CURSO DE BACHARELADO EM CIÊNCIA DA COMPUTAÇÃO

Brunno José Fagundes

SNORTIK - UMA INTEGRAÇÃO DO IDS SNORT E O SISTEMA DE *FIREWALL* DO MIKROTIK ROUTEROS

Santa Cruz do Sul, janeiro de 2016

CURSO DE BACHARELADO EM CIÊNCIA DA COMPUTAÇÃO

Brunno José Fagundes

SNORTIK - UMA INTEGRAÇÃO DO IDS SNORT E O SISTEMA DE *FIREWALL* DO MIKROTIK ROUTEROS

Prof. Me. Charles Varlei Neu
Orientador

Santa Cruz do Sul, janeiro de 2016

Brunno José Fagundes

SNORTIK - UMA INTEGRAÇÃO DO IDS SNORT E O SISTEMA DE *FIREWALL* DO MIKROTIK ROUTEROS

Trabalho de Conclusão II apresentado ao Curso de Ciência da Computação da Universidade de Santa Cruz do Sul, como requisito parcial para a obtenção do título de Bacharel em Ciência da Computação.

Orientador: Prof. Me. Charles Varlei Neu

Santa Cruz do Sul, janeiro de 2016

“Há três métodos para ganhar sabedoria:
Primeiro, por reflexão, que é o mais nobre;
segundo, por imitação, que é o mais fácil;
e terceiro, por experiência, que é o mais amargo.”
— CONFÚCIO

AGRADECIMENTOS

Agradeço a todos que, direta ou indiretamente, colaboraram para que eu alcançasse meus objetivos acadêmicos. Todo o trabalho, esforço e dedicação aplicado neste trabalho é em reconhecimento ao apoio que tive durante o processo de graduação.

Aos meus professores que, com dedicação, transferiram um pouco de seus conhecimentos. A todos os professores que tive, registro meu muito obrigado por toda a dedicação, empenho e boas lembranças que levarei comigo para toda a vida. Muitos de vocês serão lembrados com saudade, pois além de meus professores, foram grandes amigos. Meu muito obrigado a todos vocês. Agradeço ao meu orientador, Charles Neu, por toda a colaboração no desenvolvimento deste trabalho.

Meu agradecimento aos meus pais, Erli José Fagundes, em especial, minha mãe, Daisi Magda Antonio, que sempre me incentivou a continuar em frente, porém, infelizmente, não poderei compartilhar este momento "pessoalmente". Para você, minha mãe, dedico muito mais do que este trabalho, dedico toda minha trajetória de vida.

Meu sincero agradecimento a minha namorada Aline Jéssica Mainardi, a qual se manteve ao meu lado, mesmo separados por 265km. Obrigado por sempre ter incentivado, colaborado da melhor forma possível e pela compreensão durante todo esse período. Não foi fácil administrar tempo, distância e compromissos, mas por fim, conseguimos "piu". Te amo!

Aos amigos que souberam entender as limitações de tempo em decorrência do desenvolvimento deste trabalho. Meu sincero agradecimento a todos, sem citar nomes para não esquecer ninguém. Deixo aqui registrado o meu “Muito Obrigado” pelo apoio, em todos os momentos, principalmente nos mais difíceis. Tenham certeza que carrego todos vocês comigo, sempre!

RESUMO

Sistemas de Detecção de Intrusão, ou *Intrusion Detection System* - IDS - são ferramentas complementares à segurança de sistemas operacionais ou de redes de computadores. Estes sistemas têm por objetivo identificar tentativas de ataques as estruturas que estão destinados a monitorar e gerar alertas ao administrador do sistema/rede para que sejam tomadas as providências cabíveis para a solução dos problemas. Um IDS pode ainda ser integrado a outros mecanismos de proteção, como por exemplo, a um sistema de *firewall*, onde passa a atuar de forma combinada, podendo assim reagir a um ataque. Apesar da possibilidade de integração com um *firewall*, nativamente, isto somente é possível com a utilização de complementos, também conhecidos como *plugins*. Essa integração entre IDS e *firewall* também não é algo universal e compatível com todo e qualquer sistema de *firewall*, como é o caso do sistema Mikrotik RouterOS. Para realizar esta integração, se faz necessário um sistema intermediário que possibilite que, as detecções de ataques realizadas pelo IDS se transformem em regras de proteção no módulo de *firewall* do sistema Mikrotik RouterOS. O presente trabalho propõe a implantação do IDS SNORT como ferramenta de detecção de ataques e o desenvolvimento de um sistema intermediário que possibilite transformar os registros de detecção do IDS SNORT em regras de proteção junto ao *firewall* do sistema Mikrotik RouterOS. Os resultados mostram que este sistema pode ser utilizado para automatizar o processo de reação aos ataques identificados pelo IDS SNORT, sem a necessidade de intervenção direta do administrador do sistema, impedindo o acesso de *hosts* identificados como suspeitos.

Palavras-chave: Sistema de Detecção de Intrusão, SNORT, Mikrotik, Firewall, IDS, Intrusão de redes, Segurança de redes.

ABSTRACT

Intrusion Detection Systems (IDS) are tools designed to improve security in operating systems and computer networks. These systems are intended to identify attack attempts to the structure that they are designed to monitor, and generate alerts to the system/network administrators so that the appropriate action to solve the problems can be taken. An IDS can also be integrated with other protection mechanisms, such as a firewall, in order to react to an attack. Although the possibility of integration with a firewall, natively, it is possible only using addons, also known as plugins. This integration between IDS and firewall is not something universal and compatible with any and all firewall system, such as the Mikrotik RouterOS. To accomplish this integration, an intermediate system is proposed. This work proposes the implementation of SNORT IDS as an attack detection tool and the development of an intermediate system that uses the logs generated by SNORT IDS to create protection rules for the Mikrotik RouterOS firewall. The achieved results show that this system can be used to automate the reaction process to attacks identified by SNORT without the system administrator intervention, blocking suspect hosts identified.

Keywords: Intrusion Detection System, SNORT, Mikrotik, Firewall, IDS, Integration, Network intrusion, Network security.

LISTA DE ILUSTRAÇÕES

LISTA DE ABREVIATURAS

API	*Application Programming Interface*
DHCP	*Dynamic Host Configuration Protocol*
DNS	*Domain Name System*
DDoS	*Distributed Denial of Service*
DoS	*Denial of Service*
FPGA	*Field Programmable Gate Array*
FTP	*File Transfer Protocol*
HIDS	*Host Intrusion Detection System*
IDS	*Intrusion Detection System*
IPS	*Intrusion Prevention System*
MPLS	*Multi Protocol Label Switching*
NAT	*Network Translation Address*
NIDS	*Network Intrusion Detection System*
OSI	*Open System Interconection*
PHP	*Hypertext Preprocessor*
PPPoE	*Point-to-Point Protocol over Ethernet*
QoS	*Quality of Service*
SSH	*Secure Shell*
TCP/IP	*Transmission Control Protocol / Internet Protocol*
VPLS	*Virtual Private LAN Service*
VPN	*Virtual Private Network*

SUMÁRIO

INTRODUÇÃO

Com a necessidade de dispositivos estabelecerem comunicação entre si para realizar a troca de informações ou compartilhamento de recursos, redes de computadores estão sendo amplamente utilizadas e um dos maiores motivos é a popularização do acesso à Internet (SOUZA; LUCA, 2014). Entretanto, da mesma forma que o acesso à Internet é utilizado de forma adequada à finalidade para a qual foi criada, existem sistemas ou pessoas que promovem a má utilização desta rede com objetivo de prejudicar o seu funcionamento, utilizar os recursos disponíveis de forma prejudicial ou ilegal, ou ainda, adquirir informações sem a devida autorização (BHUYAN; BHATTACHARYYA; KALITA, 2014) (HUBBALLI; SURYANARAYANAN, 2014).

Com o objetivo de monitorar, identificar, registrar e informar os administradores destes sistemas e/ou redes de computadores quando algum tipo de atividade maliciosa ou suspeita ocorre, surgiram os sistemas de detecção de intrusão ou *Intrusion Detection System* - IDS (COMER, 1988) (ZAMAN; KARRAY, 2009) (NJOGU et al., 2013). Estes sistemas operam realizando a leitura de informações contidas nos pacotes transmitidos pela rede e comparam os dados coletados com uma coleção de regras que permitem identificar ataques conhecidos (JUNQI; ZHENGBING, 2008) (HOQUE et al., 2014). Estas regras podem conter assinaturas de ataques, ou seja, um conjunto de dados, ou ainda, um comportamento fora das características normais de operação da rede/sistema (CHAUDHARI et al., 2010) (ANAND, 2014).

Apesar de realizar a leitura e identificação de ataques em uma rede/sistema, registrar e alertar os responsáveis, originalmente, um IDS não é capaz de impedir ou bloquear um ataque. Da mesma forma, IDS's não são capazes de reparar os danos causados por um ataque, ficando esta função sob responsabilidade exclusiva do administrador do sistema/rede (NJOGU et al., 2013). Por outro lado, sistemas de *firewall* são responsáveis por garantir o controle do acesso à rede ou dispositivo, impedindo ou restringindo o acesso de dispositivos, de acordo com as regras de controle e políticas de segurança estabelecidas pelo administrador da rede/sistema (JUNQI; ZHENGBING, 2008).

Dentre as diversas soluções de IDS's existentes, desta-se o IDS SNORT (SNORT, 2015).

Este IDS é capaz de identificar diversos tipos de ataques, baseado tanto em assinaturas quanto por anomalias. Além disso, possibilita, através do uso de complementos, isto é, sistemas auxiliares que possibilitam realizar tarefas adicionais, uma integração com algumas soluções de *firewall*, como por exemplo o sistema IPTABLES. Contudo, sistemas como IPTABLES não são, por padrão, otimizados para esta tarefa e não possuem um *hardware* otimizado para tal finalidade, isto é, com um desenvolvimento voltado para a tarefa que se destina, diferente da solução Mikrotik RouterOS.

O sistema Mikrotik RouterOS é especializado no controle e gerenciamento de redes de dispositivos desenvolvido pela Mikrotik. Esta solução é executada de forma embarcada, utilizando para isso um *hardware*, também desenvolvido pela Mikrotik. Este sistema conta com um *firewall* que permite controlar todo o fluxo de dados transmitido pela rede. Contudo, não existe uma integração ou complemento que possibilite a integração entre o IDS SNORT e o sistema de *firewall* do Mikrotik RouterOS.

Observando esta lacuna entre as duas ferramentas de segurança, o presente trabalho apresenta uma forma de integração automática entre as duas soluções. Desta forma, é possível converter as detecções realizadas pelo IDS SNORT em regras de proteção junto ao sistema de *firewall* do Mikrotik RouterOS, ampliando a capacidade de reação aos ataques em uma rede de dispositivos, de forma automatizada, utilizando as duas ferramentas.

O restante deste trabalho de conclusão está organizado da seguinte maneira: O Capítulo 1 apresenta os conceitos necessários para a correta compreensão do mesmo. O Capítulo 2 reúne os trabalhos estudados com relação a classificação dos IDS's, melhorias e propostas de integração encontradas na literatura. Já o Capítulo 3 refere-se as principais tecnologias utilizadas no trabalho. O sistema desenvolvido é descrito no Capítulo 4. O Capítulo 5 mostra os testes realizados e os resultados obtidos. Por fim, a conclusão e considerações finais são apresentadas no Capítulo 6.

1 FUNDAMENTAÇÃO TEÓRICA

Neste capítulo é apresentado o referencial teórico com os conceitos necessários para a realização deste trabalho, visto que são fundamentais para a compreensão e desenvolvimento do mesmo. Além disso, são apresentadas as tecnologias utilizadas e alguns trabalhos com relação ao assunto encontrados na literatura.

1.1 Ataques de Intrusão

Um ataque de intrusão pode ser caracterizado, de acordo com Ahmed *et* al. (AHMED et al., 2009), como uma tentativa de utilizar recursos de um sistema de computador ou uma tentativa de acesso as informações sem autorização. Ataques desta natureza têm como objetivo corromper três dos componentes básicos da segurança de informação (BISHOP, 2004) (VACCA, 2013):

- **Autenticidade:** Trata-se da confiabilidade da informação no aspecto de sua veracidade. Ao afirmar que uma informação é íntegra estamos certificando que a mesma é oriunda de uma fonte confiável e o seu conteúdo não sofreu nenhuma alteração sem autorização.

- **Confidencialidade:** Consiste em manter a informação sob um determinado domínio de conhecimento, como por exemplo, restringi-la ao conhecimento de determinada informação ao ambiente de uma instituição.

- **Disponibilidade:** Refere-se à capacidade de utilizar um determinado recurso ou informação quando necessário, desde que haja permissão para esta ação.

Um ataque de intrusão pode explorar falhas de segurança de sistemas, aplicações, protocolos ou ainda falhas de configurações e de segurança. O padrão de rede do modelo *Transmission Control Protocol / Internet Protocol* - TCP/IP divide-se em quatro camadas: (i) Aplicação (ii) Transporte (iii) Rede e (iv) Dispositivo de Rede, ou ainda, *host* rede, sendo que cada uma delas

possui suas funcionalidades e serviços (COMER, 1988). Sendo assim, ataques de intrusão podem ocorrer nas diferentes camadas do modelo TCP/IP (ZAMAN; KARRAY, 2009) e, de acordo com a camada que atuam, esses ataques são categorizados em quatro grandes grupos, conforme apresentado na Figura 1:

CAMADA TCP/IP	CATEGORIA DO ATAQUE	COMPONENTE DE SEGURANÇA
APLICAÇÃO	EXPLORAÇÃO	CONFIDENCIALIDADE/AUTENTICIDADE
REDE	USUÁRIO PARA SUPER USUÁRIO	CONFIDENCIALIDADE/AUTENTICIDADE
TRANSPORTE	NEGAÇÃO DE SERVIÇO	DISPONIBILIDADE
HOST REDE	REMOTO PARA USUÁRIO	CONFIDENCIALIDADE/AUTENTICIDADE

Figura 1: Alguns tipos de Ataques e suas camadas de atuação

- **Exploração:** Ataques dessa categoria buscam explorar vulnerabilidades do sistema, aplicações instaladas sobre o sistema ou ainda falhas do usuário que permitam esse ataque.

- **Usuário para Superusuário:** Refere-se aos ataques que realizam tentativas de adquirir privilégios de Superusuários do sistema, possibilitando assim realizar qualquer tipo de operação.

- **Negação de Serviços ou *Denial of Service* - DoS:** Esse tipo de ataque tem como objetivo criar uma indisponibilidade de recursos ou sistemas. Para isso, um atacante gera muitas requisições de modo que os recursos ou os sistemas fiquem sobrecarregados, ao ponto de não conseguir mais atender as requisições de usuários legítimos, impedindo os acessos aos recursos ou informações. Este tipo de ataque pode ainda ser realizado de forma distribuída, sendo denominado *Distributed Denial of Service* - DDoS.

- **Remoto para Usuário:** Consiste em ataques que tem por objetivo se apropriar de sistema remotamente, explorando falhas de segurança para obter acesso ao sistema como um usuário legítimo ou administrador.

Como pode-se observar, os ataques de intrusão ocorrem das mais diversas formas e em todas as camadas do Modelo TCP/IP. Contudo, existem sistemas dedicados a identificar e detectar esses tipos de ataques: os IDS's.

1.2 Sistema de Detecção de Intrusão

Os IDSs têm como objetivo detectar, identificar e registrar essas ocorrências para que o administrador do sistema providencie uma solução para o ataque sofrido (NJOGU et al., 2013). Em 1987, Denning *et* al. (DENNING, 1987), apresentou um modelo para IDSs, onde descreveu um sistema capaz de identificar ataques independente de plataforma, aplicações ou tipo de ataque, sendo considerado um sistema de detecção de propósito geral. Além disso, o trabalho desenvolvido por Denning *et* al. deveria detectar ataques baseados em um perfil, métricas e modelos estatísticos. Porém, o modelo apresentado era baseado na detecção de intrusões para um dispositivo.

Já em 1990, Heberlein *et* al. (HEBERLEIN et al., 1990) apresenta uma proposta para um IDS ao nível de rede de computadores. Este IDS deveria ser capaz de detectar ataques baseados em um perfil de utilização da rede, identificando uma utilização que buscasse violar esse perfil, comprometendo a segurança. Após a apresentação destes dois modelos iniciais de IDSs, diversos outros modelos foram desenvolvidos. Desde então, observou-se a relevância do tema para a segurança das informações e, a partir deste momento, muitos modelos, inclusive projetos de interesses comerciais, foram desenvolvidos, como por exemplo, as soluções IDS SNORT e Bro (SNORT, 2015) (BRO, 2015) (PAXSON, 1998).

Um IDS pode ser classificado conforme o escopo de sua área de atuação, sendo assim, pode atuar tanto na identificação de ataques contra um dispositivo quanto na identificação de ataques dentro de uma rede de dispositivos. Desta forma, estes sistemas podem ser classificados em dois grandes grupos: (AL-JARRAH; ARAFAT, 2014) (BHUYAN; BHATTACHARYYA; KALITA, 2014)

- ***Host Intrusion Detection System* – HIDS:** Como HIDS, o IDS é responsável por monitorar a atividade de um dispositivo, isto é, busca identificar ataques ou eventos suspeitos que comprometam a segurança deste dispositivo, analisando apenas as informações transmitidas pela rede destinadas a este dispositivo, conforme ilustra a Figura 2.

