UNIVERSIDADE FEDERAL DOS VALES DO JEQUITINHONHA E MUCURI

Curso de Graduação em Sistemas de Informação

Marcos Paulo Vieira de Macedo

AVALIAÇÃO DE MECANISMOS DE DEFESA CONTRA ATAQUES DE FORÇA BRUTA VIA SSH EM ROTEADORES UTILIZANDO O IDS/IPS SNORT NA PLATAFORMA PFSENSE

Diamantina

2024

Marcos Paulo Vieira de Macedo

AVALIAÇÃO DE MECANISMOS DE DEFESA CONTRA ATAQUES DE FORÇA BRUTA VIA SSH EM ROTEADORES UTILIZANDO O IDS/IPS SNORT NA PLATAFORMA PFSENSE

Trabalho de Conclusão de Curso apresentado ao curso de graduação em Sistemas de Informação, como parte dos requisitos exigidos para a obtenção do título de Bacharel em Sistemas de Informação.

Orientador: Prof. Dr. Eduardo Pelli

Diamantina

2024

Dedico este trabalho aos meus pais, pilares da minha formação como ser humano.

AGRADECIMENTOS

Agradeço em primeiro lugar a Deus pela imensa oportunidade que me concedeu e pela constante força que me tem dado ao longo de cada dia desta trajetória. Minha profunda gratidão à minha família, que sempre me apoiou incondicionalmente. A minha mãe, Vânia, e ao meu pai, Cidinho, que foram meus pilares e fonte constante de incentivo. À minha querida irmã Jo e ao meu irmãozinho, Davi, pela compreensão e pelo amor incondicional em cada etapa desta jornada. À minha noiva, Camila, meu porto seguro, pela paciência, compreensão e amor constantes, que tornaram possível enfrentar os desafios e perseverar até o fim. Aos meus amigos, verdadeiros companheiros de jornada, que estiveram sempre presentes, seja com um ombro amigo, uma palavra de incentivo ou momentos de descontração que trouxeram alívio nos momentos mais desafiadores. Um agradecimento especial aos amigos Cezar e Samuel, cuja amizade e apoio foram inestimáveis. Aos meus estimados professores, que compartilharam conhecimento, experiência e sabedoria ao longo desta caminhada acadêmica. Seus ensinamentos foram fundamentais para o meu crescimento pessoal e profissional. Por fim, meu profundo agradecimento ao meu orientador, Eduardo Pelli, pela orientação dedicada, apoio constante e valiosas contribuições que foram essenciais para o desenvolvimento deste trabalho. A todos vocês, meu sincero obrigado por fazerem parte desta conquista e por tornarem possível a realização deste sonho.

Uma mudança deixa sempre patamares para uma nova mudança.

(MACHIAVELLI, 1532)

RESUMO

Com o crescente aumento dos meios digitais proporcionados pelo avanço tecnológico e difusão da Internet, se tornou comum o armazenamento de dados e informações pessoais em diversos dispositivos digitais como computadores, celulares e até mesmo em *sites* de armazenamento de dados em nuvem. Obviamente estes dados são protegidos por um tipo de *login* comumente utilizado no formato de usuário e senha. É certo que ter acesso a algum dispositivo alheio pode ser algo valioso para uma pessoa má intencionada. Neste intuito o presente documento aborda a exploração dessas senhas através de uma prática *hacker* denominada: ataque de força bruta. O objetivo do ataque é descobrir a senha de determinado acesso através de uma metodologia de tentativa e erro. Prever e se proteger desses ataques não é algo trivial, portanto o trabalho busca discorrer sobre o tema através de um estudo prático, nesse sentido, é utilizado um cenário de referência como ponto de partida, um ambiente informatizado para experimentação e discussão, com abordagens envolvendo ataques contra *hosts* locais e remotos. Manipulando soluções *Linux*, sistema operacional amplamente utilizado no contexto da Internet. O estudo analisou certos filtros de conteúdo empregados por especialistas com o objetivo de reduzir os riscos de ataques de força bruta, demonstrando que tais filtros são ineficazes quando se trata de um ataque originado de uma vulnerabilidade em um serviço autêntico liberado na rede. Para solucionar essa vulnerabilidade, o trabalho incluiu a instalação do *Snort* no *pfSense*, um *IDS/IPS* capaz de identificar e bloquear o tráfego malicioso na rede. O *Snort* apresentou resultados satisfatórios em critérios como eficácia no bloqueio e eficiência contra esses ataques, demonstrando abordagens distintas na prevenção de ataques de força bruta. O estudo destacou a importância de uma abordagem multifacetada na segurança de redes, combinando *firewalls* e sistemas de detecção e prevenção de intrusões para fortalecer a segurança e mitigar ameaças cibernéticas.

Palavras-chave: *Firewalls*. Redes de computadores. *Mikrotik*. *SSH*.

ABSTRACT

With the increasing increase in digital media provided by technological advancement and the spread of the internet, it has become common to store data and personal information on various digital devices such as computers, cell phones and even cloud data storage sites. Obviously this data is protected by a commonly used type of login in the username and password format. It is true that having access to someone else's device can be valuable to a person with bad intentions. To this end, this document addresses the exploitation of these passwords through a hacking practice called: brute force attack. The objective of the attack is to discover the password for a given access through a trial and error methodology. Predicting and protecting yourself from these attacks is not trivial, therefore the work seeks to discuss the topic through a practical study, in this sense, a reference scenario is used as a starting point, a computerized environment for experimentation and discussion, with approaches involving attacks against local and remote hosts. Handling Linux solutions, an operating system widely used in the context of the internet. The study analyzed certain content filters used by experts with the aim of reducing the risks of brute force attacks, demonstrating that such filters are ineffective when it comes to an attack originating from a vulnerability in an authentic service released on the network. To resolve this vulnerability, the work included installing Snort on pfSense, an IDS/IPS capable of identifying and blocking malicious traffic on the network. Snort presented satisfactory results in criteria such as blocking effectiveness and efficiency against these attacks, demonstrating different approaches to preventing brute force attacks. The study highlighted the importance of a multifaceted approach to network security, combining firewalls and intrusion detection and prevention systems to strengthen security and mitigate cyber threats.

Keywords: Firewalls. Computer networks. Mikrotik. SSH.

LISTA DE ILUSTRAÇÕES

LISTA DE TABELAS

LISTA DE ABREVIATURAS E SIGLAS

UFVJM	Universidade Federal dos Vales do Jequitinhonha e Mucuri
ANPD	Autoridade Nacional de Proteção de Dados
ARP	Address Resolution Protocol
ARPANET	Advanced Research Projects Agency Network
AWS	Amazon Web Services
CERN	European Organization for Nuclear Research
CIDAL	Confidencialidade, Disponibilidade, Autenticidade, Integridade e Legalidade
CPU	Central Processing Units
CSMA/CA	Carrier Sense Multiple Access with Collision Avoidance
CSMA/CD	Carrier Sense Multiple Access with Collision Detection
DARPA	Defense Advanced Research Projects Agency
DDOS	Distributed Denial of Service
DNS	Domain Name System
DoS	Denial of Service
DSL	Digital Subscriber Line
ELK	Elasticsearch, Logstash e Kibana
EVE-NG	Emulated Virtual Environment Next Generation
FreeBSD	Free Berkeley Software Distribution
FTP	File Transfer Protocol
HTTP	Hypertext Transfer Protocol
HTTPS	Hypertext Transfer Protocol Secure
IDs	Identity

IDS/IPS	Intrusion detection System / Intrusion Prevention System
IoT	Internet of Things
LANS	Local Area Network
LGPD	Lei Geral de Proteção de Dados Pessoais
MAC	Media Access Control
MAN	Metropolitan Area Network
MIB	Management Information Base
NBR	Norma Brasileira Regulamentadora
OSI	Open Systems Interconnection
PAN	Personal Area Network
PIN	Personal Identification Number
SNMP	Simple Network Management Protocol
SSH	Protocolo Secure Shell
TCP/IP	Transmission Control Protocol/Internet Protocol
TI	Tecnologia da Informação
UDP	User Datagram Protocol
UFVJM	Universidade Federal dos Vales do Jequitinhonha e Mucuri
VM	Virtual Machine
VPN	Virtual Private Network
WAN	Wide Area Network
WEB	World Wide Web
Wi-Fi	Wireless Fidelity
WLANS	Wireless Local Area Network

WWW	World Wide Web
XSS	Cross-Site Scripting

SUMÁRIO

1 INTRODUÇÃO

Há alguns anos, os teóricos da comunicação consideravam apenas a imprensa, o cinema, a rádio e a televisão como os meios de comunicação de massa (DEFLEUR; BALL-ROKEACH, 1993). No entanto, na década de 70, a comunicação por meio do computador passou a ser um elemento essencial na construção da futura infraestrutura mundial, na década de 80, a Internet já era um projeto de pesquisa que envolvia algumas dezenas de sites (COMER, 2016).

A partir da década de 90, o crescimento da Internet e sua transformação em um sistema de comunicação produtivo, alcançando milhões de pessoas em todos os países do mundo, se consolidou. Nos anos seguintes, muitos usuários já obtiveram acesso à Internet de alta velocidade por meio de conexões a cabo, *Digital Subscriber Line (DSL)*, fibra óptica e tecnologias sem fio. Com esse avanço significativo, a comunicação por computadores estabeleceu-se como um poderoso meio de comunicação de massa do século XXI (COMER, 2016).

A acessibilidade facilitada a vasta rede tem levado a um rápido crescimento no número de usuários. Em 2019, cerca de 134 milhões de brasileiros utilizaram a Internet, representando mais da metade da população do país, seja por meio de dispositivos como celulares, *tablets*, computadores ou outros aparelhos capazes de navegar na rede mundial (CETIC, 2020).

A crescente integração de redes de computadores nas operações comerciais modernas está redefinindo fundamentalmente a forma como as organizações conduzem seus negócios, abrangendo áreas como planejamento estratégico, produção, logística e interações com clientes. (OSTERLE; FLEISCH; ALT, 2001), nos estágios iniciais do surgimento da Internet, a segurança não foi a preocupação primordial (TANENBAUM, 2002). Entretanto, à medida que milhões de usuários começaram a realizar transações financeiras, fazer compras, organizar seus impostos e outras atividades *online*, a segurança da Internet manifestou-se como um problema potencialmente sério (TANENBAUM, 2002).

Conforme Diorio *et al.* (2019), a segurança da Internet enfrenta muitos desafios causados de forma intencional. Dentre os diversos incidentes que ocorrem, os ataques de força bruta são destacados como uma das principais e mais populares ameaças da atualidade (DIORIO *et al.*, 2019). Nesses ataques, os agressores tentam adivinhar *logins* e senhas de acesso de usuários legítimos de um determinado sistema ou serviço de rede, no qual utilizam o método de tentativa e erro (DIORIO *et al.*, 2019).

No decorrer do ano de 2020, registrou-se um aumento significativo no número de notificações de ataques de força bruta direcionados a sistemas de gerenciamento de conteúdo (CERT, 2020). Já em 2022, os ataques cibernéticos no Brasil tiveram um crescimento exponencial, com um aumento de 94% em comparação aos números do primeiro semestre do mesmo ano, totalizando cerca de 31,5 bilhões de tentativas de invasões nos sistemas de empresas e organizações (OLIVEIRA, 2022).

Durante a navegação na Internet, é necessário utilizar a arquitetura chamada *Transmission Control Protocol/Internet Protocol (TCP/IP)*, que permite o envio e recebimento de dados entre computadores e dispositivos (BODNAR, 2022). O intuito desse modelo *TCP/IP* é viabilizar a conexão de dispositivos à Internet, possibilitando a comunicação em todas as redes (BODNAR, 2022). Geralmente, essas conexões são protegidas por equipamentos conhecidos como *firewalls*, que definem regras para o que é permitido enviar e receber. Além disso, existem mecanismos mais robustos, como os sistemas de detecção e prevenção de intrusão, conhecidos pelo nome *Intrusion detection System / Intrusion Prevention System (IDS/IPS)*, que podem ser utilizados para garantir a segurança da rede.

Os *IDS/IPS* são dispositivos de segurança de rede que monitoram e analisam o tráfego de dados em busca de atividades suspeitas ou maliciosas. Enquanto o *IDS* é responsável por identificar intrusões e eventos de segurança, o *IPS* tem a capacidade adicional de prevenir e bloquear essas atividades indesejadas. Ambos são essenciais para assegurar a integridade e a proteção das redes de computadores (TANENBAUM; WETHERALL, 2011).

Contudo, o presente trabalho buscou colaborar com pesquisas sobre o tema e complementar estudos anteriores, explorando de forma prática os ataques de força bruta. Através de um cenário de referência, um ambiente computacional é utilizado para experimentação e discussão. As experimentações e discussões realizadas têm como objetivo permitir a identificação de abordagens e soluções comumente utilizadas nesse tipo de ataque.

2 OBJETIVO

2.1 Objetivo Geral

O presente trabalho teve como objetivo configurar e avaliar a eficácia de um Sistema de Detecção de Intrusões (*IDS*) e Sistema de Prevenção de Intrusões (*IPS*) de código aberto, especialmente no que se refere à minimização dos riscos associados a ataques de força bruta direcionados a servidores *Protocolo Secure Shell (SSH)* em roteadores da fabricante *Mikrotik*.

2.2 Objetivos Específicos

Para a obtenção do objetivo geral definiu-se então alguns objetivos específicos:

- Configurar o sistema *IDS/IPS* Snort em um ambiente controlado;
- Implementar uma infraestrutura de testes que simule cenários realistas de ataques de força bruta;
- Coletar e analisar dados gerados pelos testes para avaliar a eficácia do sistema de detecção e prevenção de intrusões na detecção e mitigação de ataques de força bruta direcionados ao serviço de servidor *SSH* em roteadores *Mikrotik*;
- Comparar os resultados obtidos com estudos relacionados e discutir as vantagens e desafios encontrados durante a implementação e integração do sistema;
- Analisar e concluir a viabilidade e eficácia do sistema de detecção e prevenção de intrusão Snort no sentido de mitigar os riscos associados a ataques de força bruta em servidores SSH de roteadores *Mikrotik*.

3 REVISÃO DE LITERATURA

A revisão da literatura neste estudo, apresentada nas próximas seções, é essencial para uma compreensão aprofundada do tema. Portanto, foram fornecidas definições e descrições de termos importantes para fundamentar teoricamente este trabalho de conclusão de curso.

3.1 Redes de Computadores

As redes de computadores surgiram em 1960, como uma forma de conectar computadores distantes e compartilhar dados e recursos. Um dos primeiros projetos foi a *Advanced Research Projects Agency Network (ARPANET)*, criada pela agência militar americana *Defense Advanced Research Projects Agency (DARPA)*, que usava a comutação de pacotes para transmitir informações entre os nós da rede (MENDES, 2020).

A partir de 1970, as redes de computadores se expandiram e se diversificaram, com o surgimento de novas tecnologias, protocolos e padrões. Alguns exemplos são as redes locais *(LANs)*, as redes sem fio *(WLANs)*, o modelo cliente-servidor, o protocolo *TCP/IP* e o sistema de nomes de domínios *(DNS)* (MARTINS, 2019).

O ano de 1990 marcou o início da popularização da Internet, que é uma rede global composta por diversas outras redes interligadas (SOUZA, 2023). A Internet possibilitou o acesso a uma grande quantidade de informações e serviços, além de facilitar a comunicação e a colaboração entre pessoas e organizações (MARTINS, 2019).

A teoria das redes de computadores é um campo que estuda os princípios, as arquiteturas, os protocolos e as aplicações das redes. Ela aborda temas como os meios físicos de transmissão, as topologias e serviços de rede, os modelos de referência (*Open Systems Interconnection (OSI)* e o *TCP/IP*), as camadas de rede (como a física, a enlace, a rede, a transporte e a aplicação), os algoritmos de roteamento e controle de congestionamento, a segurança e a qualidade de serviço das redes (MARTINS, 2019).

Buscando mais conceitos, Lucas *et al.* (2021), cita que redes de computadores são sistemas de informação que permitem a conexão à Internet e a troca de informações por meio de computadores. Isso pode acontecer por meios físicos, como fios e cabos, ou mesmo por meio de redes sem fio (MENDES, 2020).

Segundo Souza (2023), as redes de computadores servem para diversas finalidades, como: compartilhar arquivos e recursos entre dispositivos, acessar e utilizar aplicativos pela

Internet, comunicar-se por voz, vídeo, texto e imagens, realizar compras e vendas online, jogar online.

Existem vários tipos de redes de computadores, que se diferenciam pelo tamanho, alcance, topologia e arquitetura. Lucas *et al.* (2021), cita alguns dos principais tipos, são elas:

- *LAN (Local Area Network)*: rede local que abrange uma área limitada, como uma casa, uma escola ou uma empresa;
- *WAN (Wide Area Network)*: rede de longa distância que abrange uma área geográfica maior, como um país ou um continente;
- *WLAN (Wireless Local Area Network)*: rede local sem fio que usa ondas de rádio ou infravermelho para transmitir dados;
- *MAN (Metropolitan Area Network)*: rede metropolitana que abrange uma área urbana, como uma cidade ou um campus universitário;
- *PAN (Personal Area Network)*: rede pessoal que conecta dispositivos próximos a uma pessoa, como um celular, um fone de ouvido ou um relógio inteligente.

O modelo de rede mais utilizado atualmente é o modelo cliente-servidor, que permite que os dados sejam armazenados em servidores centrais, onde os dispositivos do usuário final (clientes) possam acessar essas informações. Esse modelo é usado em muitos serviços da Internet, como *email*, redes sociais e comércio eletrônico (MENDES, 2020).

Lima (2019), cita algumas vantagens e desvantagens das redes de computadores. Entre as vantagens, destacam-se: a facilidade de comunicação e a colaboração entre pessoas e organizações; permitem o acesso a informações e serviços de forma rápida e conveniente; reduzem custos e aumentam a eficiência dos processos; ampliam as possibilidades de aprendizagem e entretenimento. No entanto, há também desvantagens, como a exigência de investimento em infraestrutura e manutenção; a dependência da disponibilidade e da qualidade da conexão; os riscos de segurança e privacidade dos dados; e possíveis problemas de saúde e ambientais relacionados à exposição à radiação e ao consumo de energia.

Desta forma, Mendes (2020), cita que as redes de computadores são muito úteis para a comunicação, o acesso à informação e a realização de diversas atividades. No entanto, elas também apresentam alguns perigos e riscos que deve-se estar atentos e nos proteger.

Alguns dos principais perigos das redes de computadores são:

- *Malware*: são programas maliciosos que podem infectar os computadores e causar danos, como roubar dados, apagar arquivos, alterar configurações, espionar atividades e até bloquear o acesso ao sistema (MARQUEZIN; SILVA; PINTO, 2021).

