Consulta

Fonte: elaborado pelo autor.

6. Conclusões e Trabalhos Futuros

O sistema de filtro de pacotes é uma das primeiras ferramentas acionadas quando o tráfego de dados é estabelecido entre duas redes diferentes.

Esse sistema pode ser configurado para filtrar os campos de dados mais úteis, contidos no cabeçalho, como o tipo de protocolo, endereço *IP*, porta *TCP/UDP* e são alvos conhecidos em ataques, tais como a troca de endereços de origem de um pacote que pode ter dados sigilosos pelo endereço do próprio atacante.

Os testes experimentais com a interface apresentaram resultados satisfatórios como meio de interação com a arquitetura *Netfilter* e os componentes incorporados na ferramenta são adequados para configurações do filtro de pacotes.

A opção de configuração gráfica agrega simplicidade no processo de ajuste de regras permissivas ou restritivas. Pode ser usado em rotinas diárias de criação e edição de regras ou quando é necessário uma tomada de decisão rápida durante manutenção emergencial nas políticas de redes.

Os principais desafios encontrados neste trabalho foram: realizar a interpretação do método de configuração de regras na forma escrita e migrá-los para um modelo gráfico, sem que houvesse perda de flexibilidade de opções nos parâmetros que geram os comandos para aplicação de regras e conceder as permissões necessárias para configurar comandos que um usuário comum (que não seja superusuário) possa utilizar o utilitário *iptables* para configuração de filtro de pacotes.

Para trabalhos futuros, pretende-se agregar outras opções de configuração *iptables* na interface gráfica, bem como novas atualizações que possam ser lançadas, tornando a ferramenta mais completa.

Referências

Basile, C. and Lioy, A. (2015). Analysis of application-layer filtering policies with application to http. *IEEE/ACM Transactions on Networking (TON)*, 23(1):28–41.

Chapman, D. B. (1992). Network (in) security through ip packet filtering. In *USENIX Summer*.

Chapman, D. B., Zwicky, E. D., and Russell, D. (1995). *Building internet firewalls*. O'Reilly & Associates, Inc.

Cheswick, W. R., Bellovin, S. M., and Rubin, A. D. (2005). *Firewalls e segurança na Internet*. Bookman.

FURMANKIEWICZ, E. and FIGUEIREDO, J. (2000). Segurança máxima–o guia de um hacker para proteger seu site na internet e sua rede. *Rio de Janeiro: Campus*.

Purdy, G. N. (2004). *Linux iptables Pocket Reference: Firewalls, NAT & Accounting*. "O'Reilly Media, Inc.".

Silva, L. R. d. (2013). Estudo comparativo de soluções de firewalls: Netfilter/iptables e cisco asa.