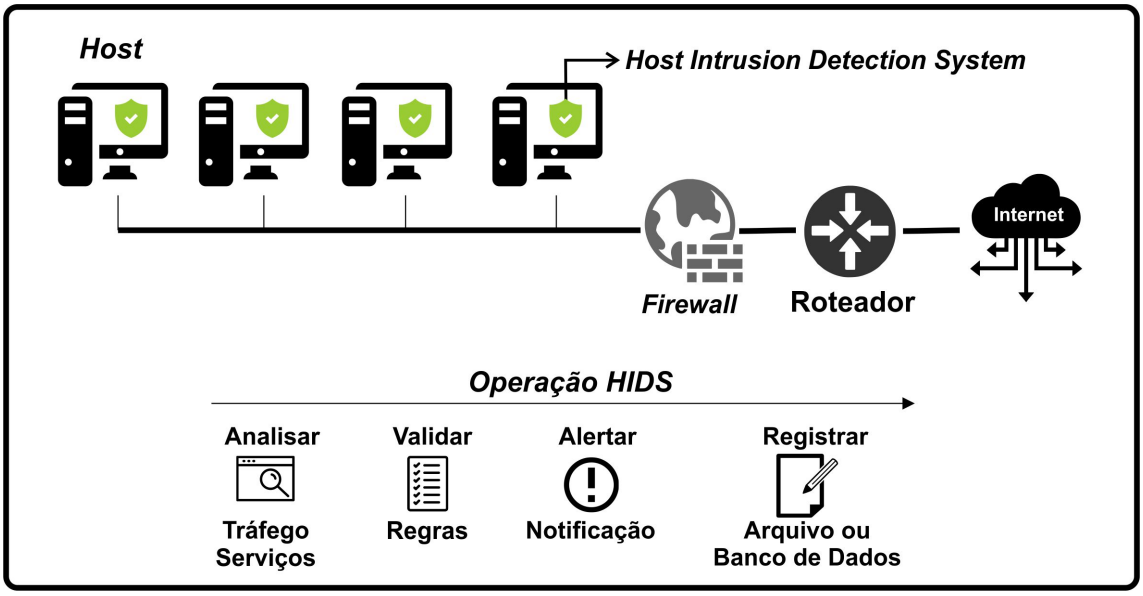

Figura 2: Modo de operação de um HIDS

- ***Network Intrusion Detection System* – NIDS:** Como NIDS, o IDS fica responsável por analisar o conteúdo transmitido na rede e busca identificar atividades suspeitas que ocorram. Neste escopo, o IDS é responsável por analisar atividades que envolvam todos os dispositivos presentes nesta rede, conforme mostra a Figura 3 (AL-JARRAH; ARAFAT, 2014) (BHUYAN; BHATTACHARYYA; KALITA, 2014).

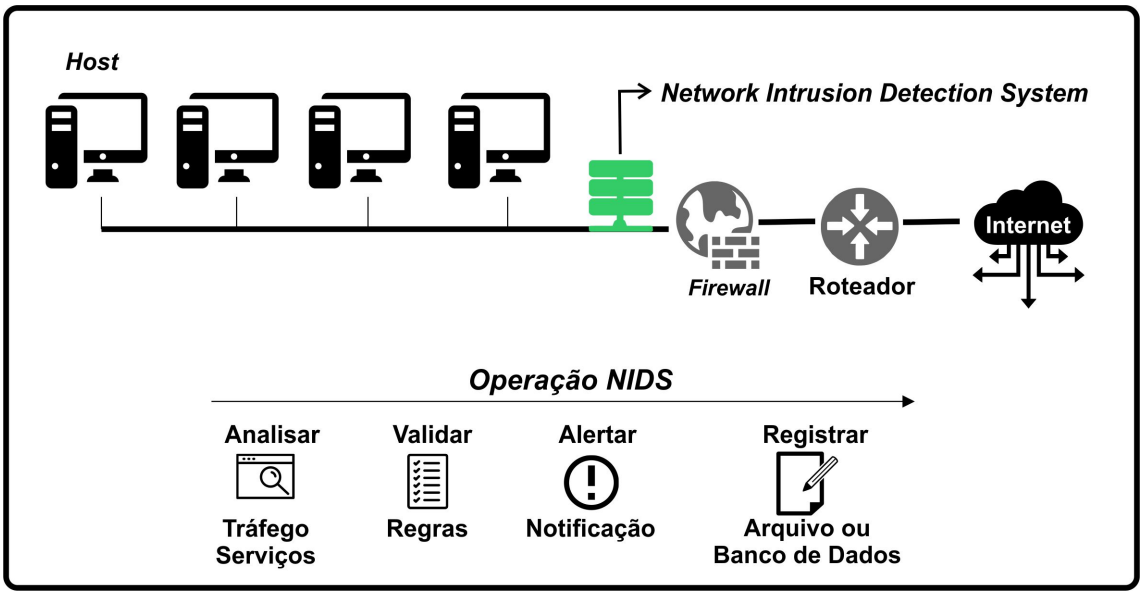

Figura 3: Modo de operação de um NIDS

Um IDS pode ainda ser classificado conforme a camada do modelo TCP/IP que opera. No trabalho apresentado por Zaman *et* al. (ZAMAN; KARRAY, 2009), esta classificação pode

ser realizada de acordo com o tipo de ataque pelo qual o IDS será responsável, uma vez que os diferentes tipos de ataques estão presentes em camadas distintas do modelo TCP/IP.

Além disso, essa classificação é justificada pelo fato de que um IDS é responsável por monitorar ataques específicos. Desta forma, ao monitorar uma camada do modelo TCP/IP em específico, o IDS pode ser otimizado para essa tarefa, apresentando resultados mais eficientes do que outro de propósito geral (ZAMAN; KARRAY, 2009).

Apesar de ser responsável por monitorar, tanto dispositivos em específico quanto redes de dispositivos, um IDS, por padrão, limita-se a três funcionalidades básicas:(i) monitorar, (ii) detectar e (iii) registrar os eventos. A responsabilidade de criar mecanismos ou realizar configurações para impedir ou interromper ataques é de responsabilidade do administrador do sistema ou rede. Entretanto, um IDS é capaz de reagir a uma ocorrência, interrompendo ou rejeitando conexões, ou ainda, disparando algum *script* para impedir o acesso à alguma informação (JUNQI; ZHENGBING, 2008).

A análise das informações realizada por um IDS pode ser realizada de duas formas distintas (SNORT, 2015):

- **Normal:** Quando opera em modo HIDS e analisa apenas as informações destinadas a um dispositivo em específico;

- **Promíscua:** Onde o IDS opera como NIDS e analisa todo o conteúdo transmitido entre os dispositivos da rede buscando identificar ataques.

No modo de operação promíscuo, deve ser observada a capacidade de análise do IDS, visto que o volume de informações processada tende a ser bem maior do que em uma análise em modo normal. Para isso é importante que o dispositivo responsável por operar como NIDS tenha um conjunto de hardware compatível com sua demanda. Caso contrário, poderá não ser capaz de processar um volume significativo de informações, o que, eventualmente, pode acarretar em não identificar um ataque (SNORT, 2015).

Para identificar um ataque, um IDS pode utilizar duas formas de detecção:

- **Baseado em Assinaturas:** Nesse modo o IDS utiliza-se de uma base de dados de ataques conhecidos. Para identificar uma tentativa de intrusão é analisado o conteúdo de cada pacote em busca de um conjunto de caracteres que permite identificar o ataque. Este conjunto de caracteres é denominado como "Assinatura do ataque" (SNORT, 2015).
- **Baseado em Anomalias:** Um IDS é capaz de identificar um ataque quando a ocorrência de algum evento é diferente de algum padrão considerado normal, por exemplo, alguma aplicação realizando uma tentativa de acesso não autorizado a um recurso do sistema (SNORT, 2015).

Quando um ataque é identificado, o IDS tem a responsabilidade de registrar essa ocorrência para alertar o administrador do sistema sobre o ocorrido. Para isso, pode utilizar diversos mecanismos, como por exemplo, registrar as ocorrências em arquivos, disparar alertas, enviar essas informações para o responsável, disparar algum *script*, ou ainda, armazenar essas informações em uma base de dados. Apesar de ter como objetivo principal identificar e registrar ataques, o IDS pode ainda reagir ao ataque, porém, essa característica reativa não é um padrão de sua funcionalidade (SNORT, 2015).

Ao registrar uma ocorrência, um IDS armazena uma coleção de informações referentes ao ataque, como por exemplo:

- Endereço IP de origem do atacante;
- Endereço IP do alvo;
- Protocolo de comunicação utilizado;
- Porta de comunicação utilizada;
- Conjunto de caracteres da assinatura do ataque;
- Data e hora da ocorrência;
- Classificação do ataque e sugestões de solução do problema.

Com esse conjunto de informações, um administrador é capaz de identificar os dispositivos envolvidos, quais as possíveis consequências deste ataque e como solucionar o problema

e evitar novos incidentes. Contudo, essa análise do conteúdo dos registros, identificação dos dispositivos e o tipo de ataque realizado é um processo, por padrão, dependente da intervenção humana. Existem ainda sistemas complementares aos IDS's que permitem uma organização e apresentação destas informações de forma mais prática, inclusive transformando-as em dados estatísticos.

Um IDS pode ser utilizado em conjunto com sistemas de controle de acesso à rede, isto é, sistemas de *firewall* (JUNQI; ZHENGBING, 2008). A combinação destas duas ferramentas de proteção, distintas e complementares, possibilita reagir aos ataques detectados. Entretanto, essa integração não é universal entre todos os IDS's e todos os sistemas de *firewall*. As funcionalidades de um sistema de *firewall* serão abordadas na Seção 1.3.

1.3 Sistema de *firewall*

Um sistema de *firewall* pode ser comparado, de acordo com Tanenbaum *et* al. (TANENBAUM; WETHERALL, 2010), com uma adaptação moderna das técnicas de segurança da Era Medieval, onde eram escavados fossos ao redor de castelos, restando apenas um ponto de passagem. Assim, os seguranças poderiam controlar o acesso, tanto de entrada quanto de saída do castelo, bem como restringir o acesso ao seu domínio, vistoriando cargas, pessoas, etc.

Em sistemas computacionais, um *firewall* tem a mesma função: controle das informações, permissões de acesso as redes de computadores, controlar o fluxo das informações, dentre outras funcionalidades (TANENBAUM; WETHERALL, 2010). Em uma rede de computadores, para buscar um certo grau de segurança, recomenda-se que estes sistemas sejam a única porta de entrada e saída. Desta forma o controle do acesso e aplicação de políticas de utilização da rede são controladas pelo *firewall*. Assim, busca-se garantir que todo o fluxo de informações será analisado e apenas o conteúdo considerado autorizado, seguro e confiável seja permitido.

A questão de controle e segurança das informações está diretamente relacionado às políticas de cada entidade ou organização. Assim, o *firewall* tem como objetivo assegurar que estas sejam respeitadas. Bishop (2004) define o sistema de *firewall* como um mediador no acesso à rede, o qual permite ou não certos tipos de acesso a ela baseando-se em políticas de segurança devidamente configuradas (BISHOP, 2004). A Figura 4 ilustra a utilização de um sistema de

firewall.

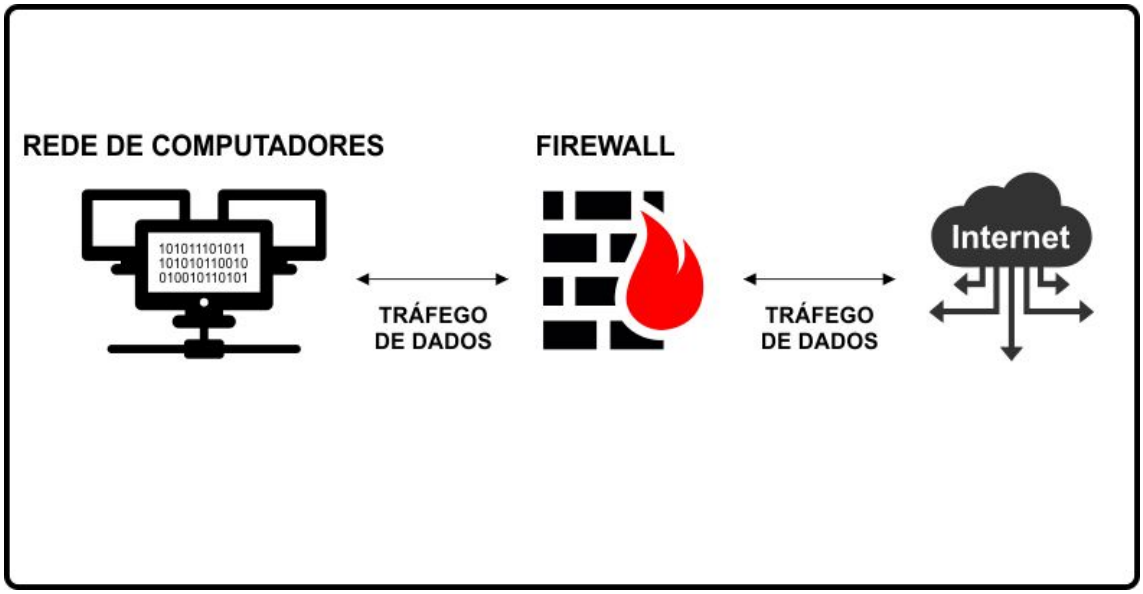

Figura 4: Modo de operação de um *firewall*

De acordo com Tanenbaum *et* al. (TANENBAUM; WETHERALL, 2010), a maioria das empresas possuem grande quantidade de dados confidenciais e o vazamento ou perda destas informações podem comprometer seus negócios. Além disso, existe o risco do uso não malicioso, mas sim, descuidado da rede, o que pode permitir o acesso de aplicações maliciosas que comprometem a segurança das informações e outros recursos. Para que eventos como estes não ocorram, o *firewall* é uma das ferramentas utilizadas para garantir que essas políticas sejam respeitadas (TANENBAUM; WETHERALL, 2010).

Após estudados os conceitos necessários para o entendimento deste trabalho, no Capítulo 2, são apresentados diferentes projetos e propostas encontrados na literatura que possuem relação com este projeto.

2 TRABALHOS RELACIONADOS

Este capítulo apresenta trabalhos encontrados na literatura relacionados aos IDS, suas aplicações e propostas de categorizar estes sistemas de acordo com a sua camada de atuação, considerando o modelo TCP/IP. Além disso, relaciona trabalhos onde a utilização combinada de IDS e sistemas de *firewall* são abordadas como soluções complementares de segurança, bem como algumas propostas de integração entre estas duas soluções.

2.1 Categorização do IDS

Dentre os trabalhos encontrados na literatura, o trabalho apresentado por Zaman *et* al. (ZAMAN; KARRAY, 2009) traz uma abordagem sobre sistemas de detecção de intrusão e sua relação com o modelo TCP/IP. Este trabalho refere-se a um estudo sobre a classificação dos IDS's de acordo com os tipos de ataques que irão monitorar e em qual camada do modelo TCP/IP eles ocorrem.

Zaman *et* al. destacam que, de acordo com a camada e tipos de ataques pelo qual o sistema será responsável por detectar e identificar, o IDS utiliza diferentes critérios para restringir o tráfego. O autor propõe que, se o IDS for utilizado exclusivamente para detectar ataques em uma das camadas proposta pelo modelo TCP/IP, este pode ser otimizado para essas detecções e ainda, pode-se ter uma redução da carga computacional, resultando em ganho de performance em sua capacidade de detecção e escalabilidade, comparando a um IDS de propósito geral que atue em diferentes camadas (ZAMAN; KARRAY, 2009).

Entretanto, apesar de apresentar propostas para categorizar os IDS's quanto a sua camada de atuação e propor que é possível desta forma uma otimização para estes sistemas aumentando suas performances no que diz respeito a capacidade de detecção, o autor não propõe nenhuma forma de integração destes sistemas aos *firewalls*. É importante ressaltar que Zaman *et* al. (ZAMAN; KARRAY, 2009) cita que IDS's podem ser utilizados como uma segunda linha de defesa. Sugere também que sua aplicação seja após um sistema de *firewall* e recomenda a combinação

entre essas duas soluções.

No trabalho de Chaudhari *et* al. (CHAUDHARI et al., 2010) é proposta uma classificação dos ataques de forma semi-supervisionada. Os autores apresentam um estudo onde classifica, através de um algoritmo especialista, os ataques identificados em dois grupos: maliciosos e normais. Ao classificar o ataque como malicioso, o algoritmo apresentado por Chaudhari *et* al. realiza a inclusão das informações do ataque à base de assinaturas do IDS.

Apesar de propor uma melhoria para os IDS's e destacar seus principais objetivos, Chaudhari *et* al. também não apresenta nenhum tipo de integração com algum tipo de sistema de proteção.