- Espionagem industrial: é a prática de obter informações confidenciais ou estratégicas de uma empresa ou organização por meios ilícitos, como invasão de sistemas, interceptação de comunicações ou roubo de documentos (FERREIRA; AMADIO, 2022).
- *Ransomware*: é um tipo de *malware* que sequestra os dados do computador e exige um resgate para liberá-los. Geralmente, o *ransomware* criptografa os arquivos e impede o acesso ao sistema, mostrando uma mensagem com as instruções para o pagamento (JÚNIOR; LINS, 2023).
- Ataques *Distributed Denial of Service (DDoS)*: são ataques distribuídos de negação de serviço, que consistem em sobrecarregar um servidor ou uma rede com uma grande quantidade de requisições falsas, impedindo que os usuários legítimos possam acessar os serviços disponíveis (LIMA, 2019).
- Violação de privacidade: é a exposição ou o uso indevido de dados pessoais ou sensíveis de usuários ou empresas, que podem ser obtidos por meio de invasão de sistemas, *phishing*, *spyware*, *cookies* ou outras técnicas .
- Ataque de força bruta: é uma tentativa de violar uma senha, um nome de usuário, uma chave de criptografia ou uma página da *World Wide Web (WEB)* oculta, usando uma abordagem de tentativa e erro e testando todas as combinações possíveis de caracteres até encontrar a correta. Esse tipo de ataque pode ser minimizado pelo uso de senhas longas e complexas, ou por mecanismos de bloqueio ou limitação de tentativas.

3.2 Internet

A Internet é uma rede global de computadores interligados por protocolos de comunicação que permitem o acesso e a troca de informações entre usuários e serviços (NASCIMENTO *et al.*, 2020). A Internet tem uma história complexa e fascinante, que envolve aspectos políticos, econômicos, sociais e tecnológicos. Ela surgiu em 1960, como um projeto militar dos Estados Unidos para criar uma rede de comunicação resistente a ataques nucleares, chamada *ARPANET*. Essa rede foi a precursora da Internet, pois usava o protocolo *TCP/IP* que é o padrão atual para a transmissão de dados na rede (EISENBERG, 2019).

No ano de 1970, a *ARPANET* se expandiu e se conectou com outras redes acadêmicas e científicas, formando a Internet. Nesse período, surgiram também os primeiros serviços de *e-mail*, *chat* e *File Transfer Protocol (FTP)* (transferência de arquivos). Na década seguinte,

a Internet se tornou mais acessível ao público em geral, com o surgimento dos primeiros provedores comerciais e dos domínios de topo (.com, .org, .edu, etc.) (EISENBERG, 2019).

O ano de 1990 foi marcada pela rápida expansão da Internet, com o advento da *World Wide Web (WWW)*, que é um sistema de hipertexto que permite a navegação entre páginas Web por meio de *links* (NASCIMENTO *et al.*, 2020). A *WWW* foi criada por Tim Berners-Lee, um cientista britânico que trabalhava no *CERN* (Organização Europeia para a Pesquisa Nuclear). Ele também desenvolveu o primeiro navegador *Web (browser)*, chamado Mosaic (EISENBERG, 2019).

A *WWW* revolucionou a Internet, pois tornou possível o acesso a uma grande variedade de conteúdo multimídia (texto, imagem, som e vídeo) de forma fácil e rápida. Com isso, surgiram novos serviços e aplicações, como os motores de busca (Google), as redes sociais (Facebook), o comércio eletrônico (Amazon), os *blogs*, os *podcasts*, os vídeos *online* (YouTube), entre outros (BARROS; SOUZA; TEIXEIRA, 2021).

A Internet continua evoluindo até hoje, com novas tecnologias e tendências, como a Internet móvel (*smartphones*), a Internet das coisas (*IoT*), a computação em nuvem, a inteligência artificial, a realidade virtual e aumentada, a *big data*, a *blockchain*, etc. Essas inovações trazem novos desafios e oportunidades para a sociedade, a economia e a cultura (NASCIMENTO *et al.*, 2020).

A Internet também tem princípios e diretrizes que orientam o seu uso e desenvolvimento. Alguns desses princípios são: a liberdade de expressão, a privacidade, a neutralidade da rede, a segurança, a inclusão digital e a governança multissetorial (BARROS; SOUZA; TEIXEIRA, 2021). Esses princípios visam garantir os direitos e deveres dos usuários da Internet, bem como promover uma rede aberta, democrática e participativa.

No Brasil, esses princípios estão consagrados no Marco Civil da Internet (Lei nº 12.965/2014), que é uma lei que estabelece as normas para o uso da Internet no país (CALGAROTO, 2021). O Marco Civil foi resultado de um amplo processo de consulta pública e debate social, envolvendo diversos atores da sociedade civil, do governo e do setor privado. Ele é considerado um modelo internacional de legislação sobre a Internet (CALGAROTO, 2021).

3.3 Protocolos de rede de computadores

Os protocolos de rede de computadores são um conjunto de regras e padrões que permitem a comunicação e a troca de dados entre dispositivos conectados em uma rede (PEREIRA *et al.*, 2021). Eles funcionam como uma linguagem comum que pode ser entendida por diferentes tipos de equipamentos, sistemas operacionais e aplicações (SANTOS; SANTOS, 2022).

Existem vários tipos de protocolos de rede, que podem ser classificados de acordo com a camada do modelo *TCP/IP* em que atuam. Esses modelos são formas de organizar as funções dos protocolos em camadas hierárquicas, facilitando o seu entendimento e padronização (SANTOS; SANTOS, 2022).

A seguir, é apresentado alguns dos principais tipos de protocolos de rede e suas funções, seguindo o modelo *TCP/IP*. Segundo (TANENBAUM; WETHERALL, 2011), esse modelo é composto por cinco camadas: aplicação, transporte, rede, enlace e física.

Camada de Aplicação: é a mais próxima do usuário final e engloba os protocolos que permitem o acesso a serviços e recursos na rede, como navegação Web, envio de *e-mails*, transferência de arquivos, etc. Alguns exemplos de protocolos dessa camada são:

- *HTTP (Hypertext Transfer Protocol)*: é o protocolo usado para transmitir páginas Web entre um servidor e um navegador. Ele usa o formato de texto para enviar requisições e respostas, seguindo o modelo cliente-servidor.
- *HTTPS (Hypertext Transfer Protocol Secure)*: é uma versão segura do *HTTP*, que usa criptografia para proteger os dados transmitidos. Ele usa um certificado digital para autenticar o servidor e estabelecer uma conexão segura com o cliente.
- *SSH (Secure Shell)*: é um protocolo que permite a conexão segura e remota entre um cliente e um servidor. Ele usa criptografia para proteger as informações transmitidas e autenticar o usuário por meio de uma senha ou uma chave *SSH*. Ele permite que o usuário execute comandos e transfira arquivos no servidor.
- *SNMP (Simple Network Management Protocol)*: é amplamente utilizado para a supervisão e administração de redes e tem como função a coleta de informações vitais sobre o desempenho da rede, diagnósticos de falhas, configurações de dispositivos e até mesmo a realização de ações corretivas remotamente.

- *FTP (File Transfer Protocol)*: é um protocolo usado para transferir arquivos entre um cliente e um servidor na rede. Ele usa duas conexões *TCP*: uma para enviar comandos e respostas e outra para enviar os dados dos arquivos.
- *DNS (Domain Name System)*: é um protocolo usado para resolver nomes de domínio em endereços *IP*. Ele funciona como um sistema hierárquico e distribuído de servidores que armazenam e fornecem as informações sobre os domínios na rede.

Camada de Transporte: é responsável por estabelecer, manter e encerrar as conexões entre os dispositivos na rede (SOUZA, 2023). Ela também controla o fluxo, a confiabilidade e a segmentação dos dados transmitidos. Os principais protocolos dessa camada são:

- *TCP (Transmission Control Protocol)*: é um protocolo orientado à conexão, confiável e baseado em fluxo. Ele garante que os dados sejam entregues na ordem correta, sem erros ou perdas, usando mecanismos de confirmação, retransmissão, controle de congestionamento e janela deslizante (ROSA *et al.*, 2022). Ele combina o *TCP*, que garante a entrega confiável dos dados com o *IP*, que identifica os endereços dos dispositivos na rede. O *TCP/IP* foi desenvolvido em 1969 pelo Departamento de Defesa dos Estados Unidos, como parte do projeto *ARPANET* que visava criar uma rede de comunicação resistente a falhas e ataques.
- *UDP (User Datagram Protocol)*: é um protocolo não orientado à conexão, não confiável e baseado em datagramas. Ele não garante a entrega, a ordem ou a integridade dos dados transmitidos, mas oferece uma comunicação mais rápida e simples, usando menos recursos da rede (SOUZA, 2023).

Camada de Rede: é responsável por rotear os pacotes de dados entre os dispositivos na rede. Ela também define os endereços lógicos dos dispositivos e os mecanismos de fragmentação e remontagem dos pacotes (ROSA *et al.*, 2022). O principal protocolo dessa camada é:

- *IP (Internet Protocol)*: é o protocolo que define o formato e o endereçamento dos pacotes de dados na rede. Ele usa um endereço de 32 *bits (IPv4)* ou de 128 *bits (IPv6)* para identificar cada dispositivo na rede. Ele também permite a fragmentação e a remontagem dos pacotes, caso eles sejam maiores que o tamanho máximo permitido pelo meio de transmissão.

Camadas de Enlace e Física: são as mais próximas do meio físico de transmissão e engloba os protocolos que permitem a comunicação entre os dispositivos e a rede. Elas também definem os endereços físicos dos dispositivos e os mecanismos de acesso ao meio (DAVOGLIO

et al., 2021). Daleffe, Amadio e Gavilan (2020), cita alguns exemplos de protocolos dessa camada, são eles:

- *Ethernet*: é um protocolo que define o formato e o endereçamento dos quadros de dados na rede. Ele usa um endereço de 48 *bits*, chamado de *Media Access Control (MAC)*, para identificar cada dispositivo na rede. Ele também usa um método de acesso ao meio chamado *Carrier Sense Multiple Access with Collision Detection (CSMA/CD)*, que permite que vários dispositivos compartilhem o mesmo meio, detectando e resolvendo colisões.
- *Wi-Fi (Wireless Fidelity)*: é um protocolo que define o formato e o endereçamento dos quadros de dados na rede sem fio. Ele usa um endereço *MAC* para identificar cada dispositivo na rede. Ele também usa um método de acesso ao meio chamado *Carrier Sense Multiple Access with Collision Avoidance (CSMA/CA)*, que permite que vários dispositivos compartilhem o mesmo meio, evitando colisões.
- *ARP (Address Resolution Protocol)*: é um protocolo que permite descobrir o endereço *MAC* de um dispositivo na rede, a partir do seu endereço *IP*. Ele funciona enviando uma mensagem de *broadcast* para todos os dispositivos na rede, perguntando quem possui o endereço *IP* desejado. O dispositivo que possui o endereço *IP* responde com uma mensagem *unicast*, informando o seu endereço *MAC*.

3.4 Segurança da Informação

Segurança da informação é um conjunto de ações e estratégias para proteger dados e informações sigilosas de possíveis violações ou ataques (LOURENÇO; DUARTE, 2020). Ela é aplicada em diferentes setores e contextos, como empresas, instituições públicas, redes sociais e dispositivos pessoais. A segurança da informação se baseia em cinco pilares: confidencialidade, disponibilidade, autenticidade, integridade e legalidade (OLIVEIRA *et al.*, 2019).

Segundo Freund, Sembay e Macedo (2019), é mencionado que:

- Confidencialidade: é o que garante que as informações só sejam acessadas por pessoas autorizadas, evitando vazamentos, espionagem ou invasões.
- Disponibilidade: é o que garante que as informações e os sistemas estejam sempre disponíveis para as pessoas autorizadas, evitando interrupções, falhas ou ataques.
- Autenticidade: é o que garante que as informações sejam de fato provenientes de suas fontes originais, evitando falsificações, adulterações ou fraudes.

- Integridade: é o que garante que as informações não sejam modificadas ou destruídas sem autorização, evitando corrupções, erros ou sabotagens.

Esses cinco pilares, conhecidos como CIDAL, são fundamentais para a implementação de um sistema de segurança da informação eficiente e confiável.

No Brasil, existem algumas leis e normas que regulam a segurança da informação, como a Lei Geral de Proteção de Dados Pessoais (LGPD) que entrou em vigor em 2020 e a Norma Brasileira Regulamentadora (NBR) ISO/IEC 27001, que estabelece requisitos para um sistema de gestão de segurança da informação (LEME; BLANK, 2020).

A LGPD tem como objetivo garantir maior privacidade, segurança e transparência no tratamento de dados pessoais, tanto de pessoas físicas quanto jurídicas. Ela se aplica a qualquer atividade que envolva o uso de dados pessoais, como coleta, armazenamento, compartilhamento e eliminação (RAPôSO *et al.*, 2019).

A LGPD também criou a Autoridade Nacional de Proteção de Dados (ANPD) que é o órgão responsável pela fiscalização, orientação e aplicação de sanções em caso de violação da lei. As sanções podem variar desde uma simples advertência até multas que podem chegar a 2% do faturamento da empresa ou até R$ 50 milhões por infração (LEME; BLANK, 2020).

A NBR ISO/IEC 27001 é uma norma internacional que define os requisitos para implementar, manter e melhorar um sistema de gestão de segurança da informação. A NBR ISO/IEC 27001 é uma norma voluntária, ou seja, as empresas podem optar por adotá-la ou não. No entanto, ela traz benefícios como maior credibilidade, confiança, competitividade e conformidade com as boas práticas de segurança da informação (RAPôSO *et al.*, 2019).

Segundo LourenÇo e Duarte (2020), as empresas podem atuar na segurança da informação de diversas formas, como:

- Criar um programa de segurança da informação com políticas, procedimentos, responsabilidades e controles definidos;
- Educar os funcionários sobre os riscos, as medidas de prevenção e as boas práticas de segurança da informação;
- Instalar antivírus, *firewall* e outras ferramentas de proteção nos dispositivos e nas redes da empresa;
- Monitorar e auditar os sistemas e os processos de segurança da informação, identificando e corrigindo possíveis vulnerabilidades;
- Realizar *backups* periódicos dos dados e armazená-los em locais seguros;

Seguir as normas e as leis de segurança da informação, como a LGPD e a NBR ISO/IEC 27001, evita multas e penalidades. Oliveira *et al.* (2019), menciona que as pessoas também podem se defender de ataques cibernéticos, adotando algumas medidas simples, como:

- Não informar dados pessoais nas redes e nunca enviar senhas, mesmo que seja por mensagens privadas;
- Não abrir *e-mails*, *links* ou anexos suspeitos que podem conter vírus ou golpes;
- Não acessar sites ou redes *wi-fi* desconhecidos ou não confiáveis, que podem expor os dados a *hackers*;
- Usar senhas fortes, diferentes e complexas para cada conta ou serviço e trocá-las periodicamente;
- Atualizar os *softwares* e os aplicativos dos dispositivos, mantendo-os com as últimas versões de segurança.

A segurança da informação afeta as empresas de diversas formas, pois elas lidam com dados valiosos de clientes, fornecedores, parceiros e funcionários. A falta de segurança pode gerar prejuízos financeiros, perda de credibilidade, danos à imagem e à reputação, além de processos judiciais e multas (OLIVEIRA *et al.*, 2019).

As empresas que prestam serviços de segurança da informação também estão sujeitas a multas e penalidades se não cumprirem as normas e os contratos estabelecidos com seus clientes. Além disso, elas devem garantir a qualidade e a eficiência dos seus serviços, bem como a capacitação e a ética dos seus profissionais (FREUND; SEMBAY; MACEDO, 2019).

3.5 Vulnerabilidades, ameaças e técnicas de ataque

A história da segurança da informação teve início por volta de 1950, quando as pessoas começaram a perceber que havia valor intrínseco nos dados (GUAREZI, 2019). Desde então, a evolução da segurança da informação tem sido marcada por avanços tecnológicos, desafios crescentes e a necessidade de proteger as informações contra as ameaças cibernéticas (SANTOS; LAURENCE, 2022).

Segundo Taliani (2022), as vulnerabilidades são falhas ou brechas na segurança de um sistema, rede ou aplicação que podem ser exploradas por agentes maliciosos para causar danos, roubar dados, executar códigos maliciosos ou obter acesso não autorizado.

Os tipos mais comuns de vulnerabilidades são: vulnerabilidades de rede, *softwares* desatualizados e ausência de uma política de segurança da informação bem estruturada (GUAREZI, 2019). As desvantagens de ter vulnerabilidades são: exposição a riscos de perda de dados, violação de privacidade, comprometimento da reputação, prejuízos financeiros e legais, entre outros (TALIANI, 2022).

Já Júnior e Lins (2023), menciona que as ameaças são as ações ou intenções dos agentes maliciosos que visam explorar as vulnerabilidades e comprometer a segurança do sistema, rede ou aplicação. A história das ameaças cibernéticas está intimamente ligada à história da segurança da informação, pois ambas evoluíram em resposta uma à outra. As primeiras ameaças cibernéticas surgiram no ano de 1970, com a criação dos primeiros vírus e *worms*. Desde então, as ameaças se tornaram mais sofisticadas e diversificadas, incluindo *malware*, *ransomware*, *phishing*, ataques de engenharia social, entre outros (SANTOS; LAURENCE, 2022).

Segundo Júnior e Lins (2023), os tipos mais comuns de ataques cibernéticos são: ataques de negação de serviço *Denial of Service (DoS)*, ataques de injeção, ataques de sequestro de sessão, ataques de *Cross-Site Scripting (XSS)*, ataques de quebra de criptografia, entre outros. Segue abaixo cada um deles:

- Ataques de negação de serviço *(DoS)*: são ataques que tentam sobrecarregar um site ou um serviço Web com tráfego excessivo, impedindo que os usuários legítimos possam acessá-lo. Uma forma de evitar esses ataques é usar ferramentas de mitigação de *DDoS*, como o *CloudFlare* que podem filtrar o tráfego malicioso e proteger os recursos do servidor;
- Ataques de injeção: são ataques que inserem código malicioso em um aplicativo Web, explorando vulnerabilidades na entrada de dados ou na linguagem de programação. Esses ataques podem permitir que os invasores acessem, modifiquem ou excluam dados sensíveis, ou executem comandos arbitrários no servidor. Uma forma de evitar esses ataques é validar e filtrar os dados de entrada, usar consultas parametrizadas e escapar dos caracteres especiais;
- Ataques de sequestro de sessão: são ataques que assumem o controle de uma sessão de usuário autenticado, enganando o servidor Web ou o navegador. Esses ataques podem permitir que os invasores realizem ações não autorizadas em nome do usuário, como transferir dinheiro, alterar senhas ou roubar informações pessoais. Uma forma de evitar esses ataques é usar *cookies* seguros e criptografados, gerar identidade *(*(IDs) de sessão aleatórios e complexos, e implementar mecanismos de *logout* e expiração de sessão;

- Ataques de quebra de criptografia: são ataques que tentam decifrar dados criptografados, como senhas, chaves ou mensagens, usando métodos de tentativa e erro ou explorando falhas nos algoritmos ou nas implementações de criptografia. Esses ataques podem permitir que os invasores acessem dados confidenciais ou comprometam a integridade ou a autenticidade dos dados. Uma forma de evitar esses ataques é usar algoritmos e protocolos de criptografia fortes e atualizados, gerar senhas e chaves longas e aleatórias e armazenar os dados criptografados de forma segura.

As consequências desses ataques podem ser: indisponibilidade de serviços, roubo de informações, alteração de dados, controle remoto de dispositivos, extorsão, fraude, entre outros (GUAREZI, 2019).

Edmundo (2021), também destaca ataques de força bruta como um tipo de ataque cibernético que consiste em tentar violar as credenciais da vítima (como nome de usuário e/ou senha), no qual um *hacker* (ou mais *hackers*) usa uma abordagem de tentativa e erro para tentar adivinhar as senhas da vítima, um *Personal Identification Number (PIN)*, um código de segurança ou uma chave de criptografia, tudo com o objetivo de ganhar acesso às contas das pessoas.