2.2 Propostas de otimização do IDS

Já o trabalho proposto por Valgenti *et* al. (VALGENTI; SUN; KIM, 2014) refere-se a um filtro de tempo real para ser aplicada ao tráfego da rede. Esse filtro tem por objetivo "separar" o tráfego seguro do tráfego considerado potencialmente suspeito, antes de ser encaminhado para análise do IDS.

Através desta abordagem, Valgenti *et* al. busca uma redução do volume de pacotes encaminhados para o IDS, aumentando a sua disponibilidade, reduzindo o número de pacotes descartados e aumentando o percentual de informações potencialmente suspeitas analisadas.

No que diz respeito a otimizar detecções, o trabalho proposto por Raghunath *et* al. (RAGHUNATH; MAHADEO, 2008) sugere um modelo de mineração de dados para detectar comportamentos anormal em uma rede, isto é, um comportamento com características e parâmetros diferentes da operação normal. O autor propõe uma abordagem onde, através de técnicas de mineração de dados, é possível identificar comportamentos associados a padrões de redes de forma automatizada.

Ainda sobre otimização de sistemas de detecção, é pertinente citar o trabalho desenvolvido por Lee *et* al. (LEE et al., 2007), onde é apresentada uma abordagem ao IDS implementado em *Field Programmable Gate Array* - FPGA. Com esta abordagem, Lee *et* al. propõem que IDSs implementados em FPGA podem obter um melhor desempenho de detecção do que soluções im-

plementadas em *software*, porém seu desempenho fica limitado à capacidade de processamento do *hardware*. Nesta abordagem, o sistema proposto foi, inclusive, comparado ao desempenho do IDS SNORT.

Por fim, é relevante destacar a abordagem realizada por Sherry *et* al. (SHERRY et al., 2015), ao problema de detecção de intrusão baseado em assinaturas quando o conteúdo está criptografado. Sherry *et* al. apresenta um sistema que opera na identificação de assinaturas mesmo tendo o conteúdo dos pacotes criptografados.

Para isso, o sistema realiza a comparação do conteúdo destes pacotes com assinaturas de ataques também criptografadas, utilizando diferentes esquemas de criptografia, buscando por expressões regulares nesse conteúdo. Assim, o sistema não realiza a quebra do sigilo destas informações e ao mesmo tempo permite a identificação de ataques.

Contudo, da mesma forma que os demais trabalhos estudados, os apresentados nesta seção não propõem uma solução para integrar as melhorias e otimizações aplicadas na fase de detecção a um sistema de proteção. Assim, mesmo que sejam atingidas estas melhorias na identificação de ataques, ainda existe a ausência de uma integração com um sistema que controle e proteja a rede de novos incidentes.

2.3 Integração de IDS e sistemas de *firewall*

O trabalho proposto por Huang *et* al. (HUANG; WANG; ZHU, 2010) apresenta uma integração entre um IDS e um sistema de *firewall*. O autor justifica esta integração quando deixa explícito as diferenças entre os sistemas, seus propósitos e suas limitações, mostrando que os sistemas, de fato, se complementam.

Huang *et* al. destaca que um sistema de *firewall* opera geralmente nas camadas de Enlace e Rede do Modelo *Open System Interconection* – OSI e não tem a capacidade de reconhecer conteúdo malicioso e comportamentos em outras camadas. Já o IDS, tradicionalmente, “escuta” o tráfego da rede buscando por comportamento suspeitos ou por ataques previamente identificados por assinaturas.

Porém, apesar de apresentar uma abordagem à integração entre os IDS (inclusive utilizando o IDS SNORT como ferramenta de detecção) e um sistema de *firewall* não especificado, este trabalho não contempla a integração ao sistema de *firewall* do Mikrotik RouterOS. Além disso, o sistema apresentado por Huang *et* al. propõe modificações no IDS para que seja possível esta integração. Ainda, os autores sugerem uma solução implementada em um mesmo dispositivo, onde compartilha recursos de hardware para realizar todas as funcionalidades de detecção e proteção.

Tal proposta não se aplica no caso do *firewall* do sistema Mikrotik RouterOS pelo fato de não ser possível implementar no mesmo dispositivo o IDS SNORT. Isso por que o sistema Mikrotik RouterOS, apesar de ser baseado no sistema operacional Linux, é uma solução proprietária e customizada.

Ainda sobre propostas de integração de IDS e sistemas de *firewall*, o trabalho de Lei-Jun *et* al. (LEI-JUN; HONG, 2010) apresenta uma proposta de integração, de fato, entre estas duas soluções de segurança (*firewall* e IDS), apontando suas vantagens, desvantagens e desafios.

Contudo, não menciona nenhum sistema em específico e conclui afirmando que este é ainda um modelo teórico. O trabalho proposto por Lei-Jun *et* al. sugere também que os dois sistemas se adaptem ao ambiente de rede. Entretanto, os autores afirmam que para alcançar esta integração e adaptação da solução ao ambiente, será necessário muita investigação e estudo sobre o assunto (LEI-JUN; HONG, 2010).

2.4 Objetivos

Considerando os trabalhos estudados na literatura, observou-se que o foco principal de estudo da maioria destes está relacionado a melhoria na detecção dos ataques. Apesar de alguns mencionarem que o IDS e o sistema de *firewall* se complementam, poucos tratam sobre esta integração, conforme mostra a Figura 5.

Trabalho	Integração	Automatização	SNORT	Mikrotik	Implementado
Zaman *et* al.	X	X	X	X	X
Chaudhari *et* al.	X	X	X	X	X
Valgenti *et* al.	X	X	X	X	X
Raghunath et *al.*	X	X	X	X	X
Lee *et* al.	X	X	X	X	X
Sherry *et* al.	X	X	X	X	X
Huang *et* al.	✔	✔	X	X	X
Lei-Jun *et* al.	✔	✔	X	X	X
Este Trabalho	✔	✔	✔	✔	✔

Figura 5: Comparativo entre trabalhos estudados e trabalho proposto

Como pode-se observar na Figura 5, mesmo aqueles que apresentam uma proposta de integração entre as duas ferramentas, não contemplam o sistema Mikrotik RouterOS. Desta forma, percebe-se a ausência de uma solução que integre detecção e proteção utilizando estas duas ferramentas. Neste sentido, é apresentada uma abordagem de integração do sistema de detecção de intrusão SNORT e o um sistema de proteção utilizando o *firewall* do sistema Mikrotik RouterOS.

Além disso, a proposta deste trabalho busca desenvolver uma solução que realize as tarefas de forma automatizada, sem a intervenção do administrador, para realizar a verificação dos registros do NIDS SNORT e para criação das regras de proteção junto ao sistema de *firewall* do Mikrotik RouterOS.

Com este objetivo, este trabalho propõe um sistema intermediário que utiliza os registros de detecção do IDS SNORT como fonte de informação para gerar regras de proteção em um sistema de *firewall* do sistema Mikrotik RouterOS. O desenvolvimento deste trabalho é apresentado no Capítulo 4, bem como o ambiente de desenvolvimento, arquitetura do projeto e a modelagem aplicada ao projeto.

3 TECNOLOGIAS ENVOLVIDAS

Este capítulo apresenta as tecnologias utilizadas para desenvolver este trabalho. Na Seção 3.1 é abordado o sistema de detecção de intrusão utilizado e sua operação. Já a Seção 3.2 refere-se ao sistema Mikrotik RouterOS, suas características e funcionalidades.

3.1 Sistema de Detecção de Intrusão SNORT

O sistema SNORT foi criado por Martin Roesch e é considerado o IDS *Open Source* mais difundido, de acordo com o site oficial do projeto, com mais de 4 milhões de *downloads* realizados e aproximadamente 500.000 usuários registrados em sua comunidade web (SNORT, 2015). É interessante ressaltar que o IDS SNORT é referenciado e utilizado em grande parte dos trabalhos estudados para desenvolver este trabalho, como por exemplo nos trabalhos de Al-Jarrah *et* al., Valgenti *et* al. e Lee *et* al. (AL-JARRAH; ARAFAT, 2014) (VALGENTI; SUN; KIM, 2014) (LEE et al., 2007).

Além disso, conta com uma base de dados de regras de detecção com mais de 36.500 regras. Essa base de dados também pode ser utilizada por outras soluções, como por exemplo, o sistema *Bro Network Security Monitor*, desenvolvido e mantido por Vern Paxson em parceria com o *International Computer Science Institute* e *National Center for Supercomputing Applications* (BRO, 2015) (PAXSON, 1998). Apesar disso, a solução SNORT possui uma grande e vasta documentação disponível e uma ampla variedade de complementos, o que aumenta a sua integração com outras soluções e viabiliza a integração com o sistema proposto.

É importante destacar que atualmente, o projeto do IDS SNORT conta com um grupo de pesquisa denominado TALOS, que tem por objetivo trabalhar de forma proativa para descobrir e solucionar problemas relacionados à *malwares*, vulnerabilidades, ataques de intrusão e atividades *hacking*. Este grupo é composto por membros renomados na área de segurança de redes e conceituadas empresas da área de segurança, conforme apresentado no próprio site do projeto. Maiores detalhes sobre este grupo de pesquisa podem ser encontradas no endereço

https://www.snort.org/talos (SNORT, 2015).

3.1.1 Modos de Operação

O IDS SNORT pode operar tanto como um HIDS quanto como NIDS. Em ambos os casos o sistema analisa os pacotes que são trafegados na rede de computadores, comparando o conteúdo encontrado com uma base de assinaturas de ataques ou através de identificação de alguma anomalia na operação da rede. Estas assinaturas são conjuntos de informações/caracteres que permitem ao IDS identificar pacotes que fazem parte de uma tentativa de intrusão. Através do comparativo do conteúdo destes pacotes com as informações contidas em sua base de assinaturas, é possível identificar com precisão possíveis tentativas de ataques à rede de computadores ou aos dispositivos . Já as anomalias são comportamentos de operação de uma rede fora de um limite aceitável, comparada aos parâmetros de operação normal da rede, como por exemplo volume de dados trafegado ou número de pacotes transmitidos (SNORT, 2015).

3.1.2 Arquitetura de Operação

A operação do IDS SNORT é realizada em etapas, ou seja, é necessário: (i) capturar um pacote de dados na rede, (ii) abrir (decodificar) este pacote, (iii) analisar as informações contidas neste pacote, (iv) comparar estas informações com as assinaturas de ataques existentes na base de dados do IDS e, caso detectado um ataque, (v) realizar o registro e (vi) alertar o administrador do sistema. Para realizar este processo o IDS SNORT divide sua operação em 3 subsistemas primários, conforme apresenta a Figura 6 (SNORT, 2015).

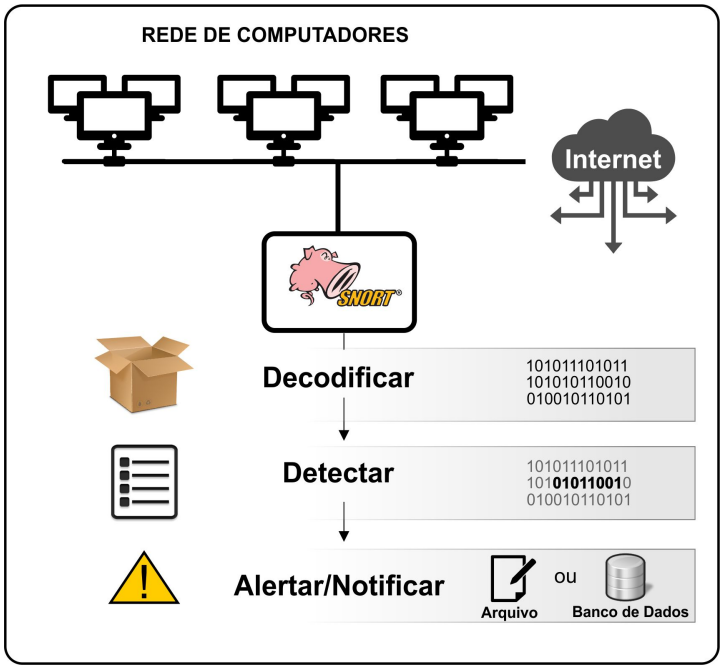

Figura 6: Fluxo de operação do IDS SNORT

Cada etapa ilustrada na Figura 6 "prepara" as informações a serem processadas pela etapa seguinte. Para realizar este processo, o IDS SNORT opera da seguinte forma:

- **Decodificador de Pacotes:** É responsável por decodificar os pacotes capturados de acordo com a camada do Protocolo TCP/IP que se encontram. Além disso, é também encarregado de inserir ponteiros nos pacotes decodificados para que estes sejam analisados pelo subsistema de detecção;

- **Engenharia de Detecção:** Tem como objetivo analisar as informações dos pacotes e compará-las às assinaturas dos ataques conhecidos pelo IDS SNORT. Caso seja detectado um ataque, é disparado o subsistema de registro e alerta;

- **Subsistema de registro e Alertas:** É encarregado por registrar os eventos e disparar os alertas configurados no sistema. O IDS SNORT permite registrar os eventos em arquivo texto, em formato syslog ou ainda em banco de dados. Os dados inseridos nesses registros podem conter informações básicas, como por exemplo, apenas endereços de origem e destino e tipo de ataque identificado, ou então completas, podendo conter inclusive informações do conteúdo do pacote analisado pelo sistema.

Apesar de identificar eventos de ataques com eficiência, o IDS SNORT não está imune

a problemas no processo de detecção. Um dos principais problemas que podemos encontrar é o caso de detecção de “falsos positivos” e “falsos negativos”. Identificar um “falso positivo” significa reconhecer um acesso legítimo como uma intrusão. Já o “falso negativo” é quando a ferramenta permite um ataque e não o identifica considerando que o mesmo é uma acesso legítimo, quando na verdade está ocorrendo um ataque ao sistema (SNORT, 2015) (HUBBALLI; SURYANARAYANAN, 2014).

No que diz respeito à compatibilidade, o IDS SNORT é compatível com diversos sistemas operacionais, podendo ser utilizado em sistemas Windows, MAC, Linux, Solaris, UNIX e derivados UNIX. Considerando as formas possíveis de registro, pode-se considerar uma solução multi-plataforma, a qual permite uma interação entre os diferentes sistemas (SNORT, 2015).

É importante salientar que, através da utilização de complementos, além dos formatos de registro mencionados, é possível armazenar as informações de registro de eventos em base de dados, como por exemplo, MySQL e PostgreSQL, aumentando o poder de análise sobre os dados registrados (SNORT, 2015).

No que diz respeito à detecção, permite ainda que o usuário do sistema desenvolva suas próprias regras de detecção. Isso permite ao administrador do sistema uma customização ou personalização de acordo com as características do ambiente em que está operando, ampliando ou refinando o poder de detecção do IDS (SNORT, 2015).

Apesar de ter como especialidade a detecção, o IDS SNORT permite ainda ser configurado para reagir aos ataques. Para isso, o sistema é configurado em modo de Prevenção de Intrusão ou *Intrusion Prevention System* - IPS. Desta forma, o IDS, quando identifica algum ataque, reage enviando, por exemplo, respostas de REJECT para o endereço que originou o ataque (SNORT, 2015).

Contudo, vale ressaltar que apesar de ser capaz de reagir a um ataque, este modo de operação pode permitir ocorrências de falsos positivos e falsos negativos, o que pode impactar diretamente no desempenho da rede se não houver nenhum tipo de tratamento para evitar estas ocorrências. Além disso, é pertinente recordar que o IDS possui uma capacidade limitada de analisar o conteúdo transmitido pela rede (SNORT, 2015) (BUL’AJOUL; JAMES; PANNU, 2013). Neste caso, seu desempenho pode vir a ser reduzido, pois acumula também a função de reagir ao

ataque, a qual não é sua especialidade.

Tais características combinadas com as diversas formas possíveis de registrar os eventos mostram-se compatíveis com a proposta de integração deste projeto. Vale ainda mencionar que os complementos disponíveis para melhor armazenamento das informações de registro do IDS SNORT viabilizam este trabalho. Devido a todas estas características, popularidade e flexibilidade de aplicação do IDS SNORT, este sistema foi eleito como ferramenta de detecção para o desenvolvimento deste projeto.

3.2 Mikrotik RouterOS

Mikrotik RouterOS é um sistema operacional desenvolvido e mantido pela Mikrotik (MIKROTIK, 2015), uma empresa fundada em 1995, sediada em Riga, capital da Letônia. O Mikrotik RouterOS é um sistema dedicado a realizar gerenciamento e controle de redes de computadores. Para isso, dispõe de diversos recursos, como por exemplo, roteamento de redes de computadores, controle e filtro de tráfego e conteúdo e sistema de proteção para estas redes. Além de recursos para controle e gestão de redes, o Mikrotik RouterOS é o sistema operacional embarcado no hardware desenvolvido pela Mikrotik, denominado RouterBOARD. A Figura 7 apresenta alguns dispositivos e componentes desenvolvidos pela Mikrotik.

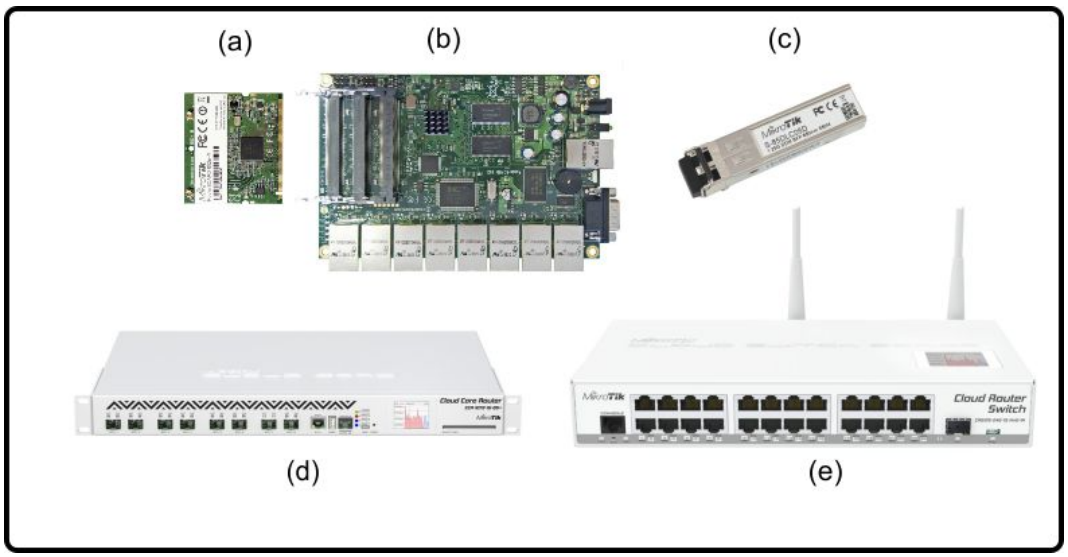

Figura 7: Dispositivos desenvolvidos pela Mikrotik
Fonte: Mikrotik

Na Figura 7 pode-se observar (a) um cartão de expansão de conexão sem fio de tecnologia

802.11 a/n ou 802.11 b/g/n utilizado em conjunto com a (b) Routerboard RB 493, composta por slots de expansão para cartões sem fio e ainda, 8 interfaces *ethernet* Gigabit e outras. Já a imagem (c) apresenta um adaptador para conexões ópticas, que pode ser utilizada nos dispositivos (d) e (e). A imagem (d) apresenta uma RouterBOARD de alto desempenho com suporte a 72 unidades de processamento e 8 portas ópticas de 10 Gigabits. Por fim, a imagem (e) é apresentado um modelo composto por conexões de rede se fio, redes *ethernet* Gigabit e conexões de fibra óptica em um mesmo dispositivo, todos gerenciados pelo mesmo sistema RouterOS, demonstrando a flexibilidade e robustez deste sistema.

No que diz respeito ao *hardware* desenvolvido pela Mikrotik, uma RouterBOARD pode suportar processadores com múltiplos núcleos de processamento ou por múltiplas unidade de processamento no mesmo hardware. No que tange a conectividade, uma RouterBOARD pode suportar, por exemplo, em um mesmo dispositivo, as recentes interfaces de rede de 10 Gigabits, interfaces de rede sem fio compatíveis com os padrões 802.11 a/b/g/n e ainda o recente padrão a/c (MIKROTIK, 2015).

O sistema Mikrotik RouterOS dispõe de um *firewall* que permite filtrar os pacotes transmitidos pela rede, fornecendo funções de segurança. O *firewall* suporta a função de *Network Translation Address* – NAT, o que previne o acesso não autorizado direto à redes conectadas em um mesmo roteador. O *firewall* do RouterOS permite ainda marcação de pacotes de acordo com parâmetros estabelecidos pelo administrador e tratamento destes pacotes de forma diferenciada pelo sistema, como por exemplo, transmissão com prioridades diferenciadas, velocidades de transmissão diferenciada, ou tratamento de rotas específicas deste conteúdo (MIKROTIK, 2015).

O *firewall* do Mikrotik RouterOS conta também com um filtro de conteúdo baseado em endereços de destino e origem específicos, lista de endereços IP, podendo ser estática ou dinâmica, filtro por padrão em seu conteúdo e/ ou ainda, expressões regulares. Além destes parâmetros, o *firewall* do Mikrotik RouterOS pode ainda aplicar filtros baseados em portas específicas ou entre um escopo de portas. Esses recursos permitem uma ampla granularidade de proteção à rede de computadores, permitindo o controle de acesso através da utilização de diversos parâmetros (MIKROTIK, 2015).

Além dos recursos mencionados anteriormente, conta com recursos de Servidor *Web-Proxy*, *Quality of Service*-QoS, Serviço de *Hotspot*, *Dynamic Host Configuration Protocol-*

DHCP, *Point-to-Point Protocol over Ethernet* -PPPoE, *Virtual Private Network*-VPN, *Multi Protocol Label Switching*-MPLS, *Virtual Private LAN Service*-VPLS, etc. Como pode-se observar, o RouterOS é um sistema completo para finalidade que se destina, podendo ser amplamente utilizado para as mais diversas funcionalidade de controle, gerência e proteção de redes de computadores de pequeno, médio e grande porte (MIKROTIK, 2015).

O sistema RouterOS conta ainda com uma lista de ferramentas para auxiliar o administrador do sistema, como por exemplo, teste de resposta (PING), ferramenta para testar a largura de banda entre dois pontos, analisador de pacotes (*Sniffer*), analisador de conexões (*Torch*), agendador de tarefas para execução automática de *script*, entre outras (MIKROTIK, 2015).

O sistema Mikrotik RouterOS acompanha nativamente os dispositivos RouterBOARD. Apesar de contar com uma *hardware* dedicado, otimizado e desenvolvido para esta finalidade, pode ser instalado em um dispositivo com arquitetura PC(x86), transformando um simples computador em um poderoso sistema de roteamento e controle de redes de computadores. Uma das principais características dos produtos Mikrotik é a capacidade de combinar diferentes tecnologias de transmissão em um mesmo dispositivo, permitindo, por exemplo, adicionar cartões de diferentes tecnologias de transmissão sem fio a um mesmo dispositivo sendo todos gerenciados pelo mesmo sistema (MIKROTIK, 2015).

Para configuração e administração do sistema, o Mikrotik RouterOS dispõe de uma ferramenta fornecida pela própria Mikrotik para realizar a configuração do sistema, denominada Winbox. Além do acesso via ferramenta WINBOX, permite sua configuração do acesso via Telnet e *Secure Shell* - SSH, onde disponibiliza um console para execução de comandos de modo em linha de comando. Além destes modos de acesso, o Mikrotik RouterOS conta com uma interface web, muito eficiente e de fácil utilização, disponibilizando grande parte das configurações disponíveis em outros modos. A Figura 8 apresenta a tela de configuração do sistema utilizando a ferramenta WINBOX (MIKROTIK, 2015).

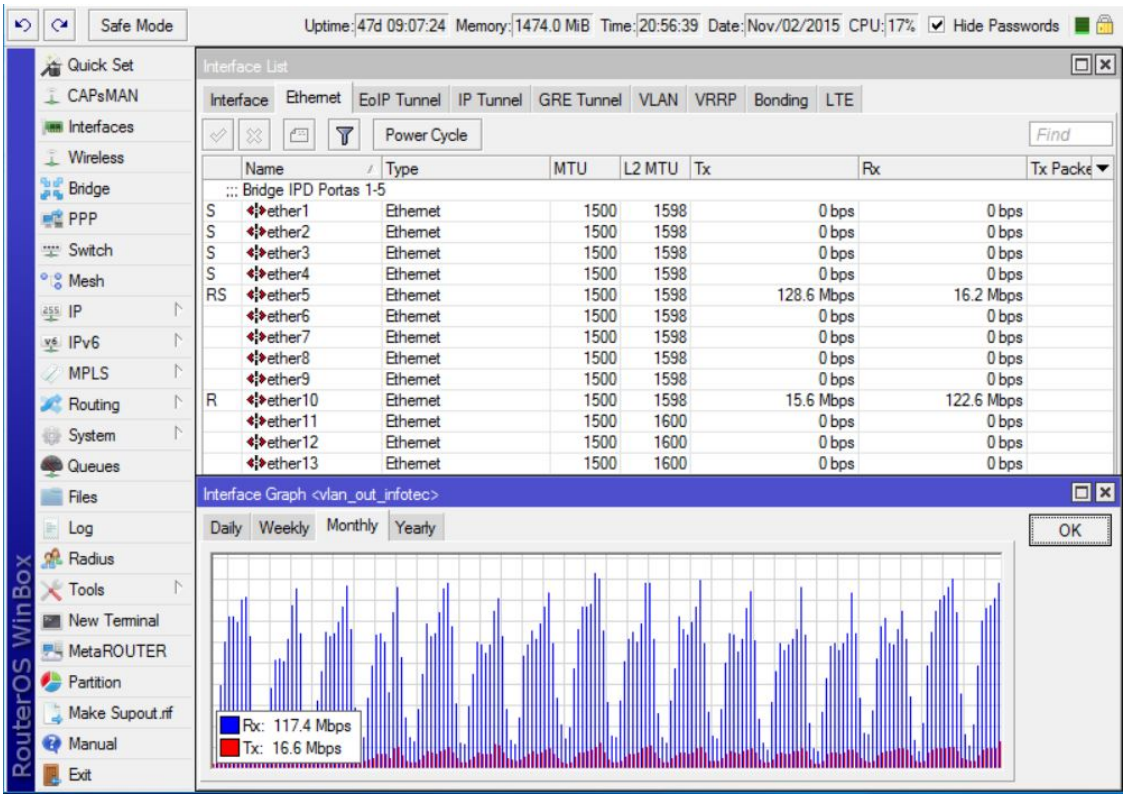

Figura 8: Ferramenta de configuração Winbox da Mikrotik

Nesta tela pode-se observar um gráfico de acompanhamento da interfaces (selecionável), bem como uma apresentação de todas as interfaces do tipo *ethernet* disponíveis para uso no *hardware* no caso uma RouterBoard 1100 AHX2). O Winbox ainda permite monitorar em tempo real informações sobre utilização do processador, memória de acesso randômico disponível, tempo que o equipamento está ligado e data/hora do sistema. À esquerda é apresentado o menu com as opções de configuração e funcionalidades do sistema.

Além da interface gráfica e acesso via console (SSH e TELNET) existe ainda uma *Application Programming Interface* - API de integração, disponível em diversas linguagens de programação, inclusive em *Hypertext Preprocessor* - PHP, PERL, C#, Python, Java, entre outras. Esta API tem o mesmo poder de configuração do acesso via console, entretanto, permite integrar outros sistemas ao Mikrotik RouterOS de forma automatizada e simplificada (MIKROTIK, 2015).

Comparado às outras soluções de *firewall*, como por exemplo o sistema IPTABLES e IPFW, o sistema Mikrotik RouterOS se mostra como uma solução eficaz por se tratar de uma proposta de *hardware* e *software* desenvolvidos buscando aproveitar ao máximo o potencial um do outro. No caso da solução da Mikrotik, tanto o *software* quanto o *hardware* são especialmente

desenvolvidos para um propósito específico e suas arquiteturas são otimizadas para tais tarefas (MIKROTIK, 2015).

Ainda, as soluções desenvolvidas pela Mikrotik podem ser aplicadas em diversos cenários. Pode-se tanto utilizar soluções Mikrotik para redes domésticas quanto, por exemplo, para soluções para gerenciamento e controle de provedores de acesso à internet. Além disso, estas soluções podem desempenhar a grande maioria das funcionalidades de soluções que apresentam um custo mais elevado, como por exemplo, soluções da líder de mercado Cisco e seu maior concorrente, a Juniper.

É importante destacar que sistemas como o IPTABLES são soluções que dependem, em geral, de um sistema operacional de propósito geral para operarem. Além disso, sua instalação é realizada, por padrão, sobre um *hardware* também de propósito geral, diferentemente da solução da Mikrotik.

Ainda traçando um comparativo entre a solução Mikrotik e outras soluções de proteção e controle, é relevante lembrar que este sistema, diferente das demais aplicações apresentadas que tratam exclusivamente da função de *firewall*, é uma solução especialista, completa e aplicada as mais diversas finalidades de gestão de redes. (MIKROTIK, 2015)

4 TRABALHO DESENVOLVIDO

A operação do sistema foi dividida em três etapas. A primeira delas refere-se a detecção dos ataques. Esta tarefa é responsabilidade do IDS SNORT, operando como NIDS, o qual tem como principal funcionalidade identificar, registrar e notificar o administrador do sistema. A segunda refere-se ao sistema intermediário, foco deste trabalho, o qual tem a finalidade de capturar os registros do NIDS SNORT, extrair as informações e utilizá-las na formatação das regras de proteção. E por fim, a terceira trata da aplicação das regras criadas pelo sistema intermediário no sistema de *firewall* do sistema Mikrotik RouterOS.

4.1 Arquitetura

A arquitetura desenvolvida para este sistema está distribuída em três etapas, conforme mencionado anteriormente. Desta forma, o sistema está dividido em (i) Detecção, (ii) Extração e Conversão e (iii) Aplicação, conforme apresenta a Figura 9, onde é ilustrado o fluxo dos processos do sistema. Cada uma destas etapas é detalhada na seção 4.2. Contudo, esta seção refere-se as funcionalidades de cada uma destas etapas.

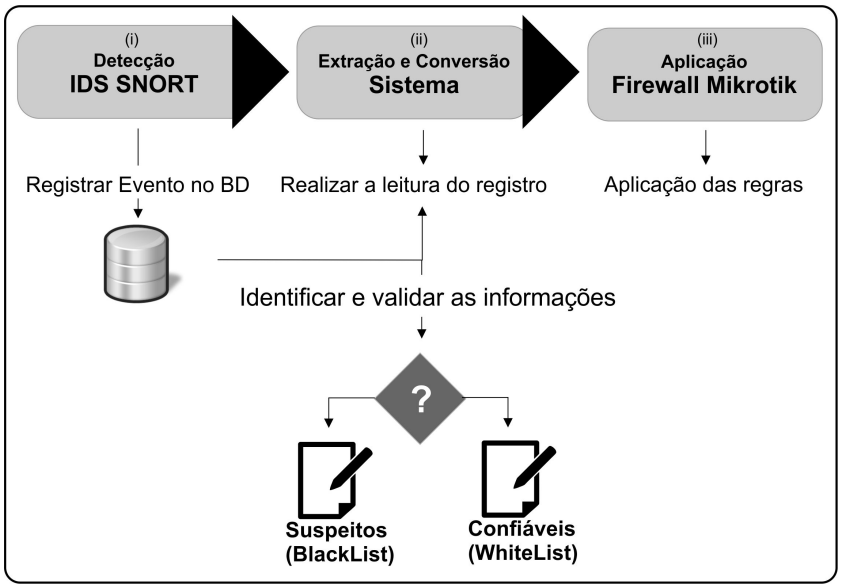

Figura 9: Fluxo dos processos do sistema desenvolvido

4.1.1 Detecção

A fase de detecção é responsável por identificar os ataques. Para isso, utiliza-se o NIDS SNORT para avaliar o tráfego transmitido pela rede com o objetivo de identificar atividades suspeitas, ou ainda, ataques à rede. Sempre que identificado um incidente com tais características, é realizado o registro desta ocorrência, onde são armazenadas informações, como por exemplo, os endereços IP's de origem e de destino. Neste registro, pode-se ainda inserir informações sobre a assinatura do ataque e/ou porta utilizada, bem como tipo de protocolo utilizado.

Para cada ocorrência o NIDS SNORT gera um registro. Com isso, todas as atividades consideradas como suspeitas ficam devidamente armazenadas em um arquivo de texto ou em banco de dados. Para o desenvolvimento deste trabalho, foram utilizados complementos que permitiram que estas informações fossem armazenadas em uma base de dados, melhorando consideravelmente o acesso e processamento destas informações. Esta base de dados é utilizada pela próxima etapa como "fonte de conhecimento", isto é, de onde é extraído todo o conteúdo necessário para operação das próximas fases do processo.

Para que fosse possível identificar ataques ou acessos não permitidos, primeiramente foi necessário estabelecer uma "política de utilização da rede". Para isso, foram elaboradas regras de detecção onde o NIDS SNORT pudesse identificar situações que desrespeitassem estas políticas, as quais estabelecem as seguintes restrições:

- Usuários da rede interna não podem acessar websites via porta TCP 80, tanto no ambiente interno quanto externo;
- Utilizar o serviço de *Domain Name System* - DNS para resolução de nomes, via porta UDP 53;
- Não utilizar serviços de *File Transfer Protocol* - FTP;
- Não permitir acesso externo a serviços de website, também via porta TCP 80;
- Não permitir conexões externas repetitivas e conexões internas ao servidor de base de dados MySQL, via porta TCP 3306;

Além de identificar *hosts* que desrespeitem essas políticas, foi elaborada uma regra que permitisse ao NIDS SNORT identificar ataques externos de PING *Flood*, isto é, onde um atacante envia inúmeras requisições de PING sem aguardar resposta, com o intuito de sobrecarregar o sistema e atingir a indisponibilidade de recursos e serviços do *host* atacado.

Para cada situação descrita nestas políticas foi criada uma regra de detecção junto ao NIDS SNORT. Cada regra, quando infringida dispara automaticamente um alerta. Ao disparar este alerta, o NIDS SNORT registra a ocorrência e os endereços IP dos *hosts* envolvidos, fornecendo informações para a próxima fase, descrita na Seção 4.1.2.