Há cinco tipos comuns de ataques de força bruta: ataques simples, ataques de dicionário, ataques híbridos, ataques reversos e *stuffing* de credenciais. Cada um desses tipos usa uma estratégia diferente para gerar ou obter as combinações de credenciais possíveis. Por exemplo, os ataques de dicionário usam uma lista pré-definida de palavras, termos e combinações para adivinhar as senhas das vítimas, enquanto os ataques reversos tentam adivinhar o nome de usuário a partir da senha conhecida (EDMUNDO, 2021).

Os ataques de força bruta podem prejudicar a segurança da informação de várias formas, como: permitir o acesso não autorizado a contas, sistemas, redes ou aplicações; possibilitar o roubo, a alteração ou a destruição de dados; facilitar a propagação de *malware* ou *ransomware*; causar danos à reputação, à confiança ou à credibilidade da vítima; gerar custos de recuperação ou de reparação; entre outros (SANTOS; LAURENCE, 2022).

3.6 Técnicas de Defesa

Técnicas de defesa são métodos, ferramentas ou práticas que visam proteger os sistemas, redes, aplicações e dados de possíveis ataques cibernéticos (PEREIRA, 2022). Segundo

Pacheco (2019), elas podem ser classificadas em três categorias: técnicas preventivas, técnicas de detecção e técnicas de resposta:

- As técnicas preventivas são aquelas que buscam evitar ou reduzir a probabilidade de um ataque ocorrer, por meio de medidas de controle de acesso, criptografia, *firewall*, antivírus, entre outras;
- As técnicas de detecção são aquelas que buscam identificar ou monitorar a ocorrência de um ataque, por meio de ferramentas de análise de tráfego, auditoria, *logs*, sistemas de detecção de intrusão, entre outras;
- As técnicas de resposta são aquelas que buscam reagir ou mitigar os efeitos de um ataque, por meio de ferramentas de recuperação de dados, isolamento de sistemas, bloqueio de conexões, sistemas de prevenção de intrusão, entre outras.

As técnicas de defesa devem ser usadas de forma integrada e contínua, seguindo uma política de segurança da informação que defina os objetivos, as responsabilidades e as ações de cada parte envolvida (PEREIRA, 2022). Elas devem ser usadas sempre que houver riscos de ataques cibernéticos, que podem afetar a confidencialidade, a integridade ou a disponibilidade das informações (PACHECO, 2019).

As técnicas de defesa são importantes porque elas podem trazer benefícios como: preservar a reputação e a credibilidade da organização, proteger os dados sensíveis e os ativos críticos, garantir a continuidade dos serviços e das operações, evitar prejuízos financeiros ou legais, cumprir as normas e as leis de proteção de dados, entre outros (MARTINS, 2022).

As empresas estão cada vez mais conscientes da necessidade de investir em técnicas de defesa, diante do aumento e da complexidade dos ataques cibernéticos. Algumas das ações que as empresas estão fazendo em relação a este cenário são: contratar profissionais qualificados em segurança da informação, capacitar os funcionários sobre as boas práticas de segurança, adotar soluções de segurança em nuvem, implementar a Lei Geral de Proteção de Dados (LGPD), realizar testes de penetração e auditorias de segurança, entre outras (MARTINS, 2022).

A seguir, é apresentado um quadro com alguns exemplos de técnicas de defesa para cada tipo de ataque cibernético que foi explanado na seção anterior:

Tabela 1 – Exemplificação de ameaças e defesas.

Tipo de ataque	Técnica de defesa
Ataques de negação de serviço *(DoS)*	Usar ferramentas de mitigação de *DDoS*, como o *Cloudflare.*
Ataques de injeção	Validar e filtrar os dados de entrada, usar consultas parametrizadas e escapar dos caracteres especiais.
Ataques de sequestro de sessão	Usar *cookies* seguros e criptografados, gerar *IDs* de sessão aleatórios e complexos e implementar mecanismos de *logout* e expiração de sessão.
Ataques de *cross-site scripting (XSS)*	Escapar, validar e filtrar os dados de saída, usar cabeçalhos de segurança *HTTP* e políticas de segurança de conteúdo e evitar o uso de código dinâmico inseguro.
Ataques de quebra de criptografia	Usar algoritmos e protocolos de criptografia fortes e atualizados, gerar senhas e chaves longas e aleatórias e armazenar os dados criptografados de forma. Além disso, usar sistemas de detecção e prevenção de intrusão *(IDS/IPS)* para monitorar e bloquear o tráfego de rede malicioso que possa tentar acessar ou comprometer os dados criptografados. Os sistemas *IDS/IPS* podem identificar e sinalizar ataques de força bruta, *phishing*, *malware* e outras ameaças cibernéticas que possam explorar as vulnerabilidades da criptografia.

Fonte: Augusto (2022) – adaptado pelo autor

A escolha entre usar um *firewall*, um *IDS*, um *IPS* ou uma combinação deles depende dos objetivos e requisitos de segurança de cada rede. Em geral, recomenda-se usar um *firewall* na borda da rede para filtrar o tráfego indesejado, um *IDS* dentro da rede para monitorar as atividades e gerar alertas, e um *IPS* em pontos críticos da rede para bloquear os ataques mais perigosos (FARES, 2021).

No cenário atual da tecnologia da informação, *firewalls*, *IDS* e *IPS* são tecnologias essenciais para garantir a segurança das redes, mas não são suficientes por si só. Eles devem ser integrados com outras soluções, como antivírus, *antispam*, *Virtual Private Network (VPN)*, criptografia, autenticação, *backup*, entre outras, para formar uma defesa em profundidade contra as ameaças cibernéticas (VAZ; RIZZETTI; FILHO, 2021).

Na Tabela 2 é explanado as diferenças entre *Firewalls*, *IDS* e *IPS*:

Tabela 2 – Diferenças entre *Firewalls*, *IDS* e *IPS*.

Tipo de ataque	Técnica de defesa	*IDS*	*IPS*
Função	Bloquear e filtrar o tráfego de rede com base em regras de segurança.	Monitorar e analisar o tráfego de rede para detectar intrusões.	Detectar e bloquear o tráfego de rede malicioso.
Posição na rede	Na borda da rede, entre o mundo externo e a rede interna.	Dentro da rede, em pontos estratégicos de monitoramento.	Na borda da rede ou dentro da rede, em pontos críticos de proteção.
Impacto no desempenho da rede	Pode causar atrasos ou perdas de pacotes se as regras forem muito restritivas ou complexas.	Não interfere no fluxo do tráfego, mas pode gerar muitos falsos positivos ou negativos se as assinaturas não forem atualizadas ou configuradas corretamente.	Pode causar atrasos ou perdas de pacotes se as assinaturas não forem atualizadas ou configuradas corretamente, ou se o sistema não for capaz de lidar com o volume de tráfego
Vantagens	Protege a rede de ataques externos ou internos, controla o acesso dos usuários, isola segmentos de rede.	Detecta ataques conhecidos ou desconhecidos, gera evidências forenses, auxilia na auditoria e conformidade de segurança.	Impede que os ataques atinjam os alvos, reduz o dano potencial, reage rapidamente às ameaças.
Desvantagens	Não detecta ataques que passam pelas regras, não analisa o conteúdo dos pacotes, pode ser contornado por técnicas de evasão.	Não bloqueia os ataques, requer intervenção humana ou de outro sistema para responder aos alertas, pode sobrecarregar os administradores de segurança.	Pode bloquear tráfego legítimo, requer atualização e configuração constantes, pode ser contornado por técnicas de evasão.

Fonte: Bachinski *et al.* (2020) – adaptado pelo autor

3.7 Métodos de Detecção: Assinatura e Anomalias

Os métodos de detecção de assinatura e anomalia são duas formas de identificar possíveis intrusões em uma rede ou sistema (PAES, 2023). Eles se baseiam em diferentes critérios para determinar se uma atividade é suspeita ou não.

Assis *et al.* (2021) menciona que o método de detecção de anomalia analisa o tráfego de rede com base em um modelo de comportamento normal, que é construído a partir de dados históricos ou estatísticos. Se o tráfego desvia significativamente do modelo, o sistema gera um alerta. Esse método é capaz de detectar ataques desconhecidos ou inovadores, mas também pode gerar muitos falsos positivos ou negativos, dependendo da qualidade e atualização do modelo.

Paes (2023), destaca as principais diferenças entre os métodos de detecção de assinatura e anomalia, são elas:

Tabela 3 – Principais diferenças entre os métodos de detecção de assinatura e anomalia

Critério	Assinatura	Anomalia
Base de comparação	Conjunto de assinaturas ou regras	Modelo de comportamento normal
Tipo de ataque detectado	Ataques conhecidos ou catalogados	Ataques desconhecidos ou inovadores
Vantagens	Baixa taxa de falsos positivos ou negativos, fácil implementação e manutenção	Alta capacidade de adaptação e aprendizado, detecção de novas ameaças
Desvantagens	Não detecta ataques novos ou variantes, requer atualização constante das assinaturas, pode ser contornado por técnicas de evasão	Alta taxa de falsos positivos ou negativos, difícil implementação e manutenção, requer grande volume de dados para construir o modelo

Fonte: Paes (2023) – adaptado pelo autor

Suponha que um atacante esteja tentando explorar uma vulnerabilidade em um servidor Web usando um pacote malformado. Se o sistema de detecção usar o método de assinatura, ele vai comparar o pacote com as assinaturas que ele possui e se ele encontrar uma correspondência, ele vai gerar um alerta. Porém, se o atacante modificar o pacote de alguma forma, ele pode evitar a detecção.

Se o sistema de detecção usar o método de anomalia, ele vai analisar o pacote com base no modelo de comportamento normal que ele possui e se ele detectar uma anomalia, ele vai

gerar um alerta. Nesse caso, o atacante não pode escapar da detecção simplesmente alterando o pacote, pois o sistema vai perceber que ele é diferente do esperado.

3.8 *SNORT*

SNORT é um *software* livre de detecção e prevenção de intrusão em rede, que pode analisar e registrar o tráfego de rede em tempo real, gerando alertas ou bloqueando pacotes suspeitos. Ele foi desenvolvido inicialmente por Martin Roesch em 1998 e atualmente é mantido pela *Cisco* (WALEED; JAMALI; MASOOD, 2022).

O *SNORT* funciona com base em regras que definem os padrões de ataques conhecidos ou anômalos (BADOTRA; PANDA, 2021). Essas regras podem ser obtidas gratuitamente pela comunidade ou assinadas pela *Cisco*, que oferece as regras mais atualizadas e testadas (ERLACHER; DRESSLER, 2020). Ele também permite que os usuários criem ou modifiquem suas próprias regras de acordo com as necessidades e características de cada rede (JAIN; ANUBHA, 2021).

O *SNORT* pode ser instalado e configurado em diferentes plataformas, como *Linux*, *Windows*, *MacOS*, entre outras (WALEED; JAMALI; MASOOD, 2022). Ele pode ser usado como um simples analisador de pacotes, como um sistema de detecção de intrusão que apenas gera alertas, ou como um sistema de prevenção de intrusão que também bloqueia o tráfego malicioso (BADOTRA; PANDA, 2021).

Erlacher e Dressler (2020), aponta vantagens e desvantagens em utilizar o *SNORT*, como:

As principais vantagens são:

- É um *software* livre e de código aberto que pode ser adaptado e personalizado pelos usuários;
- É um *software* leve e eficiente que pode processar grandes volumes de tráfego sem afetar o desempenho da rede;
- É um *software* versátil e flexível que pode ser usado para diferentes propósitos e cenários de segurança;
- É um *software* integrado e compatível que pode se comunicar e interagir com outros sistemas e ferramentas de segurança.

As principais desvantagens são:

- É um *software* complexo e técnico que requer conhecimento e experiência para instalar, configurar e gerenciar;
- É um *software* dependente e limitado que requer atualização e manutenção constantes das regras para detectar novas ameaças;
- É um *software* vulnerável e passível que pode ser contornado ou comprometido por técnicas de evasão ou ataques direcionados.

O *SNORT* é utilizado por diversas organizações e indivíduos que buscam proteger suas redes de ataques cibernéticos, como empresas, governos, universidades, entre outros. Também é utilizado para fins educacionais e de pesquisa, como em cursos, laboratórios, projetos, entre outros (WALEED; JAMALI; MASOOD, 2022).

3.9 *SNMP*

Contar com ferramentas que permitam o monitoramento e gerenciamento eficientes é essencial no vasto ecossistema das redes de computadores, onde muitos dispositivos estão interconectados e formam uma teia complexa de comunicação. O Protocolo *SNMP* é fundamental neste contexto, pois facilita a supervisão e administração de redes remotas de forma uniforme.

O *SNMP* é um protocolo de gerenciamento de redes, concebido para facilitar a troca de informações entre dispositivos de rede e uma estação de gerenciamento. Ele opera de forma distinta de outros protocolos de comunicação, como o *TCP/IP*, focando-se exclusivamente na administração e monitoramento da infraestrutura de rede (ALVARENGA; RAMOS, 2011).

De acordo com Domingos *et al.* (2005), o *SNMP* possui três componentes fundamentais para o seu funcionamento, são eles:

- Agentes *SNMP*: Presentes em dispositivos de rede, esses agentes são responsáveis por coletar e armazenar informações locais sobre o dispositivo, como uso de *Central Processing Unit (CPU)*, tráfego de rede, status de portas, entre outros.
- Gerentes *SNMP*: São as estações de gerenciamento que se comunicam com os agentes *SNMP*. Elas requisitam informações dos agentes, definem configurações e enviam comandos para os dispositivos de rede.
- *Management Information Base (MIB)*: A *MIB* é uma base de dados hierárquica que organiza as informações coletadas pelos agentes *SNMP*. Ela utiliza uma estrutura de árvore,

onde cada nó representa um objeto gerenciável, como interfaces de rede, *CPU*, memória, entre outros.

Resumidamente, o *SNMP* tem um papel crucial na administração e monitoramento de redes, oferecendo uma estrutura sólida e uniforme para supervisionar dispositivos de maneira remota. Sua habilidade de coletar informações, configurar parâmetros e notificar sobre eventos críticos faz com que o *SNMP* permaneça como uma ferramenta essencial para administradores de rede em escala global.

3.10 *Zabbix*

O *Zabbix* é uma solução de código livre e sem custos de licenciamento, criado para monitorar e gerenciar dispositivos de rede e seus serviços. Seu sistema de alerta permite enviar notificações por *e-mail* em caso de problemas, além de mensagens de texto. Com um entendimento mais avançado da ferramenta, é viável enviar comandos remotamente, o que facilita a resolução de questões. Geralmente, os dados dos dispositivos são exportados para o *Zabbix* por meio do protocolo *SNMP*. (NUNES, 2018)

Basicamente, o *Zabbix* coleta dados de dispositivos e serviços na rede, como uso de *CPU*, tráfego de rede, memória e status de serviços. Esses dados são processados e exibidos de forma clara em uma interface de usuário intuitiva. Isso possibilita que administradores de sistemas e redes identifiquem problemas, monitorem tendências de desempenho e tomem medidas corretivas, se necessário.

3.11 *Netflow*

O *NetFlow*, inicialmente exclusivo para dispositivos da fabricante *Cisco*, é um protocolo que permite aos administradores acessar informações detalhadas sobre o tráfego *IP* em redes de dados. Roteadores e *switches* coletam dados de fluxo e os enviam para coletores, proporcionando uma análise detalhada e flexível do uso de recursos (RUIZ; SAPIA, 2016).

De acordo com Quittek (2004), o *Internet Protocol Flow Information Export (IPFIX)*, surgiu da necessidade de padronizar a exportação de informações de fluxo *IP* de roteadores e outros dispositivos de rede. Para a sua criação foi utilizado como base o protocolo *NetFlow*, devido à sua simplicidade, pois não apresenta mecanismos de transporte, segurança, confiabili-

dade ou redundância. Permitindo concentrar esforços na construção de estruturas com cautela, tornando simples e evitando qualquer sobrecarga.

3.12 *Elasticsearch, Logstash* e *Kibana (ELK)*

É uma plataforma poderosa que oferece uma variedade de recursos para monitoramento e análise de dados em tempo real. Composta por diferentes componentes, como *Elasticsearch, Logstash, Kibana* e *Beats,* ela fornece uma solução abrangente para diversas necessidades de monitoramento, desde registros de aplicativos até métricas de infraestrutura (RöHRS, 2021),
.

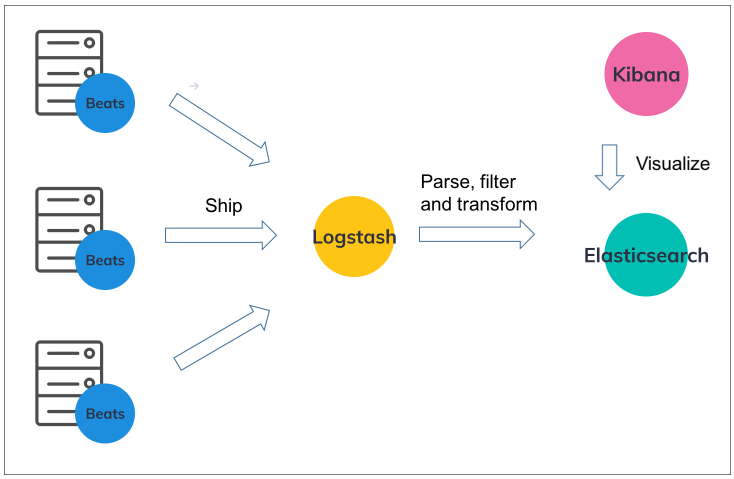
Quando se trata de monitoramento de força bruta, o *ELK* pode ser uma ferramenta valiosa para identificar e mitigar tentativas de ataques cibernéticos, a empresa AWS (2023), cita que a ferramenta pode ser utilizada no seguinte contexto:

- Coleta de Dados: O *ELK* pode ser configurado para coletar *logs* de vários pontos de entrada, como servidores Web, *firewalls* e sistemas de autenticação. Esses *logs* podem incluir informações sobre tentativas de *login*, falhas de autenticação e outros eventos relacionados à segurança.
- Análise de Padrões: Utilizando o *Elasticsearch* como mecanismo de busca e análise, é possível realizar consultas complexas nos dados coletados para identificar padrões suspeitos. Por exemplo, pode-se procurar por tentativas repetidas de *login* com credenciais inválidas vindas de um mesmo endereço *IP*.
- Visualização e Monitoramento em Tempo Real: O *Kibana* oferece recursos avançados de visualização e *dashboard*, permitindo que os administradores de segurança monitorem atividades de força bruta em tempo real. Gráficos e tabelas podem ser criados para exibir métricas importantes, como número de tentativas de *login* por hora ou origens geográficas dos ataques.
- Integração com Outras Ferramentas de Segurança: O *ELK* pode ser integrado a outras ferramentas de segurança, como sistemas de detecção de intrusões *(IDS)* e *firewalls*, para uma defesa mais abrangente contra ataques cibernéticos.

Em resumo, o *ELK* oferece uma plataforma flexível e escalável para o monitoramento de força bruta e outras ameaças de segurança cibernética. Através dos seus componentes, como os *Beats* para coleta, o *Logstash* para filtragem, o *Elasticsearch* para análise e o *Kibana*

para visualização em tempo real, ele capacita as organizações a detectar e reagir prontamente a possíveis ataques, reforçando sua segurança digital, conforme demonstrado na Figura 1:

Figura 1 – Diagrama de composição *ELK*



Fonte: Logz.io, 2023.