4.1.2 Extração e Conversão

Após o NIDS SNORT identificar e registrar os ataques contra a rede, é necessário utilizar estas informações para proteger a rede. Na segunda etapa da operação do sistema, é realizada a leitura dos registros realizados pelo NIDS SNORT, a extração das informações e formatação das regras de proteção que serão aplicadas ao módulo de *firewall* do sistema Mikrotik RouterOS. A Figura 10 apresenta a estrutura de registro de detecção do IDS SNORT, onde é possível verificar as informações possíveis de serem extraídas, relevantes à criação de regras de controle para o *firewall* do Mikrotik RouterOS.

sid	cid	signature	sig_name	sig_class_id	sig_priority	timestamp	ip_src	ip_dst	ip_proto	layer4_sport	layer4_dport
1	9.385	517	Snort Alert [1:1000991:1]	27	2	2015-10-27 20:12:48	167.838.205	921.121.138	6	52.678	80
1	9.386	517	Snort Alert [1:1000991:1]	27	2	2015-10-27 20:12:53	167.838.205	921.121.138	6	52.678	80
1	9.387	517	Snort Alert [1:1000991:1]	27	2	2015-10-27 20:13:00	167.838.205	3.627.736.578	6	52.634	80

Figura 10: Informações registradas pelo NIDS SNORT

Neste arquivo é possível identificar o endereço que originou o ataque e qual o *host* de destino do ataque. Da mesma forma, pode-se identificar a regra que gerou o alerta e quando o ataque ocorreu. Ainda é possível identificar o protocolo e as portas utilizadas no ataque.

Com estas informações já é possível a criação de regras que impeçam o acesso deste *host* de origem à rede, prevenindo novos ataques descritos nas regras utilizadas. Nesta etapa o sistema intermediário captura estas informações do registro das ocorrências e realiza a identificação dos atributos necessários para elaborar as regras de proteção, como por exemplo, endereço IP de origem utilizado para realizar o ataque.

As informações extraídas do registro de ocorrências do NIDS SNORT são aplicadas na

criação das regras de proteção do sistema de *firewall* do Mikrotik RouterOS e servem para popular uma lista de *hosts* classificados como “suspeitos” junto ao sistema Mikrotik RouterOS. Os endereços de IP identificados como origem do ataque são inseridos nesta lista e terão o seu acesso à rede negado pelo *firewall* permanentemente até que o administrador remova-os desta lista de controle. Com esta abordagem, busca-se minimizar problemas de detecção do tipo “falso negativo”, isto é, permitir acesso à rede aos *hosts* identificados como "suspeitos".

Contudo, é necessário realizar uma validação destes endereços de origem. Os endereços IP identificados como origem não serão registrados na lista de “suspeitos” antes de uma validação, isto é, caso estejam registrados em outra lista: a lista de *hosts* “confiáveis”. Esta lista é mantida pelo administrador da rede e identifica os *hosts* que não devem ter o acesso à rede bloqueado, mesmo que identificados como origem de um possível ataque pelo NIDS SNORT.

Esta abordagem busca evitar o bloqueio de ocorrências de “falsos positivos”, isto é, quando acessos legítimos à rede são identificados como ataques. Ambas as listas podem ainda ser manipuladas pelo administrador do sistema de forma manual através de um painel de gerência do sistema. É prudente mencionar que para evitar falhas do sistema, o administrador tome as medidas de segurança cabíveis para evitar que, de fato, um dispositivo inserido como "confiável" não seja vítima de um ataque e realize ataques contra sua estrutura de rede.

Após identificar e gerar as regras de proteção, é necessário aplicá-las ao sistema de proteção da rede, isto é, ao sistema de *firewall*. A seção seguinte apresenta como o sistema realiza a aplicação das regras geradas no módulo de *firewall* do Mikrotik RouterOS, efetivando a proteção contra os atacantes.

4.1.3 Aplicação

Após a extração e elaboração das regras de controle de acesso aos *hosts* identificados e validados como "suspeitos" é necessário executar a inserção destas regras no sistema de *firewall* para que se tornem ativas. Desta forma, o sistema estabelece uma conexão através da API disponibilizada pela Mikrotik, utilizando a linguagem PHP, e realiza a execução do conjunto de comandos necessários para registro destas informações, efetivando sua aplicação.

Primeiramente, após a conexão com o sistema Mikrotik, é realizada a criação de bloqueio total de acesso para todos os *hosts* que serão inseridos na lista de "suspeitos". Após, é realizada a inserção dos endereços identificados como origem do ataque na lista de *hosts* "suspeitos". Assim, o acesso à rede é totalmente negado a estes *hosts*, impedindo novos ataques através da regra de *firewall*. É válido destacar que, ao passo que se restringe o acesso de *hosts* potencialmente perigosos à rede, reduz-se também o volume de dados maliciosos, permitindo ao NIDS SNORT dedicar recursos computacionais à identificação de outros ataques com origens diferentes dos *hosts* já identificados. A Figura 11 mostra uma visão geral do projeto.

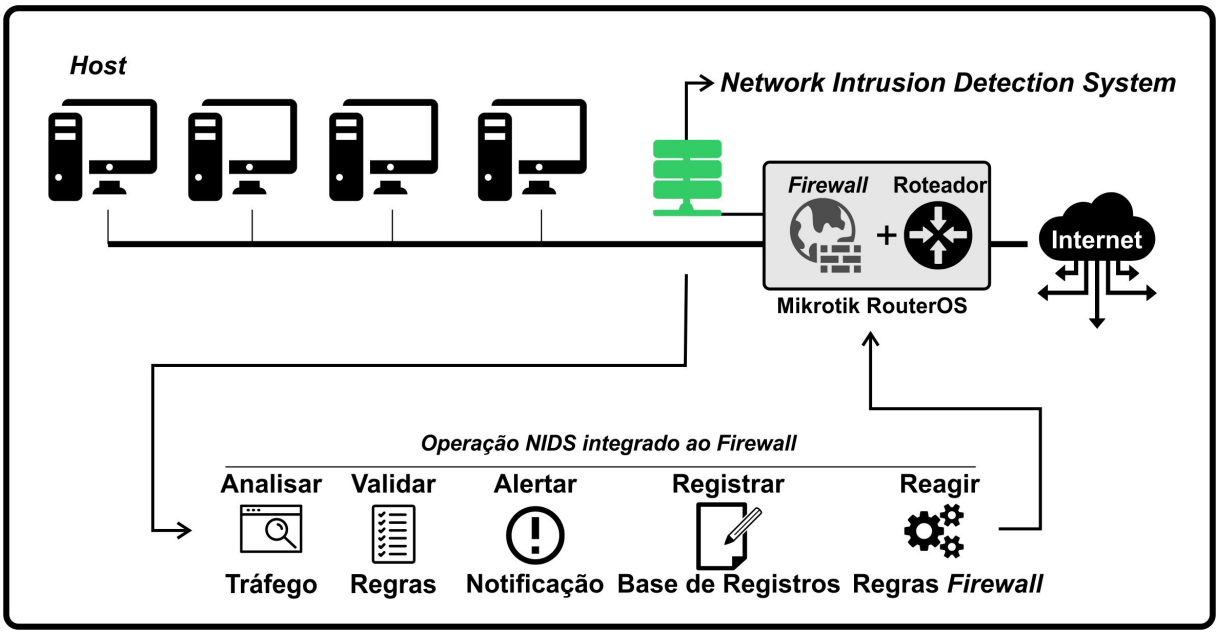

Figura 11: Visão geral do projeto

É importante salientar que este trabalho não tem foco na identificação dos ataques, mas sim na proteção da rede, utilizando-se dos registros de detecção. Assim, toda e qualquer melhoria aplicada à fase de detecção do NIDS SNORT tende a ser refletida em melhorias no sistema de proteção. Desta forma, o sistema desenvolvido trata-se de um automatizador de tarefas de responsabilidade do administrador. Assim, o sistema possibilita "transformar" registros de detecção em "regras de proteção", reduzindo a intervenção do administrador do sistema na realização destas tarefas através da automatização destas atividades.

Esta seção apresentou as funcionalidades e organização de cada etapa do sistema intermediário desenvolvido e responsável pela integração das duas ferramentas de proteção, isto é, o NIDS SNORT e o *firewall* do sistema Mikrotik RouterOS. A seguir, a seção 4.2 apresenta em detalhes a estruturação, fluxo de operações e ações executadas pelo sistema em cada uma das

etapas.

4.2 Modelagem

Esta seção detalha a operação das três etapas em que o sistema foi dividido. A etapa de detecção é apresentada na subseção 4.2.1. Já a etapa de captura das informações e extração dos dados é abordada na subseção 4.2.2. Por fim, na subseção 4.2.3 refere-se a aplicação das regras de proteção, isto é, a etapa em que as informações extraídas dos registros de detecção do NIDS SNORT são transformadas em regras de *firewall* do sistema Mikrotik RouterOS.

4.2.1 Etapa de Detecção

A etapa de detecção é realizada exclusivamente pelo NIDS SNORT, o qual é responsável por analisar o conteúdo dos pacotes de dados que são transmitidos pela rede. Caso identificada alguma atividade maliciosa, é realizado o registro deste evento, gravando as seguintes informações em uma base de dados:

- Data e hora da ocorrência
- Endereço de origem do ataque
- Endereço de destino do ataque
- Regra que disparou o alerta
- ID do registro
- Assinatura do ataque
- Protocolo utilizado
- Porta utilizada

Essa estrutura de dados é utilizada para registrar todos os ataques identificados pelo NIDS. Essa operação de registro é contínua e executada enquanto o NIDS SNORT estiver em operação.

As informações contidas nesta base de dados serão utilizadas pela próxima etapa do sistema, apresentada na subseção 4.2.2, onde é detalhada a etapa de captura destes dados e transformação destas informações em regras de proteção que serão aplicadas ao *firewall* do sistema Mikrotik RouterOS, conforme descrito na seção 4.2.3. A Figura 12 apresenta o ambiente de teste da aplicação.

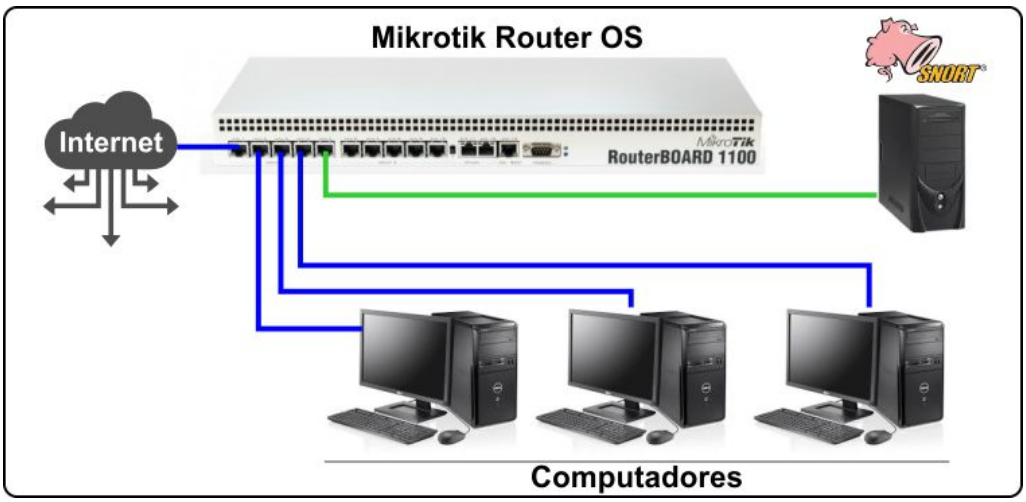

Figura 12: Ambiente de teste

A Figura 12 apresenta o ambiente utilizado para realização dos testes e validação da aplicação desenvolvida. Para que fosse possível realizar a análise dos dados, o dispositivo com o sistema NIDS SNORT foi conectado a uma RouterBOARD modelo 1100, a qual é responsável por interconectar todos os demais dispositivos da rede. Contudo, foi necessário realizar uma configuração especial nesta RouterBOARD, denominada *Interface Mirror*. Desta forma, foi possível "espelhar" todo o tráfego transmitido pela rede, isto é, encaminhar uma cópia de todas as informações transmitidas pela rede, para interface de conexão do NIDS.

Após, foi necessário habilitar o NIDS SNORT para operar em modo promíscuo, isto é, para que ele capturasse todo o conteúdo que fosse direcionado para sua interface de rede. Assim, combinando a configuração de *Interface Mirror* e o modo promíscuo de operação, foi possível analisar o conteúdo transmitido pela rede a todos os dispositivos.

Para que fosse possível identificar atividades que desrespeitassem as políticas estabelecidas pelo administrador da rede, foram elaboradas um conjunto de regras para cada situação, e adicionalmente, uma regra para identificar um ataque de PING *Flood*.

Com base neste conjunto de regras de detecção, sempre que o NIDS SNORT identificar

algum tipo de atividade com este perfil, automaticamente são identificados os *hosts* envolvidos e as informações de conexão entre eles. Feito isso, o NIDS SNORT realiza um registro na base de dados de ocorrências, inserindo informações referentes ao ataque. A Figura 13 apresenta este conjunto de regras e as respectivas funcionalidades.

Conjunto de regras de detecção

alert icmp any any -> $HOME_NET any (msg:"Ataque de PING FLOOD contra rede Interna"; threshold: type both, track by_dst, count 500, seconds 30; sid:1000990; classtype:icmp-event; rev:001;)

Função: Identificar *host*s que, num período de 30 segundos, enviem mais de 500 requisões de PING, de qualquer rede contra a rede interna, utilizando qualquer porta e através do protocolo *Internet Control Message Protocol* - ICMP.

alert tcp $HOME_NET any -> any 80 (msg:"Acesso Host Interno à Website externo"; sid:1000991; classtype:web-application-activity; rev:1;)

Função: Identificar *host* da rede interna que acessem serviços de websites utilizando a porta 80 do protocolo TCP

alert udp $HOME_NET any -> any 53 (msg:"Requisição Host Interno à DNS"; sid:1000992; classtype:not-suspicious; rev:001;)

Função: Identificar *host* da rede interna que solicitem resolução de nomes à DNS utilizando a porta 53 do protocolo UDP

alert tcp $HOME_NET any -> any 21 (msg:"Acesso Host Interno à SERVIDOR FTP"; sid:1000993; classtype:web-application-activity; rev:1;)

Função: detectar *host* da rede interna que utilizem o serviço de FTP utilizando a porta 21 do protocolo TCP

alert tcp any any -> $HOME_NET 80 (msg:"Acesso Host externo à WWW Rede Interna"; sid:1000994; classtype:web-application-activity; rev:1;)

Função: identificar *host* de qualquer rede que tentem acessar serviços de websites instalados na rede interna que utilizem a porta 80 do protocolo TCP

alert tcp any any -> $HOME_NET 3306 (msg:"Conexao Qualquer Host à Servidor MySql Interno "; threshold: type both, track by_dst, count 5, seconds 5; sid:1000995; classtype:not-suspicious; rev:001;)

Função: Identificar *qualquer host* que realize mais de 5 tentativas de conexão à base de Dados MySql, através da porta 3306 do protocolo TCP, em qualquer dispositivo da rede interna, no intervalo de 5 segundos

alert tcp $HOME_NET any -> $HOME_NET 3306 (msg:"Conexao Host Interno à MySql Interna"; sid:1000996; classtype:not-suspicious; rev:001;)

Função: Identificar *host* da rede interna que realizem conexões à base de dados MySql e, dispositivos da rede interna, utilizando a porta 3306 do protocolo TCP.

Figura 13: Regras elaboradas para detecção do NIDS SNORT

Após essa etapa de detecção, é necessário transformar essas informações em regras de proteção, com o objetivo de impedir novos ataques à rede. Essa etapa do processo é apresentada na subseção 4.2.2.

4.2.2 Etapa de Captura de informações e conversão em Regras de proteção

Esta seção apresenta a fase de captura dos dados dos registros do NIDS SNORT e extração das informações necessárias para elaborar regras de proteção, impedindo, através do sistema de *firewall* do Mikrotik RouterOS, o acesso de *hosts* identificados como origem de ataques à rede de computadores.

Esta etapa contempla o núcleo do sistema desenvolvido, pois é neste momento que ocorre a transformação das informações contidas nos registros em regras de proteção. Para isso, o sistema intermediário desenvolvido opera em momentos distintos: (i) leitura das informações de registro do NIDS SNORT, (ii) identificação dos endereços de IP que originaram o ataque, (iii) validação dos endereços de IP, (iv) a formatação das regras de bloqueio utilizando as informações extraídas e (v) a aplicação destas regras ao sistema de *firewall*. A Figura 14 apresenta a sequência desta operação.