3.12.1 Filebeat e Módulo Netflow

O *Filebeat* é um módulo essencial do *ELK* projetado para simplificar e otimizar a coleta de *logs* e dados de diferentes fontes em um ambiente de Tecnologia e Informação (TI). Sua função principal é coletar, enviar e consolidar registros de *logs* e eventos de diversas origens para o *Elasticsearch* ou outros destinos, como o *Logstash*, para posterior análise e visualização.

Este módulo é particularmente útil em ambientes onde há uma grande quantidade de dados gerados continuamente, como em servidores, aplicativos Web, sistemas de monitoramento de segurança e muito mais. Ele funciona como um agente leve que pode ser instalado em diversos *hosts* para coletar e enviar dados de forma eficiente para a análise centralizada.

O *Filebeat* conta com sub-módulos, como o *NetFlow*, que é uma poderosa extensão voltada para a coleta e análise de dados de fluxo de tráfego de rede em tempo real. Esse módulo é especificamente projetado para capturar informações detalhadas sobre o tráfego de rede, incluindo sua origem, destino, protocolos utilizados e volumes de dados transferidos.

Ao utilizar o módulo *NetFlow*, os usuários conseguem monitorar o tráfego de rede de maneira eficiente, identificando padrões de uso, anomalias e potenciais ameaças de segurança. As informações coletadas podem ser facilmente integradas ao *Elasticsearch* para análises avançadas e visualizações através do *Kibana*.

Essa funcionalidade do *Filebeat* simplifica significativamente o processo de coleta e análise de dados de fluxo de tráfego de rede, proporcionando *insights* valiosos para a otimização de desempenho, monitoramento de segurança e resolução de problemas de rede.

4 TRABALHOS RELACIONADOS

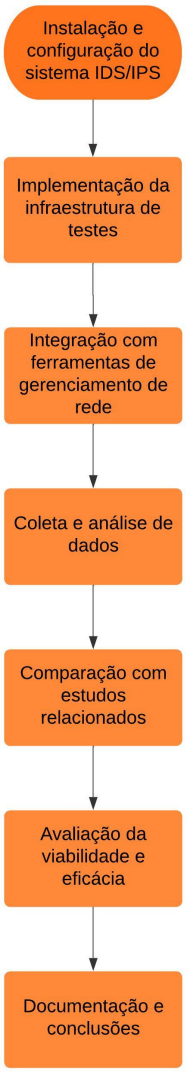
O estudo conduzido por Tessario (2013), aborda o tema de ataques de força bruta direcionados a roteadores *Mikrotik*. Ele chega à conclusão de que o *firewall* proporciona uma sensação falsa de segurança, uma vez que não consegue proteger a rede contra ameaças internas, engenharia social, vírus e *modems* internos, que representam grandes desafios para a segurança. Como direção para futuras pesquisas, o trabalho sugere a implementação do *Snort* como um sistema de prevenção de intrusões, atuando em conjunto com o *firewall* para bloquear o tráfego e reforçar a defesa da rede (TESSARIO, 2013).

O projeto desenvolvido por Mantellis e Fellipe (2018), envolve a utilização da ferramenta *Snort* em conjunto com o *PFsense*. Esses *softwares* foram empregados para analisar o tráfego de pacotes, examinar seu comportamento e fornecer dados para que o sistema possa identificar características maliciosas nos dados. Após configurar a infraestrutura e realizar simulações de ataques, Mantellis e Fellipe (2018), concluíram que o *Snort* demonstra um alto potencial para detectar e bloquear tanto ataques passivos quanto ativos.

5 MATERIAIS E MÉTODOS

A fim de viabilizar a execução do ataque e, em seguida, a identificação ou prevenção do mesmo, bem como a análise dos dados do fluxo de rede durante sua ocorrência, foram seguidos alguns procedimentos, conforme ilustrado no diagrama de fluxo apresentado na Figura 2 :

Figura 2 – Diagrama de fluxo metodologia



Fonte: AUTOR, 2024.

5.1 Definição e configuração das ferramentas

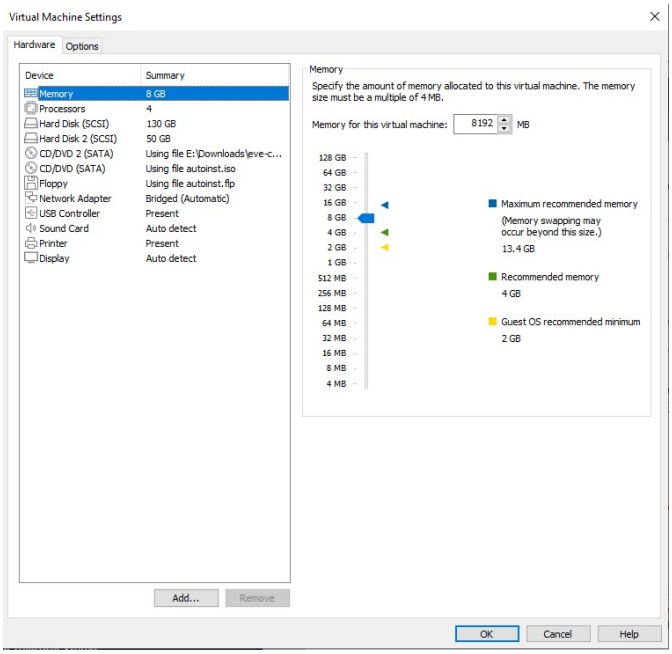
As ferramentas que foram empregadas para realização do laboratório foram as seguintes:

- *VMware Workstation 17 player*, versão 17.5.0 (instalada em *desktop* físico *Intel(R) Core(TM) i5-9400F*);
- *VirtualBox*, versão 7.0 (instalada em *desktop* físico *Ryzen 5 3600*);
- *Emulated Virtual Environment Next Generation (Eve-NG), versão 5.0.1-24* (instalada como máquina virtual no *software VMware Workstation*);
- *PFsense*, versão 2.7.2 (Solução virtualizada no *software Eve-NG*);
- *Snort*, versão 3.1.7.3.0 (*software* instalado como pacote adicional no *software PFsense*);
- Roteador *Mikrotik*, versão 7.10 (solução virtualizada no *software Eve-NG*);
- *Kali Linux Large.3-amd64* versão 6.6.9, juntamente com a solução *Hydra*, versão 9.5, ferramenta para a realização do ataque de força bruta (solução virtualizada no *software VirtualBox*).
- *Ubuntu Server 22.04.4 LTS* (sistema operacional virtualizado no *software VirtualBox*), ferramenta para instalação do monitoramento *Netflow* através do *ELK: ElasticSearch* versão 7.16.2, *Kibana* versão 7.16.2 e *Filebeat* 7.16.2 (módulo *netflow*).
- *Ubuntu Server 22.04.4 LTS* (sistema operacional virtualizado no *software VMWare*), ferramenta para instalação do monitoramento de recursos e trafégo de rede através do *Zabbix* versão 6.2.9.

5.1.1 Máquinas Virtuais

Conforme apresentado anteriormente, o trabalho foi conduzido utilizando uma máquina virtual hospedada no *software VMware Workstation 17 Player*, a qual foi instalada em um computador físico *Intel*, processador i5-9400F. Para a implementação do laboratório proposto, criou-se uma máquina virtual denominada ”*EVE-NG*”, que serviu como ambiente de teste e laboratório. Na Figura 3 estão as configurações utilizadas para a criação desta máquina.

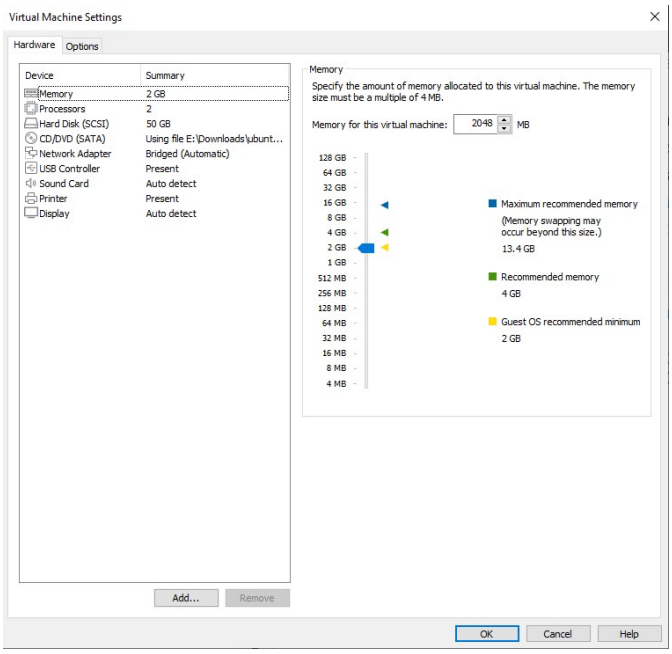
Figura 3 – Configuração máquina virtual *EVE-NG*



Fonte: AUTOR, 2024.

Na Figura 4 estão expostas as configurações utilizadas para a criação da máquina virtual *Ubuntu Server*, na qual foi instalado o software *Zabbix* para monitoramento de recursos e trafégo de rede nos testes a serem realizados.

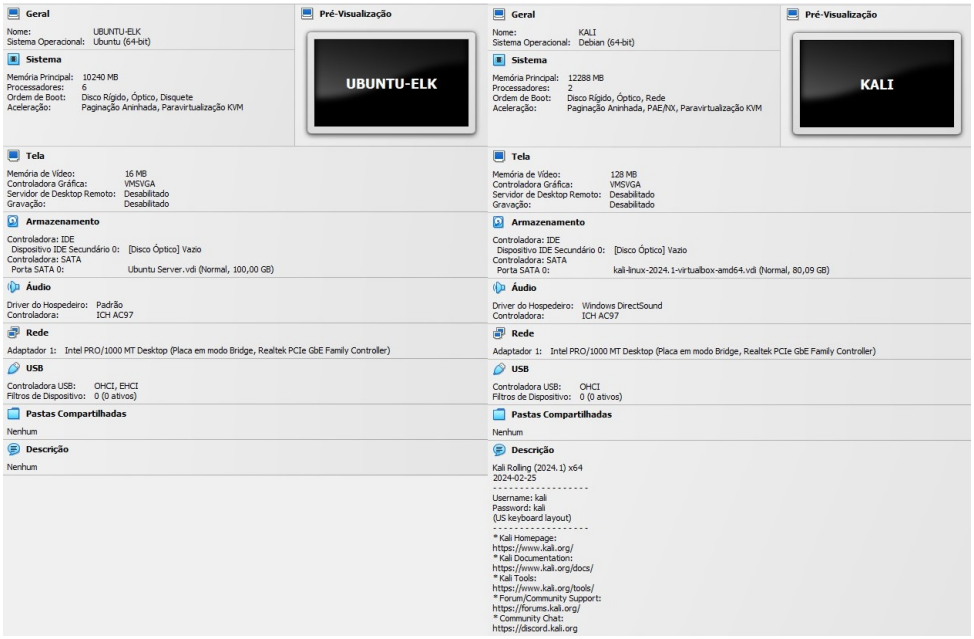
Figura 4 – Configuração máquina virtual *Zabbix*



Fonte: AUTOR, 2024.

Foi instalado na segunda máquina física, equipada com um processador *AMD Ryzen 5 3600* o *software VirtualBox*, com a ferramenta foi possível realizar a criação de duas máquinas virtuais, denominadas *Ubuntu-ELK* e *Kali*, conforme Figura 5:

Figura 5 – Configuração máquina virtual *UBUNTU-ELK* e *Kali*



Fonte: AUTOR, 2024.

5.1.2 EVE-NG

O software *EVE-NG*, foi devidamente instalado conforme mencionado anteriormente.

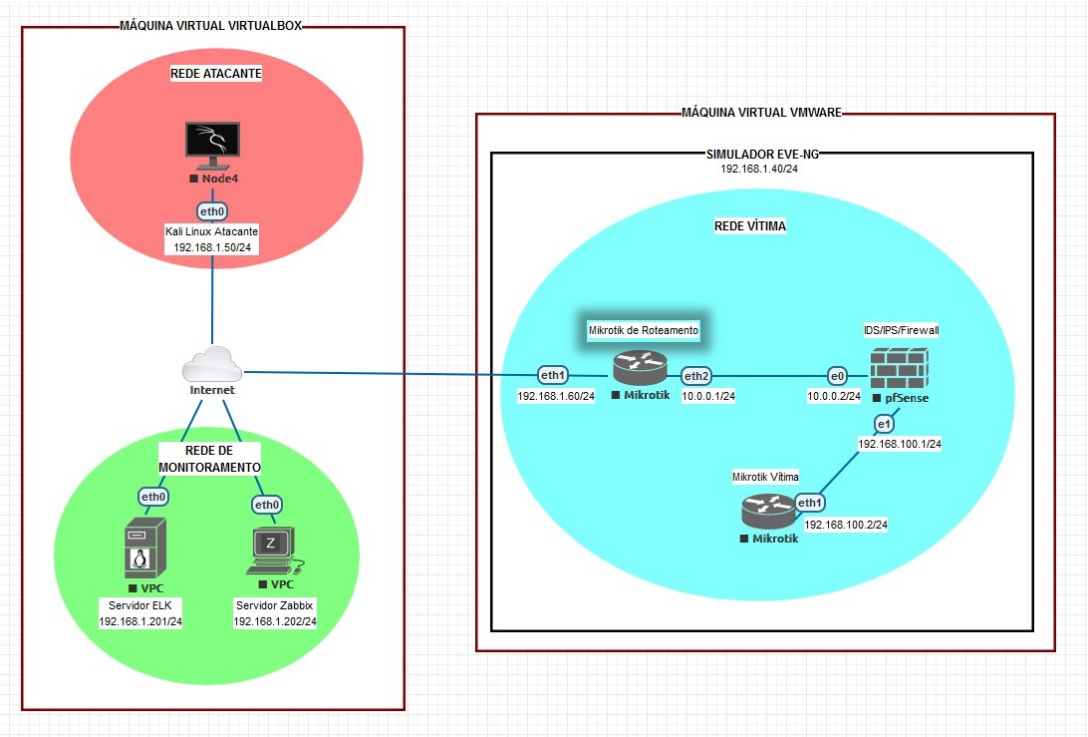
Trata-se de uma plataforma utilizada para emular redes virtuais, criando ambientes de teste e aprendizado no campo das redes de computadores. Com o auxílio do *EVE-ng*, é possível simular diferentes dispositivos de rede, como roteadores, *switches* e *firewalls*, em um ambiente virtual, o que permite que profissionais de redes e estudantes experimentem diversas configurações e cenários sem necessidade de recursos físicos.

A plataforma do *Eve-NG* oferece interface gráfica amigável, facilita o processo de criação e gerenciamento dos laboratórios de rede, tornando-o mais acessível e intuitivo.

Ademais, a plataforma suporta uma ampla variedade de dispositivos e sistemas operacionais de rede, o que a torna uma ferramenta versátil e valiosa para simular cenários reais

em um ambiente virtual controlado e seguro. A Figura 6 demonstra o laboratório desenvolvido para os testes.

Figura 6 – Laboratório para realização dos testes de ataque de força bruta - *Eve-NG*



Fonte: AUTOR, 2024.

Os testes foram conduzidos com a distribuição dos equipamentos da seguinte maneira: a máquina que possuía a instalação do *Kali Linux*, responsável pela execução do ataque, recebeu o *IP* 192.168.1.50/24. A interconexão entre a rede interna do atacante e a rede interna da vítima ocorreu através da Internet.

A rede interna da vítima é composta por três *hosts*: um *Mikrotik* para roteamento e interconexão das redes atacante e vítima, com os *IPs* 192.168.1.60/24 atríbuido a sua *eth1* (interface que comunica com a Internet) e 10.0.0.1/24 a sua *eth2* (interface que realiza a conexão com o *host PFSense*), o *Firewall PFsense*, configurado com o pacote do *Snort* funcionando como um *IDS/IPS*, recebeu o *IP* 10.0.0.2/24 em sua interface *WAN* e o *IP* 192.168.100.1/24 em sua interface *LAN*. Conectado diretamente ao *Pfsense*, encontra-se outro roteador *Mikrotik* com o *IP* 192.168.100.2/24, que foi a vítima no laboratório proposto.

Na Figura 6 é possível observar outros dois *hosts*: Servidor *ELK* com o *IP* 192.168.1.201/24 atribuído a sua interface e o servidor *Zabbix* com o *IP* 192.168.1.202/24, todos os dois foram fundamentais para realização do monitoramento durante os testes.

5.1.3 PFSense

O software *Eve-NG* foi utilizado para emular um dispositivo *firewall PFsense*, na versão 2.7.2. O *PFsense* é uma plataforma de *software* de código aberto projetada para gerenciar *firewalls*, roteadores e pontos de acesso. Baseado no sistema operacional *Free Berkeley Software Distribution (FreeBSD)*, oferece uma solução robusta para criar redes seguras e eficientes. Com recursos avançados, como filtragem de tráfego, balanceamento de carga e rede *VPN*, o *PFsense* é amplamente utilizado em ambientes corporativos e domésticos para reforçar a segurança e otimizar o desempenho das redes. Sua interface gráfica simplifica a configuração e monitoramento, sendo uma escolha popular entre profissionais de redes. Além disso, o *PFsense* permite a instalação de pacotes adicionais, como no respectivo laboratório, onde foi instalado o pacote *Snort* em sua versão gratuita de comunidade.

Após a criação das regras foi instalado no *PFsense* o pacote *Snort*, segue o guia de instalação no Anexo D, contido ao fim do documento.

5.1.4 Configuração serviço Snort no PFSense

Após a instalação do pacote *Snort* no *PFsense* e a configuração prévia na interface *WAN*, conforme descrito no Anexo D, foi criada uma regra personalizada para bloquear ataques de força bruta direcionados ao serviço *SSH*. A regra monitora o tráfego *TCP* direcionado à porta 22 do $HOME_NET e verifica se há várias tentativas de login *SSH* dentro de um curto período. Se um endereço *IP* fizer múltiplas tentativas em pouco tempo, a regra gera um alerta indicando uma possível tentativa de ataque de força bruta *SSH*. Abaixo segue a regra inserida no serviço *Snort*:

Listing 5.1 – Regra de bloqueio força bruta serviços SSH - Snort

```
1 alert tcp any any -> $HOME_NET 22 (msg:"Tentativa de ataque de força bruta
    SSH"; flow:to_server, established, no_stream; content:"SSH-"; depth:4;
    detection_filter:track by_src, count 5, seconds 60; metadata:policy max-
    detect-ips drop, service ssh; classtype:misc-activity; sid:19559; rev
    :15;)
```

A regra pode ser intrerpretada da seguinte forma:

- *alert tcp any any -> HOME_NET 22*: especifica que a regra deve alertar sempre que for detectado tráfego *TCP* destinado à porta 22 (porta padrão do *SSH*) no endereço *IP* do *$HOME_NET* (geralmente a rede interna).
- (msg: ”Tentativa de ataque de força bruta *SSH*”;): fornece uma mensagem descritiva que será incluída nos alertas gerados pelo *Snort*. Neste caso, a mensagem indica que uma tentativa de ataque de força bruta *SSH* foi detectada.
- *flow:to_server, established, no_stream;*: especifica as condições de fluxo sob as quais a regra será acionada. Indicando que o tráfego deve ser direcionado para o servidor (*to_server*), a conexão *TCP* deve estar estabelecida (*established*) e não deve haver fluxo de dados (*no_stream*).
- *content:”SSH-”; depth:4;*: a regra procura pelo padrão ”*SSH-*”no *payload* do pacote *TCP* até uma profundidade de 4 *bytes* a partir do início do *payload*. Normalmente é usado para identificar conexões *SSH*, já que geralmente os clientes *SSH* enviam essa *string* no início da conexão.
- *detection_filter:track by_src, count 5, seconds 60;*: especifica um filtro de detecção que rastreia o número de vezes que a regra foi acionada por fonte *IP*. Se um *IP* acionar a regra cinco ou mais vezes em um período de 60 segundos, isso será registrado.
- *metadata:policy max-detect-ips drop, service ssh;*: são fornecidos metadados adicionais. *policy max-detect-ips drop* indica que, se o limite de detecções por *IP* for atingido, o tráfego desse IP pode ser descartado. *service ssh* Indica que a regra se aplica ao serviço *SSH*.
- *classtype:misc-activity;*: atribui um tipo de classe à regra. Neste caso, é classificada como uma atividade miscelânea.
- *sid:19559; rev:15;*: apresentam o *ID* e a revisão da regra, que são usados para identificar exclusivamente a regra e acompanhar as alterações feitas nela ao longo do tempo.