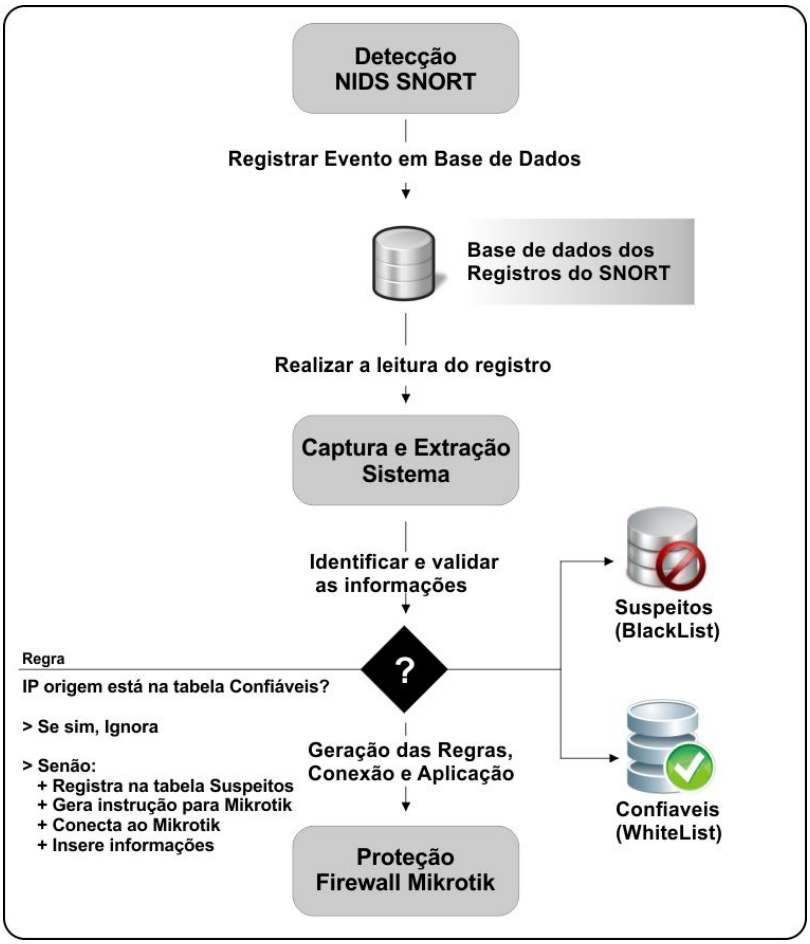

Figura 14: Sequência de tarefas do sistema

A Figura 14 apresenta o fluxo das informações dentro do sistema e cada etapa de operação.

Ilustra ainda, o momento em que o sistema valida as informações, buscado reduzir ocorrências dos falsos positivos e negativos, respeitando as listas de *hosts* suspeitos e confiáveis.

Para facilitar a utilização do sistema, foi desenvolvido uma interface de controle, na qual o administrador pode facilmente adicionar e remover os endereços de IP, tanto da lista de confiáveis quando da lista de suspeitos. Esse painel de controle do sistema é apresentado pela Figura 15.

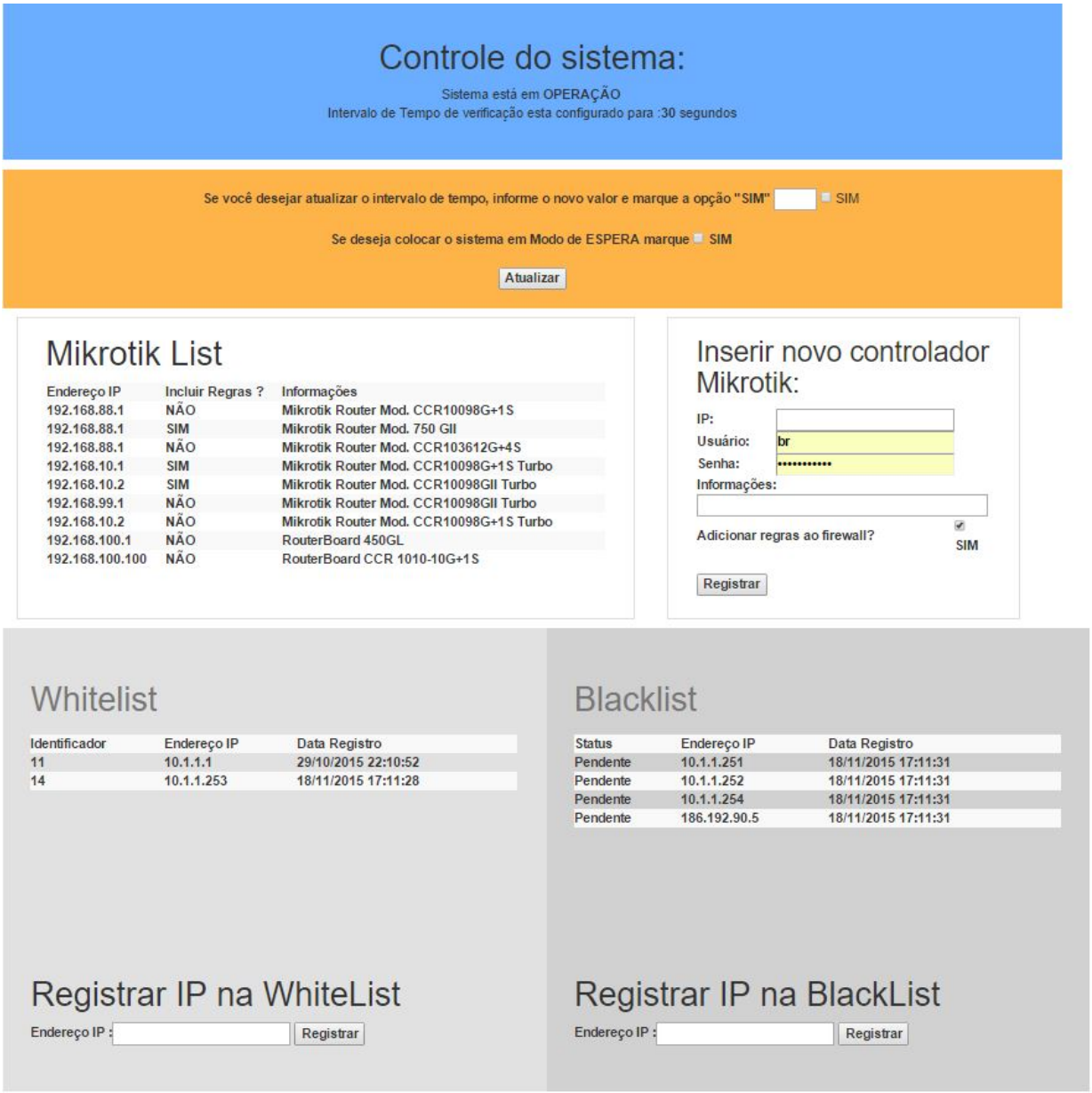

Figura 15: Painel de controle do sistema

Além das funcionalidade de controle das listas, é possível adicionar uma lista de dispositivos que deverão receber as regras, caso existam mais sistemas de *firewall* sob a responsabilidade do administrador. Isso permite prevenir ataques de *hosts* já conhecidos.

O painel de controle permite também que o administrador determine o intervalo de tempo para realizar a verificação das informações e aplicação das regras. Permite ainda que o administrador coloque o sistema em "modo de espera", isto é, em um modo que o sistema apenas mantenha sua execução sem realizar nenhum tipo de verificação ou controle.

Entretanto, para realizar as tarefas disponíveis através do painel de controle e também as tarefas de capturar e elaborar as regras de controle, o sistema utiliza de um *script* desenvolvido utilizando a linguagem PHP e utilizando recursos da API, também desenvolvida em PHP, disponibilizada pela Mikrotik, o qual é executado em segundo plano. Esse modo de execução é possível, pois, após inicializado o sistema, sua execução é cíclica, sem a necessidade da intervenção direta do administrador. Contudo, a sua operação pode ser suspensa pelo administrador via painel de controle.

Para realizar a captura das informações de registro, o sistema estabelece uma conexão com a base de dados auxiliar "midle_db" onde estão armazenadas estas informações e, realiza uma série de consultas e inserções ou remoções, conforme apresentado na Figura 16.

```
/* Tabela de query*//*ADICIONADO MIDLE_DB.*/
$query_insert_blacklist = "INSERT INTO MIDLE_DB.IP_BLACKLIST (IP_ADD) SELECT DISTINCT IP_SRC
FROM SNORT.ACID_EVENT WHERE IP_SRC NOT IN (SELECT IP_ADD FROM MIDLE_DB.IP_WHITELIST) AND
IP_SRC NOT IN (SELECT IP_ADD FROM MIDLE_DB.IP_BLACKLIST)";
/*cARREGA TODOS OS IPS QUE ESTÃO NA TABELA E NÃO FORAM VERIFICADOS*/
$query_load_blacklist = "SELECT * FROM MIDLE_DB.IP_BLACKLIST WHERE STATUS ='0'";
$query_system = "SELECT * FROM MIDLE_DB.CONFIGURATION";
$query_check_rules = "SELECT * FROM MIDLE_DB.CHECK_RULES";
```

Figura 16: Consultas realizadas para capturar as informações de registro do IDS SNORT

Na consulta denominada "query_insert_blacklist", o sistema realiza uma inserção de valores na tabela referente aos *hosts* suspeitos, denominada "ip_blacklist", pertencente a base de dados auxiliar "midle_db". A estrutura desta base de dados auxiliar é apresentada pela Figura 17.

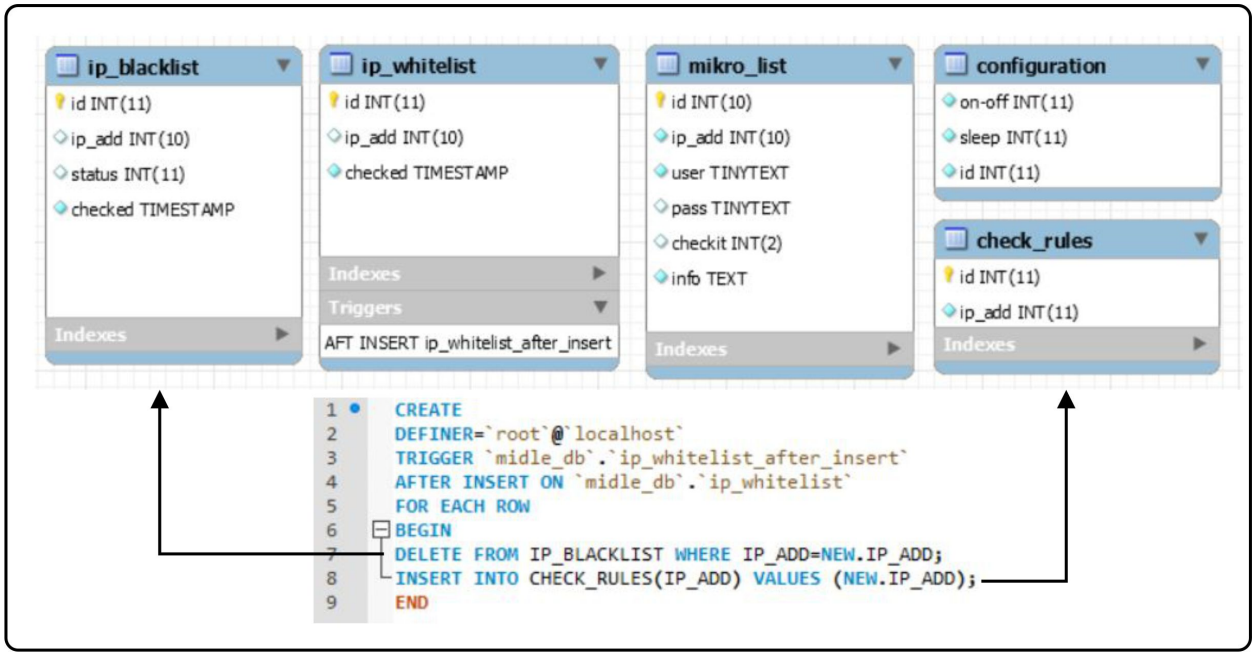

Figura 17: Base de dados auxiliar utilizada pelo sistema

Estes valores referem-se a todos os endereços de IP de origem, da tabela de registro de ocorrências mantida pelo IDS SNORT, que, não estão presentes na lista de confiáveis (ip_whitelist) e que ainda não foram inseridos na tabela de suspeitos (ip_blacklist). Com essa abordagem, evita-se duplicidade de valores na tabela de suspeitos, bem como, a criação de regras de bloqueio para endereços definidos como confiáveis, mesmo que o IDS detecte como possível atacante.

Em seguida, a consulta "query_load_blacklist" é responsável por retornar informações de todos os endereços contidos na tabela "ip_blacklist" com "status=0", isto é, que ainda não foram processados. Já a consulta "query_system" é responsável por retornar as informações de operação do sistema. E, por fim, a consulta "query_check_rules" é responsável por retornar os endereços de IP que foram removidos da tabela "ip_blacklist". Esses motivos podem ser por que (i) o administrador inseriu o endereço de IP como confiável ou (ii) o administrador removeu manualmente o endereço de IP da lista de suspeitos.

Para ambas as situações, buscando garantir a integridade das regras, é disparado automaticamente um processo (*trigger*). No caso (i), todo endereço adicionado à lista de confiáveis é automaticamente removido da lista de suspeitos. Em consequência disso, no caso (ii), todo endereço de IP removido da tabela de suspeito, deve ter suas regras removidas também do sistema de proteção do *firewall* do Mikrotik RouterOS. Por isso, os endereços removidos da tabela "ip_blacklist" são automaticamente inseridos na tabela "check_rules", responsável por realizar

esta manutenção.

Com as informações necessárias para elaborar as regras de proteção, o sistema passa a operar na formatação das regras de controle que serão aplicadas ao sistema de *firewall*. A Figura 18, apresenta as regras utilizadas para inserir as informações já processadas, utilizando a API de controle da Mikrotik.

```
$API->comm("/ip/firewall/filter/add", array("chain"=>"input", "src-address-list"=>
"SNORT_DETECTION", "action" => "drop", "comment" => "Bloqueio Total Gerado Baseado em Deteção ",));

$API->comm("/ip/firewall/address-list/add", array("list"=>"SNORT_DETECTION", "address"=>"".$src_host.""",)))

$API->comm("/ip/firewall/address-list/print", array(".proplist"=> ".id","?address" => "".$row['ip_add']."",));

$API->comm('/ip/firewall/address-list/remove', array(".id"=>"".$line['.id']."")))
```

Figura 18: Regras formatadas para controle do *firewall* do Mikrotik RouterOS

A Figura 18 apresenta o conjunto de ações utilizadas para controlar o *firewall* do sistema Mikrotik RouterOS. As ações apresentadas são responsáveis por inserir uma regra global de bloqueio para todos os *hosts* identificados, adicionar os endereços IPs a uma lista de bloqueio, pesquisar endereços na lista de IP bloqueados e remover estes endereços desta lista. Entretanto, após capturar as informações é necessário aplicá-las junto ao sistema de *firewall* do Mikrotik RouterOS.

4.2.3 Etapa de Aplicação das Regras

Após a captura de todas as informações necessárias para elaboração das regras de controle, é fundamental inserir estas regras no sistema de *firewall* do Mikrotik RouterOS para que elas sejam executadas. Para isso, o sistema executa as instruções da API Mikrotik que realizam a conexão com o sistema Mikrotik RouterOS e possibilita a execução das instruções necessárias para criação das regras de controle do *firewall*, conforme mostra a Figura 19.

```
/*Requisição da API da Mikrotik */
require("mk_api/routeros_api.class.php");
/*Criação de uma instancia da Classe API*/
$API = new RouterosAPI();
/*Tentativa de conexão com o dispositivo gerenciador da rede */
if ($API->connect($ip_add, $username, $pass)) {
```

Figura 19: Conexão para aplicação das regras de controle

Após estabelecer conexão, a aplicação executa a instrução para criar uma regra global, que

utilizará a lista de endereços, denominada SNORT DETECTION, para efetuar o bloqueio de todo conteúdo originário dos endereços contidos nesta lista.

Durante sua operação, o sistema também realiza a inserção de todos os endereços contidos na tabela "ip_blacklist" na lista de endereços que serão bloqueados pela regra geral aplicada ao *firewall*. Além disso, quando necessária a remoção de algum endereço da lista de bloqueados, seja por remoção manual deste endereço IP por parte do administrador ou por inserção deste endereço IP na lista de confiáveis, o sistema realiza uma busca na lista de bloqueados e realiza a remoção desta regra. Com esta abordagem, o objetivo é manter a integridade das políticas de acesso estabelecidas pelo administrador. As instruções de controle do sistema de *firewall* são apresentadas pela Figura 20.

Regras de controle do firewall do Mikrotik RouterOS

Bloqueio Total

```
$API->comm("/ip/firewall/filter/add", array("chain"=>"input", "src-address-list"=>"SNORT_DETECTION", "action" => "drop",
        "comment" => "Bloqueio Total Gerado Baseado em Deteção ",));
```

Inserção na lista de bloqueio (SNORT_DETECTION)

```
$API->comm("/ip/firewall/address-list/add", array("list"=>"SNORT_DETECTION", "address"=>"".$src_host."",))){
```

Pesquisa endereço IP na lista de endereços de bloqueio

```
$SEARCH = $API->comm("/ip/firewall/address-list/print", array(
".proplist"=> ".id",
"?address" => "".$row['ip_add']."",
));
```

Remove endereço IP da lista de bloqueio

```
$API->comm('/ip/firewall/address-list/remove', array(".id"=>"".$line['.id'].""))
```

Figura 20: Regras de controle do sistema de *firewall* para bloqueio, inserção, pesquisa e remoção

Ao adotar o bloqueio total, o objetivo é impedir que um *host* de origem de um ataque realize novos ataques ou explore alguma outra vulnerabilidade do sistema. Da mesma forma, ao eliminar todo e qualquer acesso de *host* definido como suspeito, evita-se que o sistema permita a ocorrência de falsos negativos.

Esta capítulo apresentou a modelagem aplicada ao projeto desenvolvido e as funcionalidades de cada etapa do processo. Detalhou ainda, todas as operações fundamentais do sistema e como estas estão interligadas entre si. Os teste realizados e a metodologia aplicada para validar este trabalho é apresentada no Capítulo 5.

5 RESULTADOS

Para validar este trabalho, foi elaborado um ambiente para testar e validar este trabalho, bem como um conjunto de regras que permitissem ao NIDS SNORT identificar atividades que desrespeitassem a política de utilização definida para a rede e/ou ataques à rede. Estas regras foram criadas para identificar *hosts* em um ambiente de rede laboratorial, conforme apresentado no capítulo anterior.

5.1 Metodologia

Esta Seção é apresentado o ambiente elaborado para realização dos teste, os dispositivos utilizados e tecnologias aplicadas no projeto. Com o objetivo de validar este trabalho foi desenvolvido o seguinte cenário:

Para desenvolver o ambiente de teste deste projeto foi utilizado uma RouterBOARD modelo 1100, com sistema Mikrotik embarcado operando na versão 6.28, como dispositivo de rede para interconectar todos os demais dispositivos. Este dispositivo conta com um processador de rede PowerPC de 800MHz e 512Mb de memória RAM. Além disso, esta RouterBOARD é composta por 13 portas *ethernet* com velocidade de 1 gigabit cada, totalmente configurável, sendo que as 5 primeiras portas *ethernet* fazem parte de um *switch* gerenciado. Tal característica permitiu realizar a configuração de *Port Mirror*, permitindo com que o conteúdo transmitido pelas demais portas de rede fosse também encaminhado para interface de conexão do NIDS SNORT, permitindo com que esse conteúdo fosse analisado pelo sistema.

Como dispositivo denominado "Servidor IDS", responsável por dar suporte ao sistema NIDS SNORT e demais complementos, bem como a base de dados Mysql utilizada para registrar as ocorrências, foi utilizado um computador tipo PC, com processador Intel Core 2 Duo de 2.0 GHz e 3Gb de memória RAM, utilizando um disco com capacidade de até 1 Tb, operando a 7.200 rotações por minuto. Este equipamento utiliza uma interface de rede também com velocidade de 1 gigabit para comunicação. Como sistema operacional, foi utilizado o Ubuntu Server,

devidamente atualizado, em sua versão 14, por questões de compatibilidade e estabilidade do sistema para operar com o NIDS SNORT.

Para realizar a troca de informação, simulando estações de trabalho, foram utilizados 3 computadores pessoais, de uso convencional, utilizando sistema operacional Windows 8. Estes dispositivos foram conectados à RouterBOARD e o conteúdo transmitidos pelas suas respectivas interfaces de conexão, foram "espelhados" para a porta de conexão do NIDS SNORT.

Para identificar e registrar as ocorrências, foi utilizado a versão 2.9.7 do IDS SNORT, configurado para operar em modo NIDS. Para validação do trabalho, não foram aplicadas as regras disponibilizadas pela comunidade do projeto, pois, conforme estudos realizados, para obter o melhor desempenho do NIDS, as regras de detecção devem ser ajustadas conforme o ambiente de aplicação. Como o objetivo deste trabalho não está na detecção de atividade suspeita, mas sim na integração das duas soluções, foram utilizadas regras específicas para o ambiente de teste, levando em consideração as políticas estabelecidas. É importante salientar que toda e qualquer melhoria na etapa de detecção resultará em uma maior proteção através da integração proposta.

Para desenvolver esse trabalho foi utilizada a linguagem de programação PHP, pela compatibilidade, documentação e suporte à API da Mikrotik. Tal linguagem foi definida também pela sua flexibilidade de desenvolver aplicações web, permitindo com que o sistema seja utilizado em diferentes plataformas.

5.2 Testes Realizados

Para simular os acessos na rede de teste foram utilizados computadores que receberam seus endereços de IP dinamicamente através de um servidor DHCP configurado na RouterBOARD. Após, foram realizados diferentes tipos de acessos aos serviços que infringissem à politica de utilização da rede, conforme representado pela Figura 21.

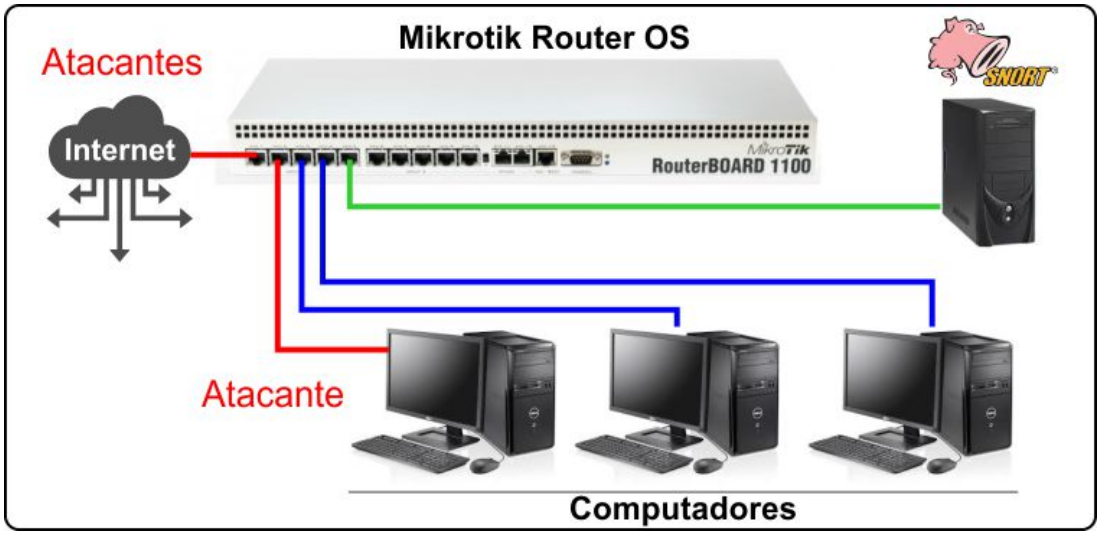

Figura 21: Ambiente desenvolvido para os testes de validação

O Ambiente ilustrado pela Figura 21 apresenta o ambiente de teste desenvolvido, onde são identificados os *hosts* capazes de realizar algum tipo de atividade que desrespeite as políticas impostas à rede. Tais atividades podem ocorrer contra qualquer outro elemento da rede. Da mesma forma que os ataques podem ocorrer tanto dentro do mesmo ambiente de rede quanto podem ser originados de fora do ambiente de rede.

Para validar a regra de detecção de acessos aos serviços que utilizam a porta 80, foram realizados acessos aos websites que utilizam o protocolo *Hypertext Transfer Protocol*, que por padrão utiliza a porta TCP para sua comunicação. Ainda foram realizados acessos ao servidor de páginas configurado junto ao NIDS SNORT, o que também não é permitido de acordo com politica estabelecida. Ao acessar estas páginas, automaticamente, através da resolução de nomes, também foi validada a detecção para o serviço de DNS, o qual utiliza por padrão a porta UDP 53.

Para verificar a regra que trata da porta TCP 3306, padrão para conexões com base de dados Mysql, foram realizados acessos de uma rede externa à base de dados instalada junto ao servidor NIDS SNORT. Da mesma forma, foram realizados acessos, internamente à rede, a mesma base, o que também contraria as regras de detecção.

Ainda sobre os testes de detecção, foi simulada uma tentativa de PING *Flood*, utilizando a ferramenta disponível pelo sistema Mikrotik RouterOS. Neste teste foram enviadas 1000 requisições de PING, com intervalo de 1 milissegundo entre elas, contra o NIDS SNORT (IP

10.1.1.254).

Já no que diz respeito a funcionalidades do sistema, o intervalo de tempo de verificações foi alterado do valor padrão de 10 segundos para 30 segundos e observado a operação do sistema. Da mesma forma, que a funcionalidade de suspensão do serviço, que foi ativada e desativada, buscando validar este recurso.

Após, foram realizados testes para validar os recursos de controle de listas de endereços IP, onde foram adicionados endereços à lista de confiáveis e verificado o tratamento do sistema à estes endereços. Da mesma forma, foram realizadas a remoção destes endereços da lista de confiáveis e observado se o sistema tornava a considerar estes endereços para criação das regras de proteção.

Os resultados obtidos através dos testes realizados são abordados e ilustrados na seção 5.3, onde são detalhadas estas informações.

5.3 Resultados Obtidos

Os resultados obtidos através da análise da utilização da rede foram os esperados. O sistema NIDS SNORT conseguiu, através das regras elaboradas, identificar os testes realizados e registrar as informações referentes à estas ocorrências com sucesso.

No caso do acesso à websites utilizando a porta TCP 80, o NIDS SNORT identificou e registrou as ocorrências, as quais foram utilizadas pelo sistema para a geração das regras de proteção. Da mesma forma que o sistema desenvolvido foi capaz de ler as informações do registro do NIDS SNORT e realizar a inserção destas informações à tabela auxiliar, responsável por armazenar os endereços de IP que terão seu aceso bloqueado (midle_db.ip_blaclist), conforme ilustra a Figura 22.

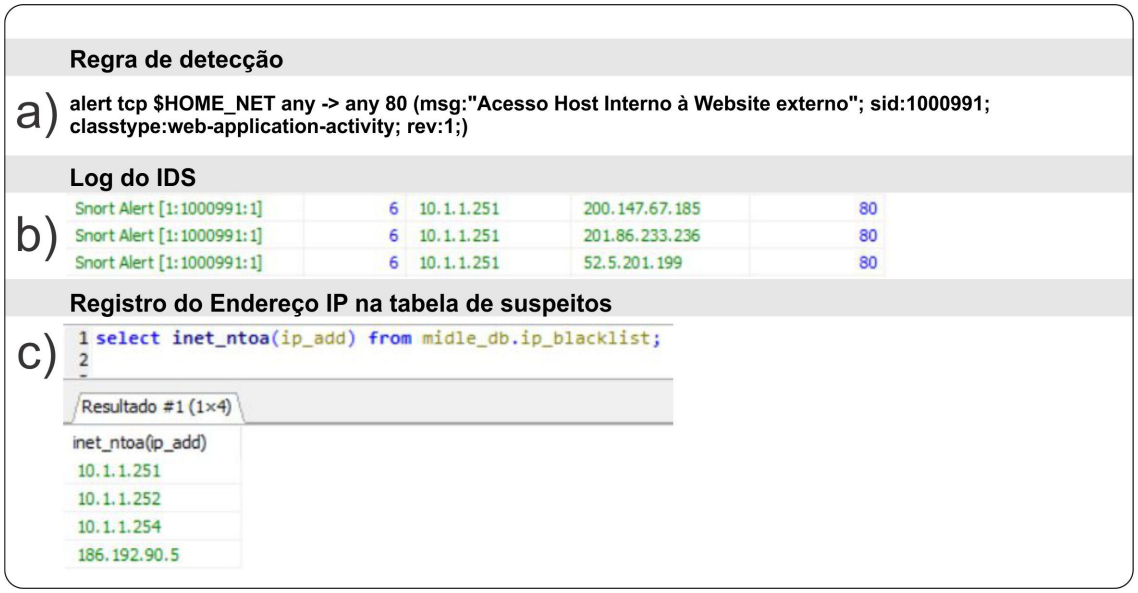

Figura 22: Registros do NIDS, verificação e inserção de endereços na tabela de bloqueio

Na Figura 22.a é identificada a regra de detecção (1000991) responsável pelo registro apresentado pela figura 22.b, que exibe os valores referentes à regra de detecção que gerou o registro, o protocolo utilizado, endereço de origem e destino e a porta de comunicação, respectivamente. Por fim, a figura 22.c mostra o endereço de IP de origem desta detecção (10.1.1.251) inserido na tabela midle_db.ip_blacklist, a qual é responsável por armazenar as informações dos *hosts* que devem ser bloqueados.

No que diz respeito à simulação de ataque de PING *Flood*, utilizando a ferramenta disponível pelo sistema Mikrotik, o sistema NIDS SNORT foi capaz de identificar e registrar este ataque. A Figura 23.a ilustra a utilização da ferramenta e apresenta, na Figura 23.b o registro gerado pelo NIDS SNORT fazendo referência à regra de detecção desenvolvida.

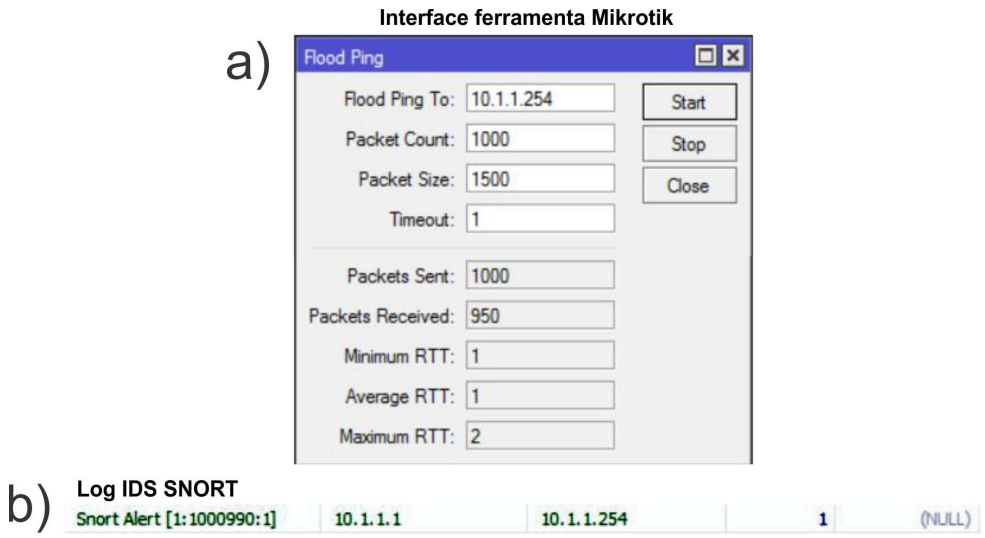

Figura 23: Simulação de PING Flood utilizando ferramenta do Mikrotik RouterOS

Apesar de ter simulado o ataque de PING *Flood* através da RouterBOARD Mikrotik, a qual estava utilizando o endereço IP 10.1.1.1, este foi apenas detectado, sem a inserção do mesmo na lista de bloqueio. Isto por que este endereço IP está presente na lista de confiáveis, conforme mostra a Figura 24. Esse resultado apresentado, valida com sucesso à operação do sistema respeitando a lista de confiáveis, evitando o bloqueio do acesso aos *hosts* e reduzindo a ocorrência de falsos positivos.

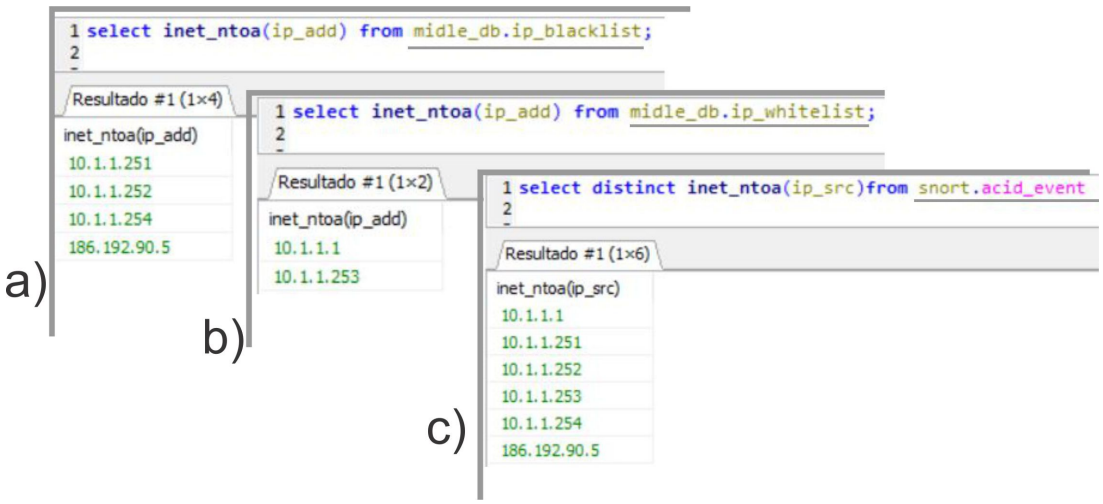

Figura 24: Informações nas tabelas de controle do sistema

A Figura 24.a mostra as informações presentes na tabela midle_db.ip_blacklist, isto é, os endereços de IP que devem ser bloqueados. Já a Figura 24.b apresenta as informações de *hosts* confiáveis, ou seja, os endereços de IP registrados que mesmo que detectado, não terão regras de bloqueio geradas. Por fim, a Figura 24.c ilustra todos os endereços de IP detectados pelo NIDS SNORT.