5.1.5 Roteador Mikrotik Vítima

O alvo selecionado para o ataque de força bruta foi o serviço de servidor *SSH* do roteador *MikroTik* vítima. Os roteadores *MikroTik* são dispositivos de rede avançados que operam com o sistema operacional *RouterOS*, desenvolvido pela empresa *MikroTik*. A escolha desse dispositivo como alvo foi motivada pelo seu amplo uso em ambientes corporativos e por

provedores de serviços de Internet, sendo reconhecidos pela sua versatilidade e desempenho. Abaixo segue a configuração realizada no roteador *Mikrotik*:

Listing 5.2 – CLI - Command Line Interface Mikrotik Vítima

```
# 2024-04-22 22:17:16 by RouterOS 7.10
# software id =
#
/interface ethernet
set [ find default-name=ether1 ] comment=PFSENSE
set [ find default-name=ether2 ] disabled=yes
set [ find default-name=ether3 ] disabled=yes
set [ find default-name=ether4 ] disabled=yes
/interface wireless security-profiles
set [ find default=yes ] supplicant-identity=MikroTik
/port
set 0 name=serial0
/snmp community
set [ find default=yes ] disabled=yes
add addresses=::/0 name=labBruteforce write-access=yes
/ip address
add address=192.168.100.2/24 interface=ether1 network=192.168.100.0
/ip dhcp-client
add interface=ether1
/ip dns
set servers=192.168.100.1,8.8.8.8,8.8.4.4
/ip firewall nat
add action=masquerade chain=srcnat
/ip route
add dst-address=0.0.0.0/0 gateway=192.168.100.1
/ip ssh
set always-allow-password-login=yes forwarding-enabled=remote
/ip traffic-flow
set enabled=yes
/ip traffic-flow target
add dst-address=192.168.1.201
/snmp
set enabled=yes trap-community=labBruteforce
/system identity
```

```
35 set name=RB-VITIMA
36 /system note
37 set show-at-login=no
```

O roteador *Mikrotik* foi configurado para ser acessado via comando *SSH*, usando a porta porta padrão 22. Além dessa configuração, o *firewall* do *Mikrotik* foi ajustado para interceptar ataques. Este roteador possibilita a adição de regras de *firewall*, contribuindo para a detecção de ataques de força bruta.

Com a implementação das configurações citadas e a subsequente realização dos ataques, o próprio roteador intercepta a investida e adiciona o *IP* do *host* atacante a uma *blacklist*, que consiste em uma lista de *IPs* proibidos de trocarem informações com o roteador. Especificamente as regras de *firewall* configuram uma série de medidas de segurança para proteger contra tentativas de força bruta no serviço *SSH*. A primeira regra identifica e bloqueia endereços *IP* listados na ”ssh_blacklist”ao detectar tentativas de acesso à porta 22. Além disso, há um sistema de etapas que monitora novas conexões *SSH*: *IPs* suspeitos são primeiro adicionados à lista ”ssh_stage1”e, se persistirem, progridem para ”ssh_stage2”e ”ssh_stage3”. Cada etapa aumenta o tempo de bloqueio, com o estágio final resultando em um bloqueio de 10 dias para os *IPs* mais persistentes.

Segue abaixo a inclusão das regras de *firewall* no terminal do roteador:

Listing 5.3 – CLI - Command Line Interface Mikrotik Vítima firewall

```
1 in /ip firewall filter
2
3 add chain=input protocol=tcp dst-port=21 src-address-list=ftp_blacklist
     action=drop \
4 comment="drop ftp brute forcers"
5
6 add chain=output action=accept protocol=tcp content="530 Login incorrect"
     dst-limit=1/1m,9,dst-address/1m
7
8 add chain=output action=add-dst-to-address-list protocol=tcp content="530
     Login incorrect" \
9 address-list=ftp_blacklist address-list-timeout=3
10
11 in /ip firewall filter
12
```

```
13 add chain=input protocol=tcp dst-port=22 src-address-list=ssh_blacklist
       action=drop \
14 comment="drop ssh brute forcers" disabled=no
15
16 add chain=input protocol=tcp dst-port=22 connection-state=new \
17 src-address-list=ssh_stage3 action=add-src-to-address-list address-list=
       ssh_blacklist \
18 address-list-timeout=10d comment="" disabled=no
19
20 add chain=input protocol=tcp dst-port=22 connection-state=new \
21 src-address-list=ssh_stage2 action=add-src-to-address-list address-list=
       ssh_stage3 \
22 address-list-timeout=1m comment="" disabled=no
23
24 add chain=input protocol=tcp dst-port=22 connection-state=new src-address-
       list=ssh_stage1 \
25 action=add-src-to-address-list address-list=ssh_stage2 address-list-timeout
       =1m comment="" disabled=no
26
27 add chain=input protocol=tcp dst-port=22 connection-state=new action=add-
       src-to-address-list \
28 address-list=ssh_stage1 address-list-timeout=1m comment="" disabled=no
```

5.1.6 Hydra - Kali Linux

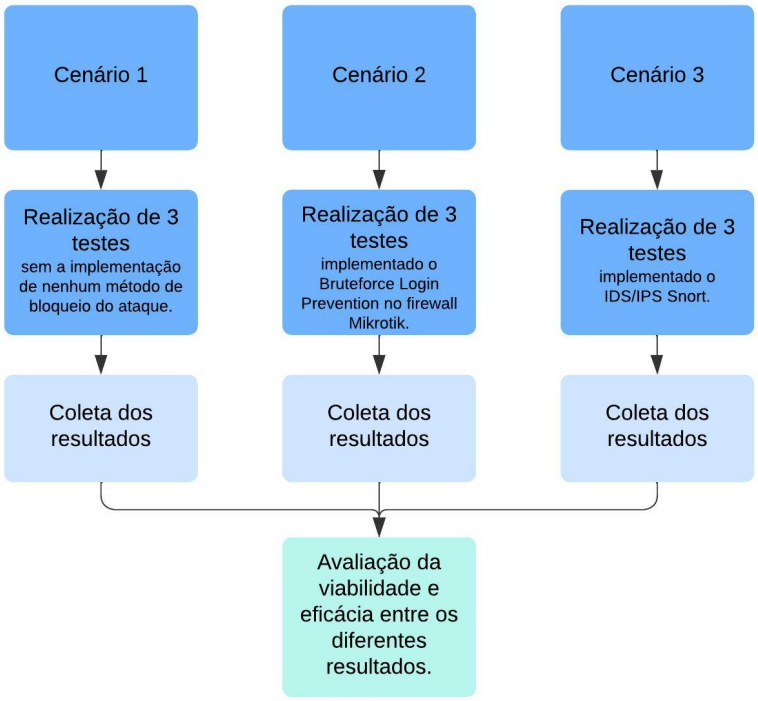
O *Hydra*, no *Kali Linux*, é uma ferramenta de *hacking* poderosa, focada em ataques de força bruta e tentativas de *login* automáticas em serviços *online*. Com suporte a diversos protocolos, como *SSH*, *FTP* e *HTTP*, ele permite a realização de testes de penetração para identificar vulnerabilidades de autenticação em sistemas.

Sua flexibilidade se destaca na personalização, possibilitando a especificação de listas de senhas, usuários e outros parâmetros. Essa adaptabilidade torna o *Hydra* uma escolha valiosa para a realização do ataque a seguir.

5.2 Realização do Ataque

Para a realização dos testes, foram executadas simulações de ataques utilizando a ferramenta *Hydra*. Para essa finalidade, foi adquirido um documento online contendo uma compilação das principais senhas utilizadas por usuários nos últimos anos. Esse documento inclui uma lista com 3 mil senhas, que foram testadas sequencialmente pela ferramenta ao longo de períodos de 10 minutos em cada teste. Os testes foram divididos em 3 cenários conforme descrito na Figura 7:

Figura 7 – Fluxograma divisão dos cenários



Fonte: AUTOR, 2024.

Com o objetivo de garantir a consistência dos resultados, foram conduzidos três testes preliminares sem a aplicação de bloqueios de *firewall* ou sistemas de *(IDS/IPS)*. Em seguida, o *firewall* do *Mikrotik* foi ativado isoladamente em uma etapa intermediária, seguido pela ativação exclusiva do sistema *IDS/IPS Snort*.

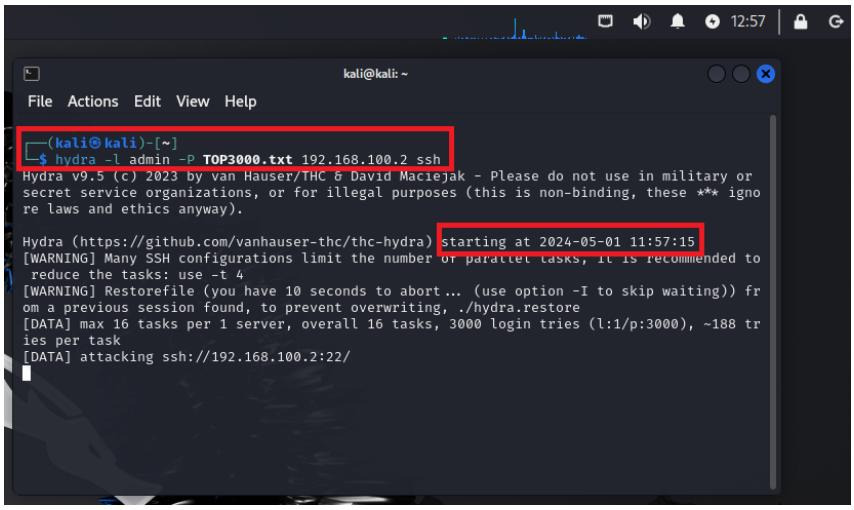
Após a conclusão de todos os testes, os resultados foram coletados e comparados. O propósito foi analisar os diferentes comportamentos e determinar a eficácia relativa em termos de segurança, consumo de recursos de máquina e eficiência. Para a coleta e análise dos dados, foram empregados os *softwares* previamente mencionados: *Zabbix* e *ELK*.

5.2.1 Primeiro cenário

No primeiro cenário foram realizados três testes, partindo da máquina atacante *Kali Linux* pela ferramenta *Hydra*, sem nenhuma fonte de segurança ou bloqueio configurada.

A Figura 8, apresenta o exato momento do primeiro ataque:

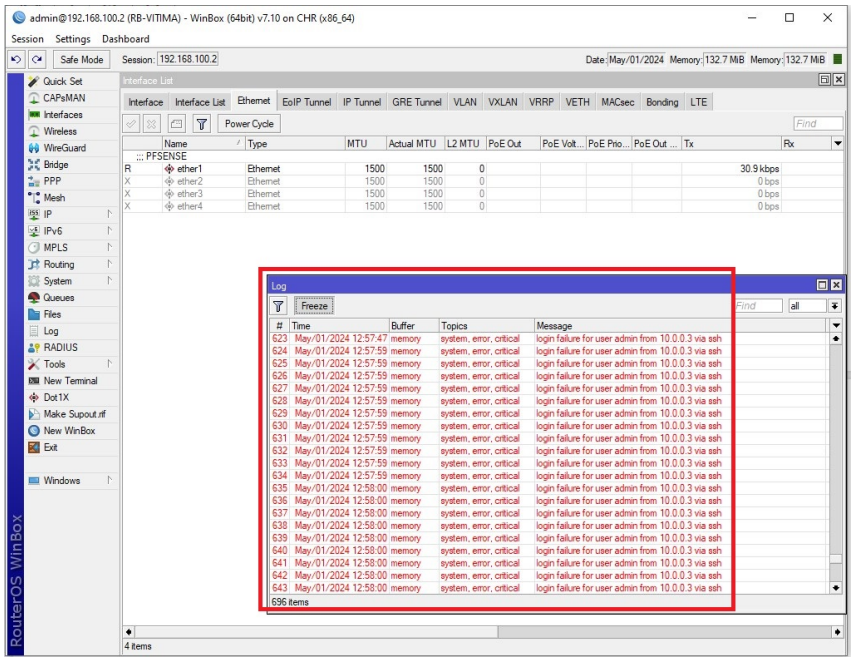
Figura 8 – Terminal do *Kali Linux* - Exibição do primeiro ataque de força bruta - Cenário 1



Fonte: AUTOR, 2024.

A Figura 9 retrata o exemplo de um momento de tentativas de acesso via *SSH* falhos no console de *logs* do roteador *Mikrotik* da vítima.

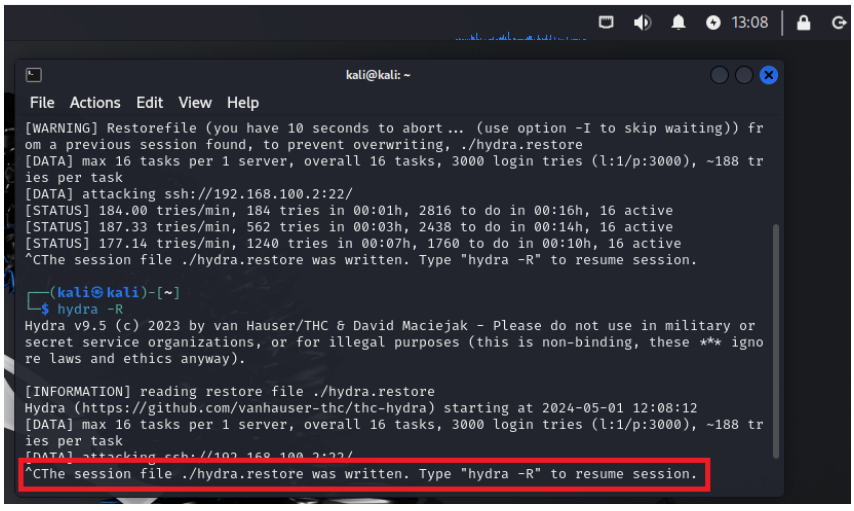
Figura 9 – *Logs* do console *Miktrotik* durante o primeiro ataque de força bruta - Cenário 1



Fonte: AUTOR, 2024.

Na imagem, é visível diversos registros de tentativas de acesso sem sucesso. Esse comportamento persiste até a interrupção manual do ataque, como mostrado na Figura 10:

Figura 10 – Terminal do *Kali Linux* - Exibição do fim do primeiro ataque - Cenário 1

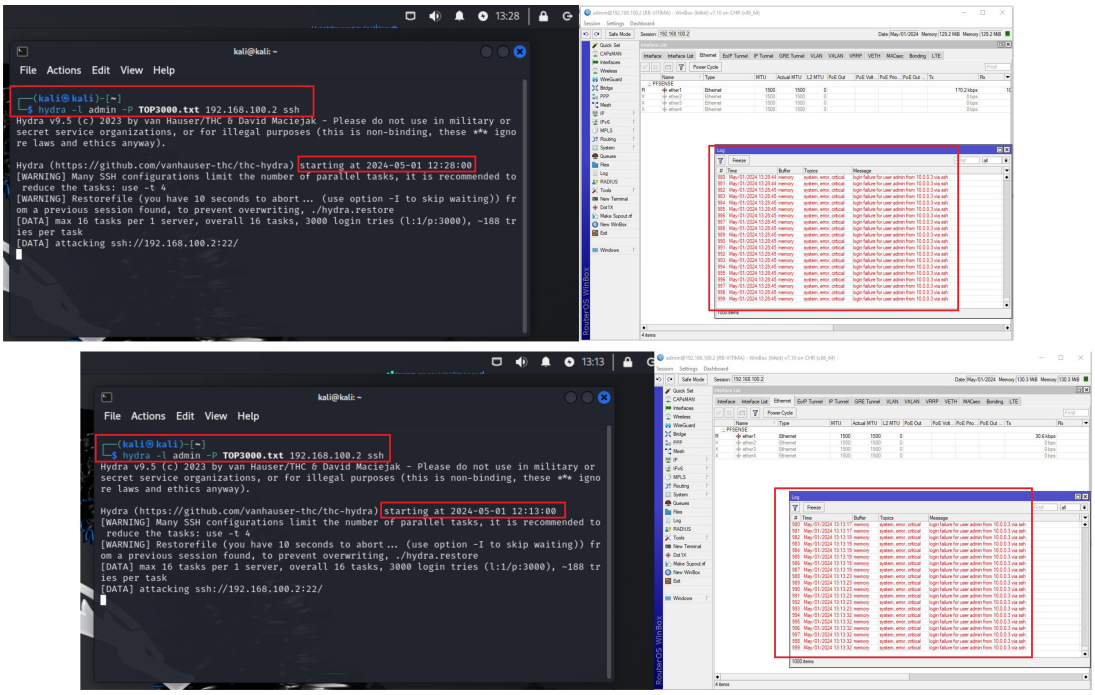


Fonte: AUTOR, 2024.

Após o término do primeiro ataque, foi feito um intervalo de 5 minutos antes de iniciar o segundo e, em seguida, o terceiro ataque, ambos com duração de 10 minutos. A Figura 11 mostra os comandos utilizados nos ataques e a tela de *logs* do *Mikrotik*:

segundo e terceiro ataques:

Figura 11 – Terminal do *Kali Linux* e *console Mikrotik* - Exibição do segundo e terceiro ataques - Cenário 1



Fonte: AUTOR, 2024.

5.2.2 Segundo cenário

No segundo cenário, da mesma forma que no primeiro, foram realizados três testes. No entanto, no roteador *Mikrotik*, que foi a vítima, foram configuradas regras em sua aba de filtros para bloquear qualquer tentativa de ataque *SSH*. Conforme Figura 12:

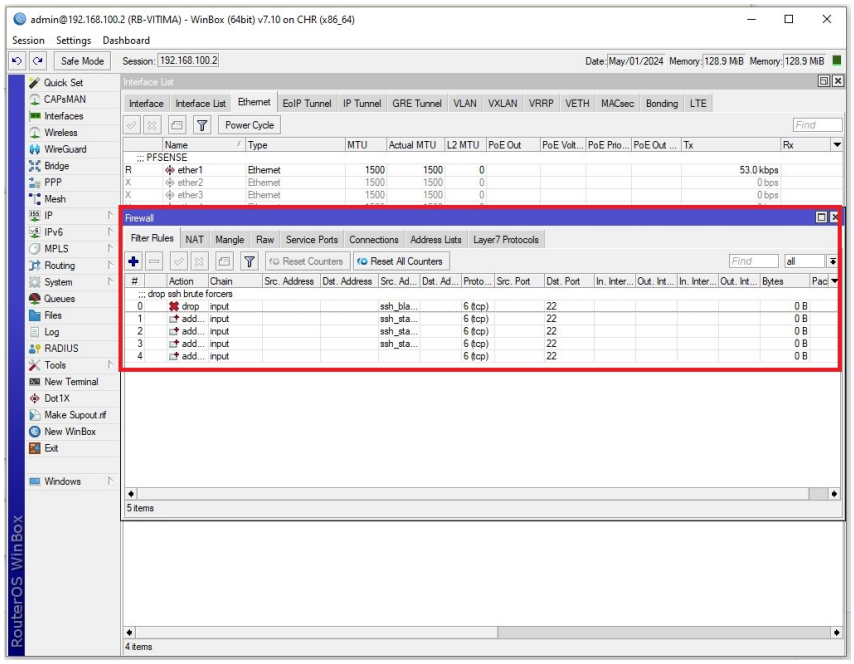
Abaixo segue explicação de cada regra aplicada para o cenário em questão:

```
1 add chain=input protocol=tcp dst-port=22 src-address-list=ssh\_blacklist
    action=drop \ comment="drop ssh brute forcers" disabled=no:
```

A regra acima verifica se há tentativas de conexão *TCP* na porta 22 (a porta padrão do *SSH*) e se o endereço *IP* de origem está na lista *ssh_blacklist*. Se o *IP* de origem estiver na lista, a ação será *"drop"*, o que significa que o pacote será descartado. Esta regra visa bloquear o tráfego de *IP's* conhecidos por realizar ataques de força bruta ao *SSH*.

```
1 add chain=input protocol=tcp dst-port=22 connection-state=new \ src-address
    -list=ssh\_stage3 action=add-src-to-address-list address-list=ssh\
    _blacklist \ address-list-timeout=10d comment="" disabled=no:
```

Figura 12 – Configuração de regras no console do roteador *Mikrotik* - Cenário 2



Fonte: AUTOR, 2024.

É ativada quando uma nova conexão *TCP* é feita na porta 22. Se o endereço *IP* de origem estiver na lista *ssh_stage3*, ele será adicionado à lista *ssh_blacklist*. Isso significa que o *IP* passará a ser bloqueado por um período de 10 dias. A lista *ssh_stage3* é uma lista intermediária que armazena endereços *IP* que já foram detectados como suspeitos em ataques anteriores.

```
1 add chain=input protocol=tcp dst-port=22 connection-state=new \ src-address
    -list=ssh\_stage2 action=add-src-to-address-list address-list=ssh\
    _stage3 \ address-list-timeout=1m comment="" disabled=no:
```

Esta regra é similar à anterior, mas é ativada quando uma nova conexão *TCP* é feita na porta 22 por um endereço *IP* listado na *ssh_stage2*. O *IP* é então movido para a lista *ssh_stage3*, onde é considerado mais suspeito e bloqueado por um período mais longo (1 mês).

```
1 add chain=input protocol=tcp dst-port=22 connection-state=new src-address-
    list=ssh\_stage1 \ action=add-src-to-address-list address-list=ssh\
    _stage2 address-list-timeout=1m comment="" disabled=no:
```

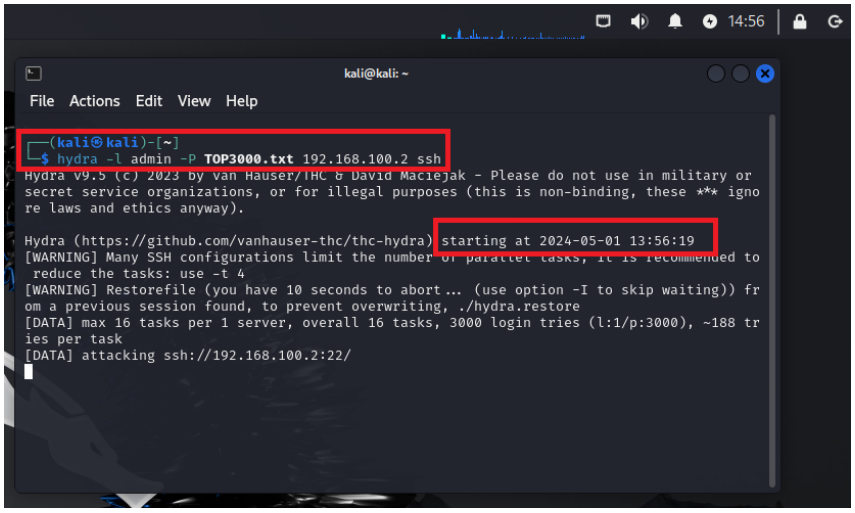
Esta regra é ativada quando uma nova conexão *TCP* é feita na porta 22 por um endereço *IP* listado na *ssh_stage1*. O *IP* é então movido para a lista *ssh_stage2*, onde é considerado mais suspeito e bloqueado por um período de 1 mês.

```
add chain=input protocol=tcp dst-port=22 connection-state=new action=add-
    src-to-address-list \ address-list=ssh\_stage1 address-list-timeout=1m
    comment="" disabled=no:
```

Esta é a regra inicial que é acionada quando uma nova conexão *TCP* é feita na porta 22. O endereço *IP* de origem é adicionado à lista *ssh_stage1*, onde é considerado suspeito, mas não bloqueado imediatamente. Ele é posteriormente movido para listas mais restritivas (como *ssh_stage2* e *ssh_stage3*) se continuar tentando conectar-se ao servidor *SSH*.

A Figura 13 apresenta o início do primeiro ataque do cenário 2:

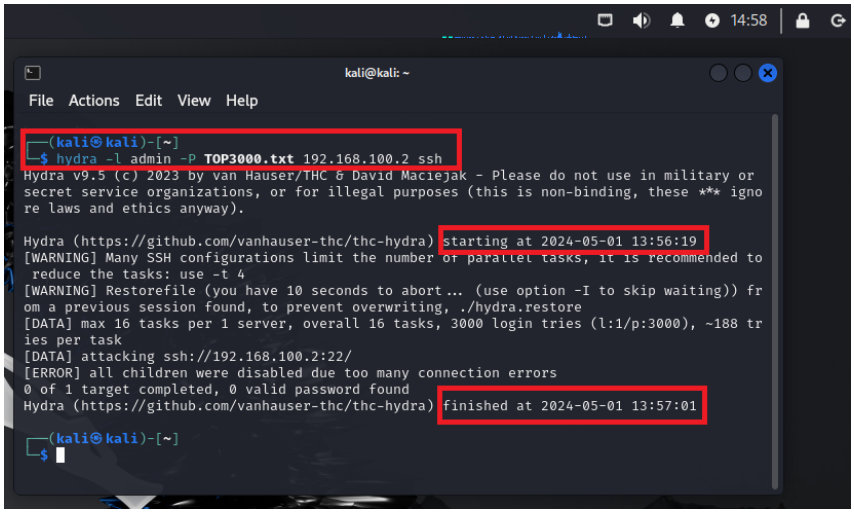
Figura 13 – Terminal *Kali Linux* durante a execução do primeiro ataque - Cenário 2



Fonte: AUTOR, 2024.

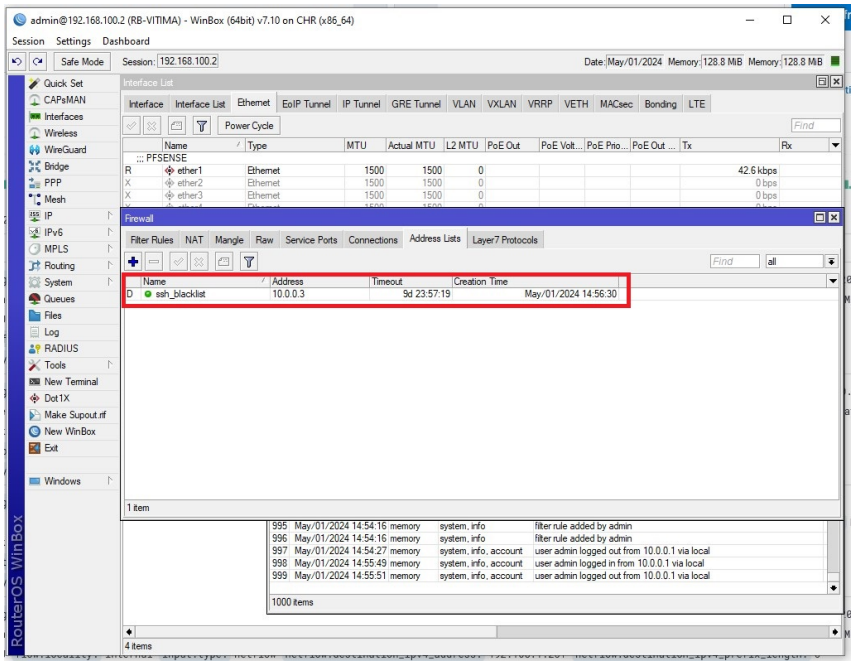
Ao contrário do primeiro cenário, o segundo não durou 10 minutos. O ataque foi interrompido rapidamente em menos de 1 minuto, assim que as regras configuradas no *Mikrotik* identificaram a tentativa de ataque em seu dispositivo. As Figuras 14 e 15 mostram o encerramento do ataque e a adição do *IP* atacante a uma *blackList*.

Figura 14 – Terminal *Kali Linux* durante o fim do primeiro ataque - Cenário 2



Fonte: AUTOR, 2024.

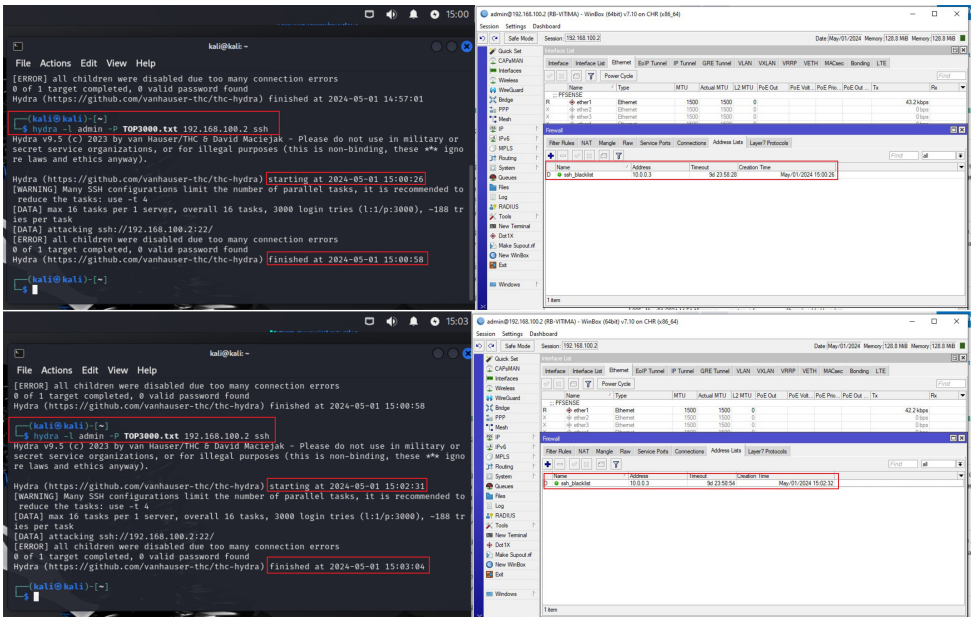
Figura 15 – Console *Mikrotik* - Bloqueio do *IP* atacante - Cenário 2



Fonte: AUTOR, 2024.

O segundo e terceiro ataques apresentaram o mesmo padrão de comportamento, encerrando o ataque e efetuando imediatamente o bloqueio do *IP* do atacante, como mostrado na Figura 16:

Figura 16 – Terminal do *Kali Linux* e console *Mikrotik* - Exibição do segundo e terceiro ataques - Cenário 2

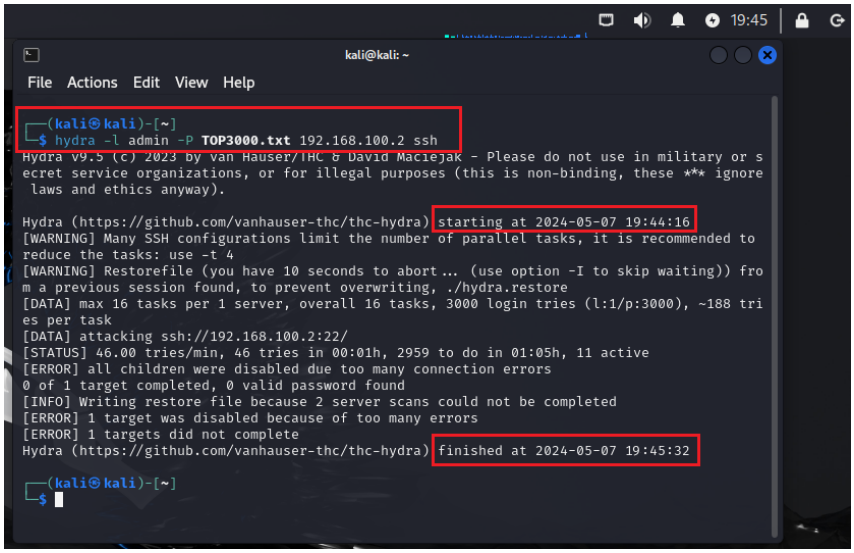


Fonte: AUTOR, 2024.

5.2.3 Terceiro cenário

Para a implementação do terceiro cenário, todas as regras de *firewall* do *Mikrotik* foram desativadas e foi empregado o *IDS/IPS Snort*, com uma regra específica para bloquear ataques de força bruta ao serviço *SSH*. Durante o primeiro teste, o comando do *Hydra* foi executado conforme nos cenários anteriores. Assim como no segundo cenário, o ataque foi prontamente bloqueado logo em seu início, durando pouco mais de 1 minuto, como ilustrado na Figura 17.

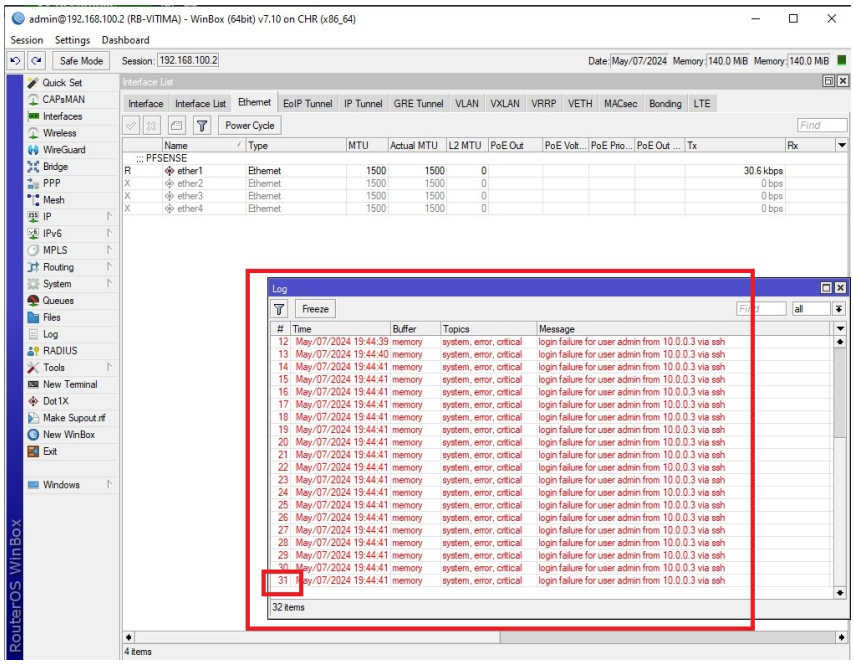
Figura 17 – Terminal *Kali Linux* durante a execução do primeiro ataque - Cenário 3



Fonte: AUTOR, 2024.

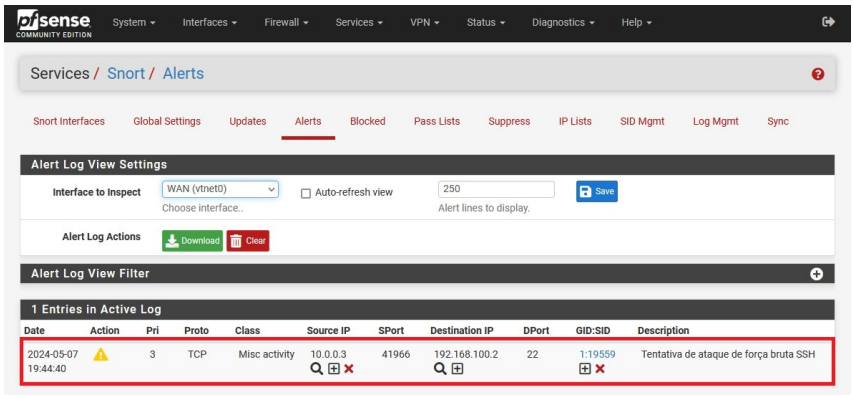
Observa-se que o dispositivo *Mikrotik* alvo recebeu 31 tentativas de acesso malsucedidas antes que o endereço *IP* do atacante fosse identificado, como demonstrado nas Figuras 18 e 19:

Figura 18 – Console *Mikrotik* durante a execução do primeiro ataque - Cenário 3



Fonte: AUTOR, 2024.

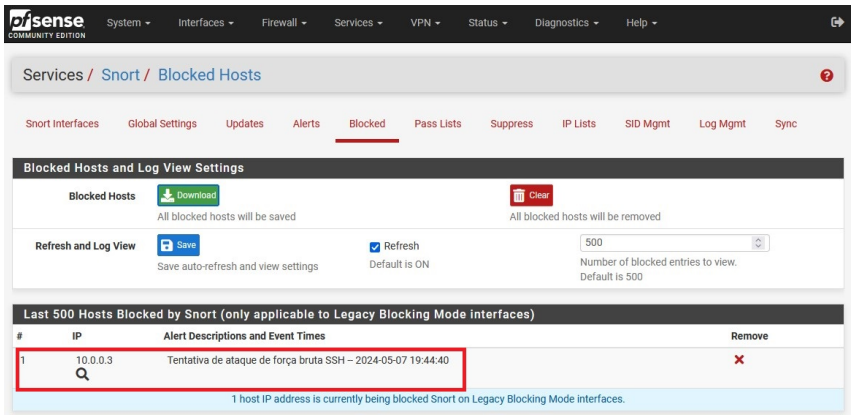
Figura 19 – Alerta do *IDS/IPS* sobre tentativa de ataque de força bruta aos serviços *SSH* - Cenário 3



Fonte: AUTOR, 2024.

A Figura 20 ilustra o momento preciso em que o endereço *IP* do atacante suspeito é bloqueado, devido a diversas tentativas de ataque de força bruta, resultando na interrupção de todo o tráfego proveniente de sua origem.