Sobre os testes de acesso à base de dados Mysql utilizando a porta TCP 3306, o NIDS SNORT foi capaz de identificá-los com sucesso. Também identificou as tentativas de comunicação com servidores DNS utilizando a porta UDP 53. A Figura 25 apresenta as regras responsáveis por essa detecção e os registros realizados pelo NIDS SNORT.

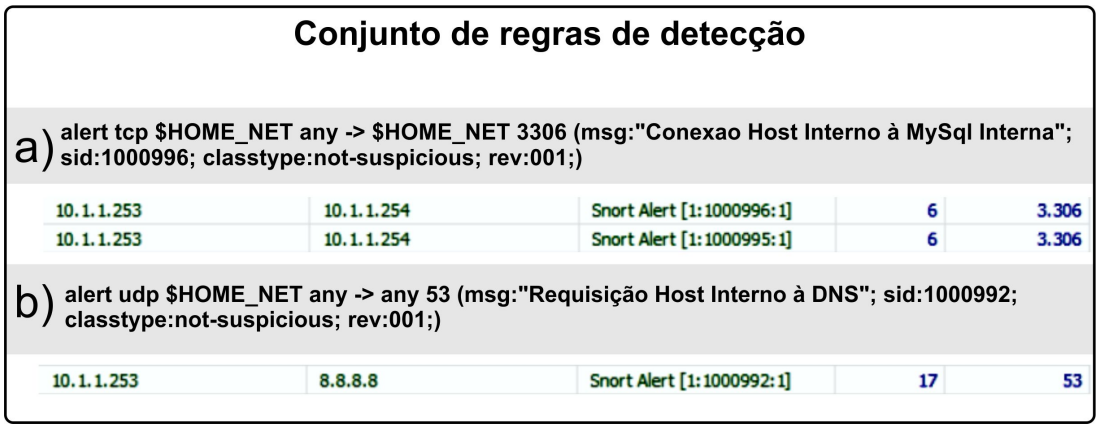

Figura 25: Exemplos de regras disparadas e registro do IDS

A Figura 25.a exibe a regra de detecção referente à detecção de acessos à base Mysql e o respectivo registro do IDS quando esta regra é disparada. Por sua vez, a Figura 25.b apresenta a regra referente à requisições DNS e o respectivo registro realizado. Em ambos os casos é possível observar nos registros as informações de endereços IP de origem e destino, a regra que se refere, código do protocolo utilizado e portas de comunicação, respectivamente.

Através dos alertas gerados pelo NIDS SNORT, o sistema desenvolvido também conseguiu identificar os endereços de IP de origem dos ataques e bloquear o acesso destes dispositivos à rede, através da inserção deste endereços de IP na lista de endereços do *firewall* do Mikrotik RouterOS, respeitando a lista de *hosts* confiáveis.

Foram verificados se os endereços de IP registrados como suspeitos, isto é, aqueles que devem ter regras de bloqueio junto ao *firewall* do Mikrotik RouterOS, de fato, estavam inseridos na lista de bloqueio SNORT_DETECTION do *firewall*. A Figura 26.a apresenta os endereços inseridos na tabela midle_db.ip_blacklist. Já a imagem 26.b a lista gerada no sistema de *firewall*, com todos os endereços devidamente inseridos.

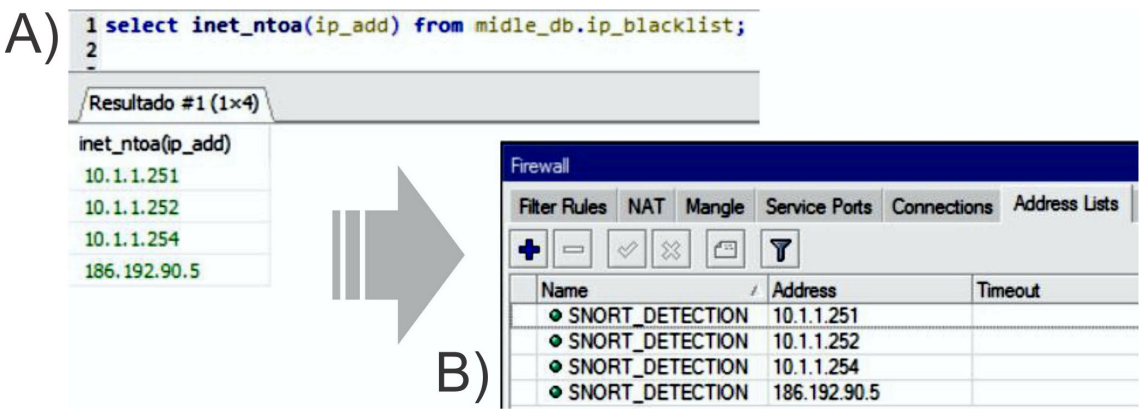

Figura 26: Comparativo: Informações de registro X Regras *firewall*

O sistema também foi capaz de realizar a remoção dos endereços de IP da lista de bloqueio do Mikrotik RouterOS, quando estes endereços de IP foram adicionados à lista de *hosts* confiáveis. Desta forma garantiu-se as configurações estabelecidas pelo administrador. Essa funcionalidade validou o controle das tabelas do sistema e a aplicação da *trigger*, apresentada pela Figura 27, responsável por manter a integridade destas regras.

```
1 BEGIN
2 DELETE FROM IP_BLACKLIST WHERE IP_ADD=NEW.IP_ADD;
3 INSERT INTO CHECK_RULES(IP_ADD) VALUES (NEW.IP_ADD);
4 END
```

Figura 27: *Trigger* aplicada à tabela de *hosts* confiáveis

Para avaliar o tempo de processamento e fornecer valores de referência ao que diz respeito ao desempenho do sistema foram aferidos valores para o tempo de processamento das informações e consumo de recursos do sistema.

- **Tempo de leitura para os 6 endereços de IP identificados**: 0.012 milissegundos. É importante destacar que neste cenário a aplicação, o NIDS e base de dados operam no mesmo dispositivo;
- **Tempo para processamento**: apresentou variação de 0,12 ao máximo de 0,36 milissegundos, considerando como tempo de processamento o intervalo de tempo necessário para realizar a leitura, validação dos 6 registros e inserção e/ou remoção no sistema de *firewall* do Mikrotik RouterOS;
- **Ciclo total**: consiste no tempo para processamento adicionado do tempo de latência até o dispositivo que receberá as regras. Este tempo de latência depende diretamente da distribuição física do dispositivo Mikrotik. Nos testes realizados o dispositivo estava conectado

diretamente ao dispositivo do NIDS SNORT, com tempo de resposta inferior a um milissegundo;

- **Consumo de memória do sistema**: O sistema em operação contínua, não ultrapassou o consumo máximo de 4,7 Megabytes de memória, operando na maioria das vezes com um consumo estável de 4,6 Megabytes;

- **Uso de processador**: Ao que se refere à utilização do processador, não foi possível aferir valores significativos, permanecendo em 0% durante toda a execução.

Com base nestes valores o administrador do sistema pode estimar o tempo de processamento e o volume máximo de dados processado em um ciclo, o que possibilita a ele ajustar o tempo de ciclo conforme a demanda de seu sistema.

Finalizando, é possível afirmar que, dentro do ambiente testado, o NIDS SNORT foi capaz de identificar ataques baseado nas regras desenvolvidas. Da mesma forma o sistema proposto cumpriu com seu objetivo e realizou o bloqueio do acesso dos *hosts* identificados como "suspeitos". Para realizar esse bloqueio, o sistema utilizou-se dos registros de detecção, através de regras inseridas automaticamente no sistema de *firewall* do Mikrotik RouterOS, formando uma ferramenta reagente aos ataques.

6 CONCLUSÃO E CONSIDERAÇÕES FINAIS

Os resultados obtidos através dos testes realizados com o trabalho desenvolvido permitem afirmar que a integração entre o IDS SNORT e o sistema de *firewall* do Mikrotik RouterOS foi possível. Tais resultados demonstraram que o sistema intermediário implementado foi capaz de aplicar, de forma automatizada, regras de controle de acesso ao sistema de *firewall* e impedir o acesso de *hosts* identificados como suspeitos.

Desta forma, o trabalho apresentado mostra que o processo de automatização de reação aos ataques identificados pelo IDS SNORT é possível e independente da intervenção direta do administrador do sistema. É válido ressaltar que o sistema proposto ainda não é totalmente automatizado, sendo dependente da intervenção do administrador para configurações de inicialização, tempo de ciclo do processo e de inserção de *hosts* estabelecidos como confiáveis.

Como melhorias possíveis aplicáveis ao sistema desenvolvido, pode-se mencionar a possibilidade de integração do presente trabalho com outros sistemas de IDS e outros sistemas de *firewall*. Contudo, estas novas integrações demandam estudos mais aprofundados sobre métodos possíveis de comunicação entre estes sistemas e padrões de registro das informações dos IDS's.

Outro ponto que pode ser estudado com atenção diz respeito ao tempo ótimo de ciclo de leitura das informações de registro, isto é, o menor tempo de ciclo necessário para processar estas informações, possibilitando ao sistema repetir o processo com menos tempo ocioso possível. Ao mesmo tempo, é passível de estudos específicos, o tema que trata sobre a performance do IDS caso o registro de ocorrências seja realizado em uma base de dados instalada em outro dispositivo e a aplicação de regras em dispositivos geograficamente distribuídos. Esse tema pode implicar diretamente no cálculo do tempo ótimo do ciclo de leitura das informações.

Outra rotina que pode ser implementa e testada é no que diz respeito à integração e inserção de regras em um conjunto de dispositivos Mikrotik. O sistema apresenta uma implementação inicial desta funcionalidade, porém esta etapa não foi testada e avaliada. Pontos como o tempo

necessário para realizar um ciclo de inserções e como tratar casos onde existe um volume grande de regras para realizar as suas aplicações podem ser foco de estudos futuros, uma vez que este trabalho considerou um dispositivo Mikrorik e um baixo volume de regras para serem aplicadas.

Por fim, além das melhorias sugeridas, é importante destacar que otimizações aplicadas à fase de detecção do IDS automaticamente são refletidas na geração de regras de proteção. Desta forma, o desenvolvimento de novos algoritmos e métodos de detecção, bem como também, o aprimoramento dos existentes, são uma boa direção para pesquisas futuras.

REFERÊNCIAS

AHMED, M.; PAL, R.; HOSSAIN, M.; BIKAS, M.; HASAN, M. NIDS: a network based approach to intrusion detection and prevention. In: COMPUTER SCIENCE AND INFORMATION TECHNOLOGY - SPRING CONFERENCE, 2009. IACSITSC '09. INTERNATIONAL ASSOCIATION OF, 2009. p.141–144.

AL-JARRAH, O.; ARAFAT, A. Network Intrusion Detection System using attack behavior classification. In: INFORMATION AND COMMUNICATION SYSTEMS (ICICS), 2014 5TH INTERNATIONAL CONFERENCE ON, 2014. p.1–6.

ANAND, Vijay. Intrusion Detection: tools, techniques and strategies. In: ND ANNUAL ACM SIGUCCS CONFERENCE ON USER SERVICES, 42., 2014, New York, NY, USA. ACM, 2014. p.69–73.

BHUYAN, M.; BHATTACHARYYA, D.; KALITA, J. Network Anomaly Detection: methods, systems and tools. *Communications Surveys Tutorials, IEEE*, v.16, n.1, 2014. p.303–336.

BISHOP, Matt. *Introduction to Computer Security*. Addison-Wesley Professional, 2004.

BRO. *The Bro Network Security Monitor*. Disponível em: <https://www.bro.org/>. Acesso em: Dezembro de 2015.

BUL'AJOUL, W.; JAMES, A.; PANNU, M. Network Intrusion Detection Systems in High-Speed Traffic in Computer Networks. In: IEEE 10TH INTERNATIONAL CONFERENCE ON E-BUSINESS ENGINEERING, 2013., 2013, Washington, DC, USA. IEEE Computer Society, 2013. p.168–175.

CHAUDHARI, N.; TIWARI, A.; THAKAR, U.; THOMAS, J. Semi-supervised classification for intrusion Detection System in networks. In: CYBERNETICS AND INTELLIGENT SYSTEMS (CIS), 2010 IEEE CONFERENCE ON, 2010. p.120–125.

COMER, Douglas. *Internetworking with TCP/IP*: principles, protocols, and architecture. Upper Saddle River, NJ, USA: Prentice-Hall, Inc., 1988.

DENNING, D.E. An Intrusion-Detection Model. *Software Engineering, IEEE Transactions on*, Fevereiro de 1987. v.SE-13, n.2, p.222–232.

HEBERLEIN, L.; DIAS, G.; LEVITT, K.; MUKHERJEE, B.; WOOD, J.; WOLBER, D. A network security monitor. In: RESEARCH IN SECURITY AND PRIVACY, 1990. PROCEEDINGS., 1990 IEEE COMPUTER SOCIETY SYMPOSIUM ON, 1990. p.296–304.

HOQUE, N.; BHUYAN, M. H.; BAISHYA, R.; BHATTACHARYYA, D.; KALITA, J. Network Attacks. *J. Netw. Comput. Appl.*, London, UK, UK, Abril de 2014. v.40, n.C, p.307–324.

HUANG, X.; WANG, X.; ZHU, S. Study on Intelligent Firewall System Combining Intrusion Detection and Egress Access Control. In: INTELLIGENT SYSTEM DESIGN AND ENGINEERING APPLICATION (ISDEA), 2010 INTERNATIONAL CONFERENCE ON, 2010. v.2, p.456–459.

HUBBALLI, N.; SURYANARAYANAN, V. Review: false alarm minimization techniques in signature-based intrusion detection systems: a survey. *Comput. Commun.*, Amsterdam, The Netherlands, The Netherlands, Agosto de 2014. v.49, p.1–17.

JUNQI, W.; ZHENGBING, H. Study of Intrusion Detection Systems (IDSs) in Network Security. In: WIRELESS COMMUNICATIONS, NETWORKING AND MOBILE COMPUTING, 2008. WICOM '08. 4TH INTERNATIONAL CONFERENCE ON, 2008. p.1–4.

LEE, J.; HWANG, S. H.; PARK, N.; LEE, S.-W.; JUN, S.; KIM, Y. S. A High Performance NIDS Using FPGA-based Regular Expression Matching. In: ACM SYMPOSIUM ON APPLIED COMPUTING, 2007. p.1187–1191.

LEI-JUN, L.; HONG, P. A Defense Model Study Based on IDS and Firewall Linkage. In: INFORMATION SCIENCE AND MANAGEMENT ENGINEERING (ISME), 2010 INTERNATIONAL CONFERENCE OF, 2010. v.2, p.91–94.

MIKROTIK. *Mikrotik*. Disponível em: <http://www.mikrotik.com/>. Acesso em: Dezembro de 2015.

NJOGU, H. W.; JIAWEI, L.; KIERE, J. N.; HANYURWIMFURA, D. A Comprehensive Vulnerability Based Alert Management Approach for Large Networks. *Future Gener. Comput. Syst.*, Amsterdam, The Netherlands, The Netherlands, Janeiro de 2013. v.29, n.1, p.27–45.

PAXSON, Vern. Bro: a system for detecting network intruders in real-time. In: CONFERENCE ON USENIX SECURITY SYMPOSIUM - VOLUME 7, 7., 1998, Berkeley, CA, USA. USENIX Association, 1998. p.3–3.

RAGHUNATH, B. R.; MAHADEO, S. N. Network Intrusion Detection System (NIDS). *Emerging Trends in Engineering & Technology, International Conference on*, Los Alamitos, CA, USA, v.0, 2008. p.1272–1277

SHERRY, J.; LAN, C.; POPA, R. A.; RATNASAMY, S. BlindBox: deep packet inspection over encrypted traffic. *SIGCOMM Comput. Commun. Rev.*, New York, NY, USA, Agosto de 2015. v.45, n.5, p.213–226.

SNORT. *The Snort Project*. Disponível em: <http://www.snort.org/>. Acesso em: Dezembro de 2015.

SOUZA, L. D. F.; LUCA, G. G. de. Lei 12.965 de 2014: democratização da internet e efeitos do marco civil na sociedade da informação. In: REVISTA PARADIGMA, 2014. p.76–96.

TANENBAUM, A. S.; WETHERALL, D. J. *Computer Networks*. 5th.ed. Upper Saddle River, NJ, USA: Prentice Hall Press, 2010.

VACCA, John R. *Computer and Information Security Handbook, Second Edition*. 2nd.ed. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 2013.

VALGENTI, V.; SUN, H.; KIM, M. S. Protecting Run-Time Filters for Network Intrusion Detection Systems. In: ADVANCED INFORMATION NETWORKING AND APPLICATIONS (AINA), 2014 IEEE 28TH INTERNATIONAL CONFERENCE ON, 2014. p.116–122.

ZAMAN, S.; KARRAY, F. TCP/IP Model and Intrusion Detection Systems. In: ADVANCED INFORMATION NETWORKING AND APPLICATIONS WORKSHOPS, 2009. WAINA '09. INTERNATIONAL CONFERENCE ON, 2009. p.90–96.