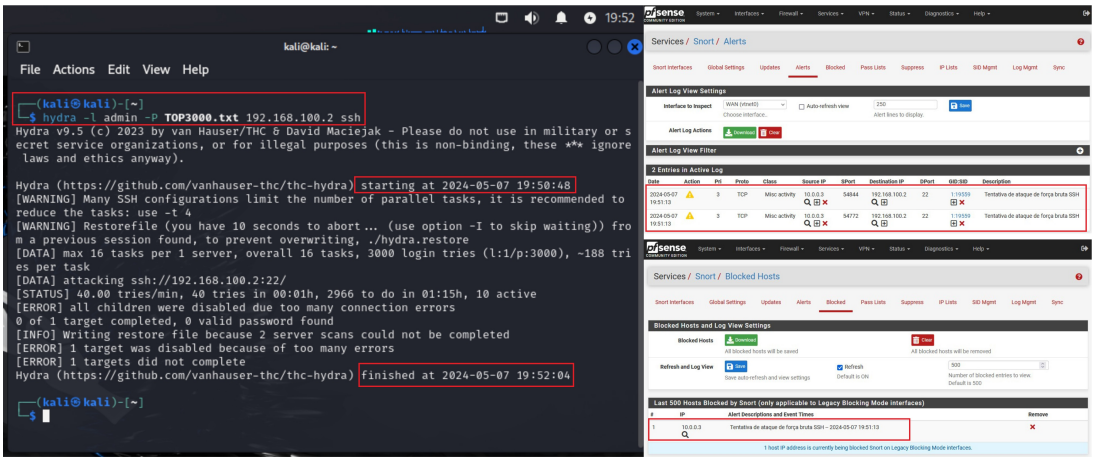
Figura 20 – Bloqueio do *IDS/IPS* sobre tentativa de ataque de força bruta aos serviços *SSH* - Cenário 3



Fonte: AUTOR, 2024.

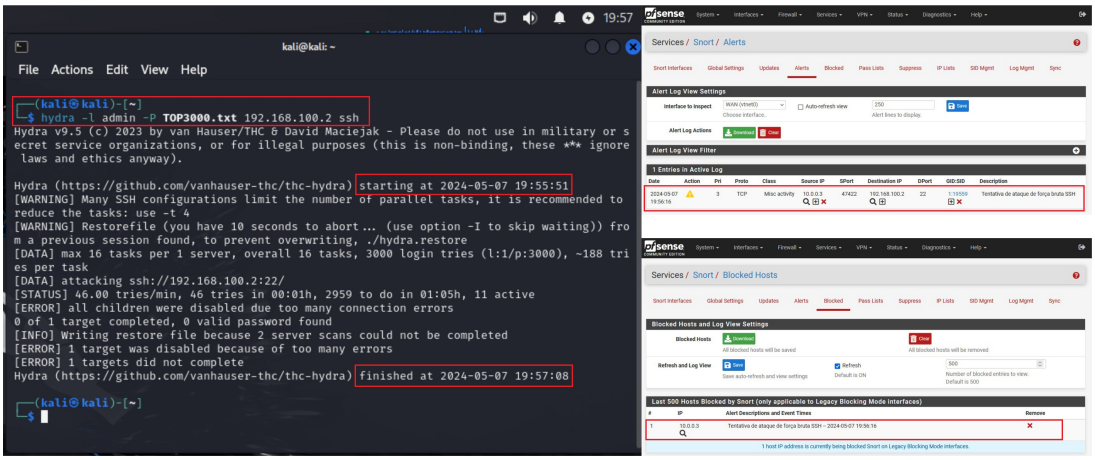
O segundo e terceiro ataques também apresentaram o mesmo padrão de comportamento, encerrando o ataque após identificar o comportamento suspeito e efetuando imediatamente o bloqueio do *IP* do atacante, como mostrado nas Figuras 21 e 22:

Figura 21 – Segundo bloqueio do *IDS/IPS* sobre tentativa de ataque de força bruta aos serviços *SSH* - Cenário 3



Fonte: AUTOR, 2024.

Figura 22 – Terceiro bloqueio do *IDS/IPS* sobre tentativa de ataque de força bruta aos serviços *SSH* - Cenário 3



Fonte: AUTOR, 2024.

6 RESULTADOS E DISCUSSÃO

6.1 Dados Coletados

Ao longo dos três cenários realizados, foram obtidos diversos dados, incluindo informações de fluxo por meio do protocolo *NetFlow* e métricas de desempenho de rede e processamento utilizando o protocolo *SNMP*.

Os dados coletados pelo *ELK*, *plugin Elastiflow* e *IPFIX* em conjunto são acessados por meio da própria interface do *Kibana*, permitindo a definição de um intervalo de tempo, que, neste caso, corresponde ao período de realização do ataque. Embora muitos dados sejam coletados pelo *software*, alguns deles sem relevância para os estudos foram descartados para otimizar as análises. Na Tabela 4, é possível observar todos os tipos de dados que foram capturados pelo *ELK*.

Tabela 4 – Dados Coletados *Netflow*

agent.hostname	agent.name
agent.id	agent.type
agent.ephemeral_id	agent.version
destination.ip	destination.mac
destination.locality	destination.port
ecs.version	event.action
event.category	event.created
event.dataset	event.ingested
event.kind	event.module
event.type	flow.locality
flow.id	input.type
network.bytes	network.direction
network.packets	network.transport
observer.ip	related.ip
service.type	source.bytes
source.ip	source.locality
source.mac	source.packets
@timestamp	netflow.destination_ipv4_address
netflow.destination_mac_address	netflow.flow_end_sys_up_time
netflow.flow_start_sys_up_time	netflow.icmp_code_ipv4
netflow.icmp_type_ipv4	netflow.igmp_type
netflow.ip_class_of_service	netflow.ip_header_length
netflow.ip_next_hop_ipv4_address	netflow.ip_total_length
netflow.ip_ttl	netflow.ip_version
netflow.is_multicast	netflow.iana_number
netflow.octet_delta_count	netflow.packet_delta_count
netflow.post_destination_mac_address	netflow.post_nat_destination_ipv4_address
netflow.post_nat_source_ipv4_address	netflow.post_napt_destination_transport_port
netflow.post_napt_source_transport_port	netflow.protocol_identifier
netflow.source_ipv4_address	netflow.source_ipv4_prefix_length
netflow.source_mac_address	netflow.source_transport_port
netflow.system_init_time_milliseconds	netflow.tcp_acknowledgement_number
netflow.tcp_control_bits	netflow.tcp_sequence_number
netflow.tcp_window_size	netflow.transport
netflow.type	netflow.udp_message_length
netflow.exporter.address	netflow.exporter.source_id
netflow.exporter.timestamp	netflow.exporter.uptime_millis

Fonte: AUTOR, 2024.

Os dados *SNMP* foram obtidos por meio do *software Zabbix*. Na tabela 5, é possível discernir todos os dados adquiridos via protocolo *SNMP*. Alguns desses dados, por vez, revelaram-se irrelevantes para os propósitos do estudo em questão, e, portanto, serão devidamente desconsiderados.

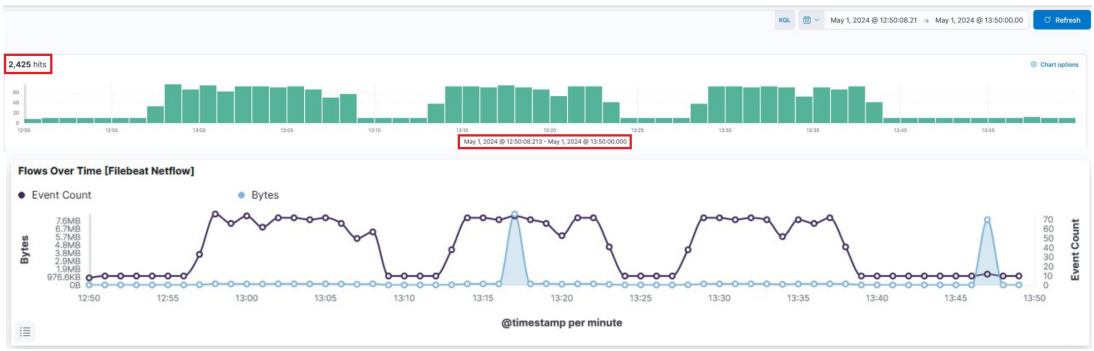
Tabela 5 – Dados Coletados *SNMP*

CPU utilization	Space utilization
Total space	Used space
Firmware version	Hardware model name
Hardware serial number	ICMP loss
ICMP ping	ICMP response time
Bits received	Bits sent
Inbound packets discarded	Inbound packets with errors
Interface type	Operational status
Outbound packets discarded	Outbound packets with errors
Speed	Memory utilization
Operating system	SNMP agent availability
SNMP traps (fallback)	System contact details
System description	System location
System name	System object ID
Total memory	Uptime (hardware)
Uptime (network)	Used memory
Available memory	Context switches per second
CPU idle time	CPU interrupt time
CPU iowait time	CPU nice time
CPU system time	CPU user time
CPU idle time: CPU utilization	Free memory
Free swap space	Free swap space in %
Interrupts per second	Load average (1m avg)
Load average (5m avg)	Load average (15m avg)
Memory (buffers)	Memory (cached)
Number of CPUs	DHCP server status
DNS server status	Firewall rules count
Fragmented packets	Inbound IPv4 packets blocked
Inbound IPv4 packets passed	Inbound IPv4 traffic blocked
Inbound IPv4 traffic passed	Inbound IPv6 packets blocked
Inbound IPv6 packets passed	Inbound IPv6 traffic blocked
Inbound IPv6 traffic passed	Outbound IPv4 packets blocked
Outbound IPv4 packets passed	Outbound IPv4 traffic blocked
Outbound IPv4 traffic passed	Outbound IPv6 packets blocked
Outbound IPv6 packets passed	Outbound IPv6 traffic blocked
Outbound IPv6 traffic passed	Rules references count
Normalized packets	Packet filter running status
Packets dropped due to memory limitation	Packets matched a filter rule
Packets with bad offset	Short packets
Source tracking table current	Source tracking table limit
Source tracking table utilization in %	State of nginx process
States table current	States table limit
States table utilization in %	

Fonte: AUTOR, 2024.

No primeiro cenário, observou-se a ocorrência de 3 ataques, nos quais não se emprega nenhum mecanismo de bloqueio contra ataques de força bruta. Esta ausência de contramedidas resulta em um substancial aumento de tentativas de acessos, conforme evidenciado no gráfico da Figura 23. Durante o período de teste, com uma duração de uma hora, foram registrados 2.425 *hits*, os quais representam fluxos de rede monitorados. Nesse intervalo, é possível identificar 3 picos de tráfego ascendente, cada um com duração média de 10 minutos. Este comportamento é diretamente correlacionado com a ocorrência dos ataques.

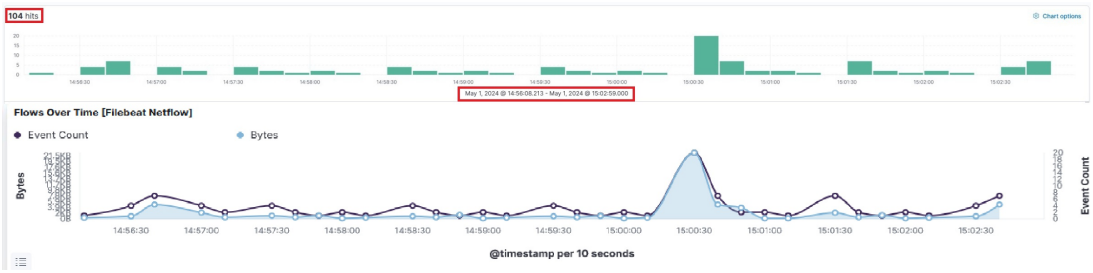
Figura 23 – Gráfico *NetFlow* gerado pela ferramenta *ELK* - Cenário 1



Fonte: AUTOR, 2024.

No segundo cenário, destaca-se a implementação de uma regra de *firewall* no roteador *Mikrotik*, alvo do ataque. Essa medida consiste em adicionar o endereço *IP* à lista negra e bloqueá-lo quando forem registradas 10 tentativas de *logins* malsucedidas via *SSH* em um intervalo de um minuto. Assim como no cenário anterior, foram realizadas 3 tentativas de ataques, as quais foram prontamente bloqueadas. Em contraste com o primeiro cenário, observou-se uma significativa redução no período de testes, que foi reduzido para 10 minutos, e, principalmente, uma diminuição considerável no número de acessos. Isso ocorre devido à efetivação da regra de bloqueio, que impede a comunicação dos *IPs* presentes na lista negra. Além disso, percebeu-se uma notável alteração na estrutura do gráfico, que já não apresenta os 3 picos de fluxo evidenciados no gráfico da Figura 24.

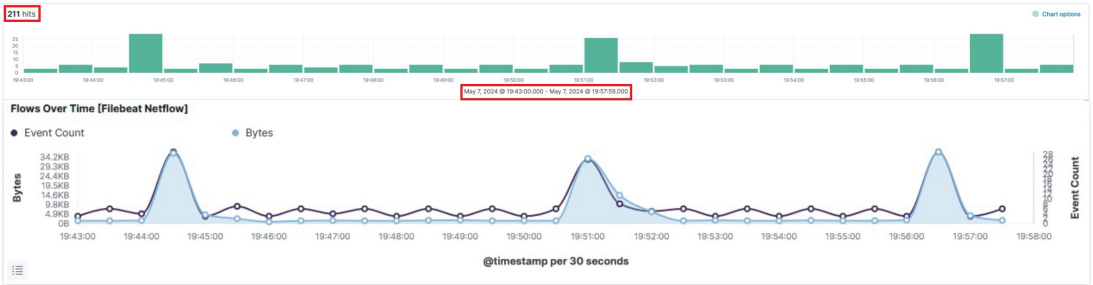
Figura 24 – Gráfico *NetFlow* gerado pela ferramenta *ELK* - Cenário 2



Fonte: AUTOR, 2024.

O terceiro cenário é identificado pelo emprego do *software Snort* em conjunto com o *PFsense*. Durante os ataques, foi ativada uma regra para bloquear qualquer *IP* que esteja realizando uma exploração de ataque de força bruta na porta 22, dentro da rede interna, especificamente no *Mikrotik* alvo. No gráfico exibido na Figura 25, foi perceptível uma redução no fluxo de rede. Diferentemente do segundo cenário, observou-se um comportamento distinto no gráfico, caracterizado por três pequenos momentos de picos que coincidem com o início do ataque e a intervenção do *software* de bloqueio.

Figura 25 – Gráfico *NetFlow* gerado pela ferramenta *ELK* - Cenário 3



Fonte: AUTOR, 2024.

Após a execução dos ataques e a análise destes mediante as técnicas apresentadas, alguns resultados foram conclusivos, cada um com suas próprias características e eficácia, levando em consideração aspectos relevantes como infraestrutura de rede, consumo de recursos e eficiência.

6.2 Avaliação da viabilidade e eficácia

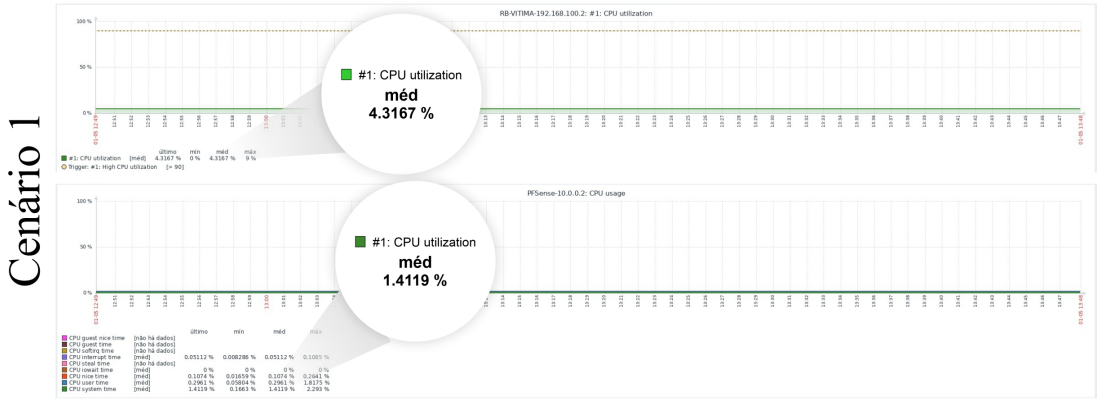
Mantellis e Fellipe (2018) definem a eficácia de um *IDS/IPS* por sua capacidade de intervir de forma imediata para bloquear um ataque durante sua ocorrência, isolando a rede interna de endereços *IP* maliciosos.

É notável que no Cenário 2 de testes, com a implementação do *firewall* diretamente no *MikroTik*, foi possível validar o recurso "*Bruteforce Login Prevention*", o qual constitui uma funcionalidade integrada aos roteadores *MikroTik* com o propósito de proteger contra ataques de força bruta. Quando ativado, tornou-se viável monitorar as tentativas de *login* originadas de endereços *IP* específicos. Conforme evidenciado nos testes realizados, representados na Figura 15, após um determinado número de tentativas de *login* malsucedidas em um intervalo de tempo pré-definido, o endereço *IP* do dispositivo de origem foi imediatamente bloqueado.

Já no Cenário 3 de testes, o *Snort* demonstrou eficácia na prevenção de ataques de força bruta. Além de fornecer alertas imediatos sobre tentativas de ataque, o *Snort* também executou o bloqueio tempestivo do endereço *IP* de origem do ataque. No entanto, ele adotou um método distinto em comparação ao *Mikrotik*. Enquanto o *Mikrotik* baseou seu bloqueio nas tentativas malsucedidas de *login* em seu *console*, o *Snort* realizou uma análise dos pacotes em trânsito, buscando identificar padrões como sequências de tentativas repetidas e rápidas de *login* com diferentes credenciais, taxas anormais de tentativas de conexão em um curto intervalo de tempo e ocorrências de autenticação malsucedida em serviços específicos. Essa abordagem possibilita uma conclusão mais precisa e efetiva, resultando no bloqueio apropriado do atacante, conforme a Figura 20.

Mishra *et al.* (2022) cita em seu artigo, que ao implementar diferentes *firewalls* de código aberto, a análise considerou o consumo de *CPU* como fator essencial na eficiência da solução. Assim como na execução do Cenário 1, no qual não foi implementada nenhuma medida de bloqueio contra ataques de força bruta, foram monitorados os recursos computacionais dos dispositivos presentes na rede da vítima. Especificamente, observou-se a taxa de utilização da *CPU* do *Mikrotik* e do *IDS/IPS Snort*, com médias registradas de 4.3167% e 1.4119%, respectivamente, conforme demonstrado na Figura 26:

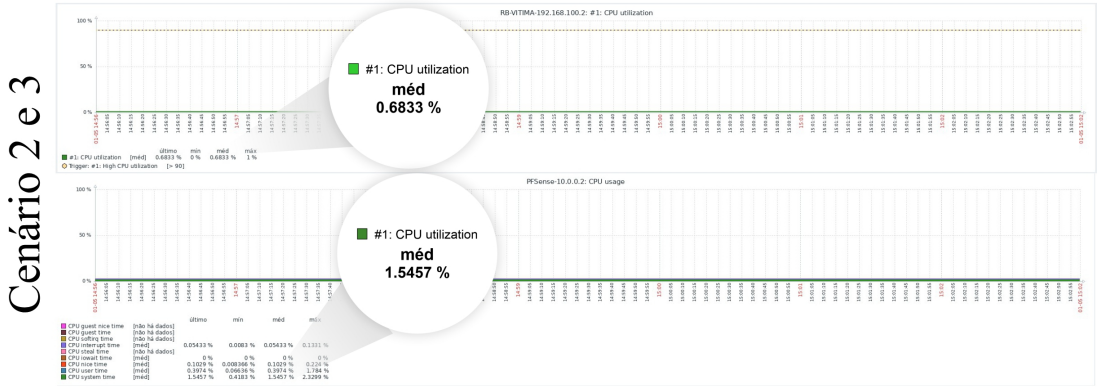
Figura 26 – Recursos computacionais do *Mikrotik* e *IDS/IPS* durante Cenário 1



Fonte: AUTOR, 2024.

Ao longo da execução dos Cenários 2 e 3, nos quais o *firewall* do *Mikrotik* foi habilitado e posteriormente foram aplicadas as regras de bloqueio do *Snort*, foi observada uma significativa redução na utilização da *CPU* do *Mikrotik*. Esta passou a apresentar médias de 0.6833%, enquanto houve uma leve oscilação na utilização da *CPU* do *IDS/IPS Snort*, cuja média foi de 1.5457%, conforme evidenciado na Figura 27:

Figura 27 – Recursos computacionais do *Mikrotik* e *IDS/IPS* durante Cenários 2 e 3



Fonte: AUTOR, 2024.

Com os resultados foi possível concluir que o emprego das regras do *Mikrotik* e posteriormente do *Snort* não resultou em sobrecarga de processamento, pelo contrário, agiram favoravelmente diminuindo a quantidade de pacotes maliciosos recebidos e consequentemente de processamento. Observou-se que a eficiência computacional dessas soluções é crucial para garantir que a proteção não cause degradações significativas no desempenho do sistema (MISHRA *et al.*, 2022).

7 CONCLUSÃO

Em conclusão, os resultados obtidos através da execução dos ataques e subsequente análise, mediante as técnicas apresentadas, revelam a eficácia de abordagens distintas na prevenção de ataques de força bruta nos cenários propostos. No Cenário 2, a implementação do *firewall* diretamente no *MikroTik* proporcionou uma validação bem-sucedida do recurso "*Bruteforce Login Prevention*", demonstrando a capacidade de proteger contra tentativas maliciosas de *login*. A configuração altamente personalizável desse recurso, aliada à capacidade de monitorar e bloquear endereços *IP* específicos, oferece uma camada adicional de segurança aos roteadores *MikroTik*, mitigando significativamente o risco de invasões.

Por vez, no Cenário 3, a utilização do *IDS/IPS Snort* apresentou resultados igualmente promissores na detecção e bloqueio de ataques de força bruta. Ao empregar uma abordagem baseada na análise detalhada dos pacotes em trânsito, o *Snort* foi capaz de identificar padrões característicos de tentativas maliciosas de *login*, resultando em alertas imediatos e efetivos bloqueios dos endereços *IP* dos atacantes.

Em resumo, os resultados obtidos destacam a importância de uma abordagem proativa na segurança de redes, seja tanto a implementação de medidas de proteção diretas nos dispositivos, como *firewalls*, quanto o uso de sistemas de detecção e prevenção de intrusões, como o *Snort*. Ao integrar essas diferentes camadas de defesa, é possível fortalecer significativamente a segurança da rede, reduzindo a probabilidade e o impacto de ataques de força bruta e outras ameaças cibernéticas.

Como trabalhos futuros, têm-se a possibilidade de avaliar a eficácia de soluções complementares em conjunto com o "*Bruteforce Login Prevention*"e o *Snort* para fortalecer ainda mais a segurança da rede. Isso pode incluir o uso de sistemas de autenticação multifatorial, implementação de políticas de senha mais robustas e o emprego de técnicas de *machine learning* para detecção de anomalias de tráfego.

REFERÊNCIAS

ALVARENGA, I. D.; RAMOS, B. L. Simple network management protocol (snmp). 2011. Disponível em: <https://www.gta.ufrj.br/grad/11_1/snmp/index.html>.

ASSIS, F. M. F. de; COUTINHO, M. A.; FILHO, J. B. da S.; MACEDO, E. L. C.; MORAES, L. F. M. de. Coleta e detecção de anomalias em fluxos de rede. 2021. Disponível em: <https://sol.sbc.org.br/index.php/wgrs/article/view/17188>.

AUGUSTO, A. L. A. Cibersegurança na computação quântica. 2022. Disponível em: <https://repositorio.ufrn.br/handle/123456789/50316>.

AWS. **O que é a pilha ELK?** 2023. Disponível em: <https://aws.amazon.com/pt/what-is/elk-stack/>.

BACHINSKI, R. E.; ALVES, V. M.; SILVA, E. F. da; CASTANHO, C. L. de O.; CAVALHEIRO, B. O.; LIMA, S.; ESPINDOLA, P. Implementação de funcionalidade preventiva em um sistema de detecção de intrusão inteligente. 2020. Disponível em: <https://ojs.brazilianjournals.com.br/ojs/index.php/BRJD/article/view/18635>.

BADOTRA, S.; PANDA, S. N. Sistema de detecção precoce de ddos baseado em snort usando opendaylight e sistema operacional de rede aberta em rede definida por software. 2021. Disponível em: <https://link.springer.com/article/10.1007/s10586-020-03133-y>.

BARROS Álvaro Gonçalves de; SOUZA, C. H. M. de; TEIXEIRA, R. Evolução das comunicações até a internet das coisas: a passagem para uma nova era da comunicação humana. 2021. Disponível em: <https://pdfs.semanticscholar.org/fc9e/86ed78a37d3bf92717d7561ee6f1f964c801.pdf>.

BODNAR, D. **O que é um TCP/IP e o que isso significa?** 2022. Disponível em: <https://www.avg.com/pt/signal/what-is-tcpip#:~:text=TCP%2FIP%20%C3%A9%20a%20abrevia%C3%A7%C3%A3o,eles%20em%20todas%20as%20redes.>

CALGAROTO, C. O direito à privacidade na internet: panorama, responsabilização civil e inovações do marco civil da internet (lei nº 12.965/2014). 2021. Disponível em: <https://repositorio.animaeducacao.com.br/handle/ANIMA/16882>.

CERT. **Incidentes Notificados ao CERT.br - Período 1999–2020**. 2020. Disponível em: <https://stats.cert.br/historico/incidentes/2020-jan-jun/analise.html>.

CETIC. **Pesquisas e indicadores**. 2020. Disponível em: <https://cetic.br/pt/pesquisas/>.

COMER, D. E. **Redes de Computadores e Internet**. [S.l.]: Bookman Editora, 2016.

DALEFFE, E. A. G. de O.; AMADIO, R. A.; GAVILAN, J. C. Redes de computadores com Ênfase em vpn. 2020. Disponível em: <https://repositorio.kanix.com.br/arquivos/2020/3e24148e15f071656cf97cd4cec26e76.pdf>.

DAVOGLIO, A. C.; WULFF, M. T. S. G.; MYSZAK, S. R.; LIBRELATO, G. R. Redes de computadores. 2021. Disponível em: <https://www.academia.edu/download/82405891/REDES_DE_COMPUTADORES.pdf>.

DEFLEUR, M. L.; BALL-ROKEACH, S. **Teorias da comunicação de massa**. [S.l.]: Zahar, 1993. ISBN 8571102023.

DIORIO, R. F.; SERAFIM, E.; ALVES, K. R.; MEIRA, M. C. Ataques de força bruta: Um estudo prático. 2019.

DOMINGOS, T.; PEREIRA, S.; REIS, D.; SILVA, C.; BARRéRE, E. Gerenciamento de uma rede através do protocolo snmp. 2005. Disponível em: <https://www.aedb.br/seget/arquivos/artigos05/335_EAGLE_SEGET.pdf>.

EDMUNDO, T. H. R. Implantação de teste de vulnerabilidades em aplicações web seguindo metodologia owasp. 2021. Disponível em: <http://ric.cps.sp.gov.br/handle/123456789/12764>.

EISENBERG, J. Internet e política. 2019. Disponível em: <https://cadernosdolegislativo.almg.gov.br/ojs/index.php/cadernos-ele/article/view/329>.

ERLACHER, F.; DRESSLER, F. Detecção de intrusão baseada em fluxo de alta velocidade usando assinaturas compatíveis com snort. 2020. Disponível em: <https://ieeexplore.ieee.org/abstract/document/8999496/>.

FARES, A. A. Y. R. Proposta de integração de um sistema de detecção de intrusão (ids) entre uma rede sdn e uma honeynet. 2021. Disponível em: <http://icts.unb.br/jspui/handle/10482/41908>.

FERREIRA, W. B.; AMADIO, R. A. SeguranÇa da informaÇÃo: Um estudo sobre a parte fÍsica da rede de computadores da prefeitura municipal de jaciara-mt. 2022. Disponível em: <https://repositorio.kanix.com.br/arquivos/2019/f75ee79f51ef530b6ec62d9bd70a3d81.pdf>.

FREUND, G. P.; SEMBAY, M. J.; MACEDO, D. D. J. D. Proveniência de dados e segurança da informação: relações interdisciplinares no domínio da ciência da informação. 2019. Disponível em: <https://periodicos.unb.br/index.php/RICI/article/view/21203>.

GUAREZI, J. Engenharia social: avaliação de riscos e vulnerabilidades tendo o fator humano como o elo mais fraco da segurança da informação. 2019. Disponível em: <https://repositorio.animaeducacao.com.br/handle/ANIMA/11039>.

JAIN, G.; ANUBHA. Aplicação de snort e wireshark na análise de tráfego de rede. in: Série de conferências do iop: Ciência e engenharia de materiais. 2021. Disponível em: <https://iopscience.iop.org/article/10.1088/1757-899X/1119/1/012007/meta>.

JúNIOR, L. C. S.; LINS, T. S. Ransomware : análise técnica e prevenção. 2023. Disponível em: <https://monografias.ufop.br/handle/35400000/6049>.

LEME, R. S.; BLANK, M. Lei geral de proteção de dados e segurança da informação na área da saúde. 2020. Disponível em: <https://www.cadernos.prodisa.fiocruz.br/index.php/cadernos/article/view/690>.

LIMA, L. F. de. Como mitigar a utilização de dispositivos iot em ataques ddos: uma revisão sistemática. 2019. Disponível em: <http://ric.cps.sp.gov.br/handle/123456789/4245>.

LOURENÇO, R. M.; DUARTE, R. P. Gestão de segurança da informação. 2020. Disponível em: <http://ric.cps.sp.gov.br/handle/123456789/8340>.

LUCAS, T. J.; COSTA, K. A. P. da; MORAES, E. A.; JúNIOR, P. R. G. H.; NEVES, M. J. das. Stacking-based committees para detecção de ataques em redes de computadores - uma abordagem por exaustão. 2021. Disponível em: <https://sol.sbc.org.br/index.php/sbrc/article/view/16753>.

MACHIAVELLI, N. **O Príncipe**. [S.l.: s.n.], 1532.

MANTELLIS, G. G.; FELLIPE, K. Detecção e prevenção de invasões em redes corporativas utilizando o package snort em integração com o pfsense. 2018. Disponível em: <https://www.unifafibe.com.br/revistasonline/arquivos/revistasisunifafibe/sumario/55/07112018195841.pdf>.

MARQUEZIN, G. M.; SILVA, K. de Fátima do N.; PINTO, W. F. Botnet (rede zumbi). 2021. Disponível em: <http://ric.cps.sp.gov.br/handle/123456789/7190>.

MARTINS, J. Interconexão de redes: Tecnologias e protocolos para a integração corporativa. 2019. Disponível em: <https://www.academia.edu/download/61720034/1991_Paper_-_Interconexao_de_Redes_-_Joberto_-_Congresso_Nacional_de_Informatica20200108-32479-1phbf2b.pdf>.

MARTINS, R. R. Autoproteção para a camada de aplicação: uma abordagem baseada em técnicas de aprendizado e no laço de controle mape-k. 2022. Disponível em: <https://repositorio.unesp.br/handle/11449/217074>.

MENDES, D. R. **Redes de Computadores: Teoria e Prática**. [S.l.]: Novatec Editora, 2020.

MISHRA, A.; ALZOUBI, Y. I.; GILL, A. Q.; ANWAR, M. J. Cybersecurity enterprises policies: A comparative study. 2022. Disponível em: <https://www.mdpi.com/1424-8220/22/2/538>.

NASCIMENTO, P. A. M. M.; RAMOS, D. L.; MELO, A. A. S. de; CASTIONI, R. Acesso domiciliar à internet e ensino remoto durante a pandemia. 2020. Disponível em: <https://repositorio.ipea.gov.br/handle/11058/10228>.

NUNES, L. A. O uso do zabbix no monitoramento de infraestrutura dos clouds e servidores de uma empresa de software. 2018. Disponível em: <https://repositorio.animaeducacao.com.br/items/182d68e9-e0dc-46c4-89a8-4a638914c389>.

OLIVEIRA. **Levantamento mostra que ataques cibernéticos no Brasil cresceram 94Disponível em: <https://www.cnnbrasil.com.br/tecnologia/levantamento-mostra-que-ataques-ciberneticos-no-brasil-cresceram-94/>.**

OLIVEIRA, N. S. de; GOMES, M. A.; LOPES, R.; NOBRE, J. C. Segurança da informação para internet das coisas (iot): uma abordagem sobre a lei geral de proteção de dados (lgpd). 2019. Disponível em: <https://seer.ufrgs.br/reic/article/view/88790>.

OSTERLE, H.; FLEISCH, E.; ALT, R. **Business Networking: Shaping Collaboration Between Enterprises**. [S.l.]: Springer Ciência e Mídia de Negócios, 2001.

PACHECO, F. G. Análise das técnicas de segurança do framework laravel contra ataques as aplicações web. 2019. Disponível em: <http://repository.ufrpe.br/handle/123456789/1427>.

PAES, D. S. F. Detecção preditiva de anomalias em redes de computadores com utilização de aprendizagem de máquina. 2023. Disponível em: <https://repositorio.unifei.edu.br/jspui/handle/123456789/3552>.

PEREIRA, G. F. Técnicas de proteção contra ameaças digitais do tipo ransomware em plataformas windows. 2022. Disponível em: <http://repositorio.unesc.net/handle/1/8855>.

PEREIRA, O. J.; ABRANTES, R. P.; FERNANDES, F. A.; SILVA, L. S. da. A taxonomia de bloom revisada como suporte para o planejamento de uma disciplina de redes de computadores. 2021. Disponível em: <https://sol.sbc.org.br/index.php/wei/article/view/15891>.

QUITTEK, J. Requirements for ip flow information export (ipfix). 2004. Disponível em: <http://www.ietf.org/rfc/rfc3917.txt>.

RAPôSO, C. F. L.; LIMA, H. M. de; JUNIOR, W. F. de O.; SILVA, P. A. F.; BARROS, E. de S. Lgpd - lei geral de proteÇÃo de dados pessoais em tecnologia da informaÇÃo: Revisão sistemática. 2019. Disponível em: <https://revistas.cesmac.edu.br/administracao/article/view/1035>.

ROSA, V. M. V. D.; SILVA, J. C. M. D.; KRüGER, V. M.; BORGES, A. P.; BALBINOT, M. Z.; DECEZARE, J. K.; PELLOSO, M. Mapeamento das características do protocolo tcp explorada por ataques ddos. 2022. Disponível em: <https://publicacoes.ifc.edu.br/index.php/mic/article/view/2788>.

RUIZ, V.; SAPIA, H. Coleta, tratamento e armazenamento de fluxos no padrÃo netflow. 2016. Disponível em: <https://revistas.unoeste.br/index.php/ce/article/view/1549/1609>.

RöHRS, R. Elasticsearch (elk). 2021. Disponível em: <https://www.ufsm.br/pet/sistemas-de-informacao/2021/06/08/elasticsearch-elk>.

SANTOS, A.; LAURENCE, J. Análise de vulnerabilidades de segurança em software de produtos tecnológicos de uma indústria da grande florianópolis. 2022. Disponível em: <https://repositorio.animaeducacao.com.br/items/ed08ac81-d332-487e-98ab-caf7b70d328e>.

SANTOS, F. R.; SANTOS, A. N. dos. Análise de vulnerabilidades de segurança em software de produtos tecnológicos de uma indústria da grande florianópolis. 2022. Disponível em: <https://repositorio.animaeducacao.com.br/items/ed08ac81-d332-487e-98ab-caf7b70d328e>.

SOUZA, J. P. B. de. Comunicação entre sma embarcados: Uma arquitetura baseada em protocolos da camada de aplicação. 2023. Disponível em: <https://www.researchgate.net/publication/373772562>.

TALIANI, E. A. R. Análise da evolução das ameaças cibernéticas entre 2010 e 2021. 2022. Disponível em: <https://ric.cps.sp.gov.br/handle/123456789/12217>.

TANENBAUM, A. S. **Redes de Computadores**. [S.l.]: Fourth Editora, 2002.

TANENBAUM, A. S.; WETHERALL, D. **Redes de Computadores**. [S.l.]: Artmed Editora, 2011.

TESSARIO, A. Utilização do snort para a avaliação da eficácia de firewalls. 2013. Disponível em: <https://lume.ufrgs.br/handle/10183/66083>.

VAZ, G. M.; RIZZETTI, T. A.; FILHO, W. P. Um estudo de caso sobre a implantação de um ambiente de prevenção de intrusões com a ferramenta suricata. 2021. Disponível em: <https://sol.sbc.org.br/index.php/sbseg_estendido/article/view/17358>.

WALEED, A.; JAMALI, A. F.; MASOOD, A. Which open-source ids? snort, suricata or zeek. 2022. Disponível em: <https://www.sciencedirect.com/science/article/abs/pii/S1389128622002420>.

ANEXO A – TUTORIAL DE INSTALAÇÃO DO *VMWARE WORKSTATION*

Segue o *link* que disponibiliza o tutorial de instalação e configuração:
<https://docs.vmware.com/br/VMware-Workstation-Player-for-Windows/17.0/com.vmware.player.win.using.doc/GUID-CEC8A8B9-358F-47DF-90DC-24AAE2588A3F.html>

ANEXO B – TUTORIAL DE INSTALAÇÃO DO *EVE-NG* NO *VMWARE WORKSTATION*

Segue o *link* que disponibiliza o tutorial de instalação e configuração:

<https:

//www.eve-ng.net/index.php/documentation/installation/virtual-machine-install/>

ANEXO C – TUTORIAL DE VIRTUALIZAÇÃO DO *PFSENSE* NO *EVE-NG*

Segue o *link* que disponibiliza o tutorial de instalação e configuração:

<https://www.eve-ng.net/index.php/3380-2/>

ANEXO D – TUTORIAL DE INSTALAÇÃO DO PACOTE *SNORT* NO *PFSENSE*

Segue o *link* que disponibiliza o tutorial de instalação e configuração:

<https://techexpert.tips/pt-br/pfsense-pt-br/instalacao-de-snort-em-pfsense/>

ANEXO E – TUTORIAL DE VIRTUALIZAÇÃO DO *MIKROTIK* NO *EVE-NG*

Segue o *link* que disponibiliza o tutorial de instalação e configuração:

<https:

//www.eve-ng.net/index.php/documentation/howtos/howto-add-mikrotik-cloud-router/>

ANEXO F – TUTORIAL DE VIRTUALIZAÇÃO DO *KALI LINUX* NO *EVE-NG*

Segue o *link* que disponibiliza o tutorial de instalação e configuração:
<https://www.eve-ng.net/index.php/documentation/howtos/howto-create-own-linux-host-image/>


UFVJM