MONOGRAFIA DE PROJETO FINAL DE GRADUAÇÃO

ESTUDO, IMPLEMENTAÇÃO E ANÁLISE DE RESULTADOS DE FERRAMENTA DE DETECÇÃO DE INTRUSÃO PARA PROTEÇÃO DE ENDPOINTS EM AMBIENTE CONTROLADO

Hítalo Bruno Pereira Alves

Curso Superior de Engenharia de Redes de Comunicação

DEPARTAMENTO DE ENGENHARIA ELÉTRICA
FACULDADE DE TECNOLOGIA
UNIVERSIDADE DE BRASÍLIA

UNIVERSIDADE DE BRASÍLIA
Faculdade de Tecnologia

MONOGRAFIA DE PROJETO FINAL DE GRADUAÇÃO

ESTUDO, IMPLEMENTAÇÃO E ANÁLISE DE RESULTADOS DE FERRAMENTA DE DETECÇÃO DE INTRUSÃO PARA PROTEÇÃO DE ENDPOINTS EM AMBIENTE CONTROLADO

Hítalo Bruno Pereira Alves

Monografia de Projeto Final de Graduação submetida ao Departamento de Engenharia Elétrica como requisito parcial para obtenção do grau de Bacharel em Engenharia de Redes de Comunicação

Banca Examinadora

Dr. Georges Daniel Amvame Nze, EnE/UnB _____
Orientador

Dr. Fábio Lúcio Lopes de Mendonça, EnE/UnB _____
Examinador Interno

Ms. Diego Martins de Oliveira, IFB/Brasília _____
Examinador Externo

FICHA CATALOGRÁFICA

ALVES, H.B.P.
ESTUDO, IMPLEMENTAÇÃO E ANÁLISE DE RESULTADOS DE FERRAMENTA DE DETECÇÃO DE INTRUSÃO PARA PROTEÇÃO DE ENDPOINTS EM AMBIENTE CONTROLADO [Distrito Federal] 2022.
xvi, 84 p., 210 x 297 mm (ENE/FT/UnB, Bacharel, Engenharia de Redes de Comunicação, 2022).
Monografia de Projeto Final de Graduação - Universidade de Brasília, Faculdade de Tecnologia.
Departamento de Engenharia Elétrica

1. Proteção de Endpoints	2. Detecção de Intrusão
3. Proteção de Dados	4. Ataques cibernéticos
I. ENE/FT/UnB	II. Título (série)

REFERÊNCIA BIBLIOGRÁFICA

ALVES, H.B.P. (2022). *ESTUDO, IMPLEMENTAÇÃO E ANÁLISE DE RESULTADOS DE FERRAMENTA DE DETECÇÃO DE INTRUSÃO PARA PROTEÇÃO DE ENDPOINTS EM AMBIENTE CONTROLADO*. Monografia de Projeto Final de Graduação, Departamento de Engenharia Elétrica, Universidade de Brasília, Brasília, DF, 84 p.

CESSÃO DE DIREITOS

AUTOR: Hítalo Bruno Pereira Alves
TÍTULO: ESTUDO, IMPLEMENTAÇÃO E ANÁLISE DE RESULTADOS DE FERRAMENTA DE DETECÇÃO DE INTRUSÃO PARA PROTEÇÃO DE ENDPOINTS EM AMBIENTE CONTROLADO.
GRAU: Bacharel em Engenharia de Redes de Comunicação ANO: 2022

Hítalo Bruno Pereira Alves
Depto. de Engenharia Elétrica (ENE) - FT
Universidade de Brasília (UnB)
Campus Darcy Ribeiro
CEP: 70919-970 - Brasília-DF - Brasil

Dedico este trabalhos aos meus bisavós Sr. Floriano de Souza e Srª. Rosalina Alves que além de serem luz em minha vida, são e sempre serão minha fonte de inspiração.

AGRADECIMENTOS

Em larga escala dos sonhos da população do interior, um dos mais cobiçados é a formação profissional, pois em muitos casos, salvaguarda o esporte, a profissionalização gera o recurso para que o restante da escala dos sonhos possam se tornar realidade. O trajeto e a missão concluída é o orgulho de que a cada dia que se passou, que se passa e que se passará, você será melhor, será um contribuinte ou o realizador dos sonhos dos seus, dos que estavam ali lutando por cada grama de suor que derramastes para esta conquista. Imaginava que um documento que continham algumas muitas páginas iria representar muitas coisas na minha vida, mas não imaginava que representaria tanto na vida de tantas pessoas, como a meus amados bisavós, que acreditaram desde os primeiros meses que eu poderia alavancar uma carreira de sucesso, sem sequer coitados, poderem imaginar o futuro.

Agradeço imensamente a Deus por esta conquista e juntamente a ele, ao meu amado Tio Antônio, que infelizmente não pode acompanhar esta jornada, nem seu início, nem seu fim. Mas esteve dentro do seu plano espiritual, olhando por todos nós aqui neste plano carnal. Obrigado Tio, por ter sido uma das maiores inspirações deste projeto e por ter em tantas vezes, me guiado nele.

Agradeço aos meus bisavós, seu Floriano, dona Rosa... Estes que foram e sempre serão meus pais, que todos os finais de semana deste semestre, me ligavam ansiosos aguardando que esta data chegasse e chegou. Graças a vocês, sou o mais novo engenheiro da família.

Agradeço aos meus tios, primas e irmãos de forma igualitária, mas em especial as três pessoas que foram base para que eu nunca desistisse: Gil, Raíssa e Rafaella Alves. Quero que com base nesse pequeno trecho, entendam que sempre serão imortais e essenciais para que agora esse engenheiro possa desempenhar bem seu papel profissional.

Agradeço aos meus pais, que da sua forma mais generosa, pode me acolher e me fortalecer pra que eu pudesse ter a oportunidade de me graduar em uma Universidade Federal. Vocês colaboraram incansavelmente e da sua maneira, sem julgamentos, retalhações ou prioridade, para que eu chegasse onde imaginavam que eu seria capaz.

Agradeço aos meus amigos, colegas de vida e de profissão. Cito todos em um, pois cada qual sabe exatamente todo o papel fundamental que desenvolveram na minha vida. Alcides e Wesley, amigos de vida, irmãos de alma. Rosivaldo, Heitor e Felipe Lara, companheiros de jornada e futuramente, de profissão.

E claro, agradeço a companheira que me ensinou a ser forte, a ser persistente e a ter paciência com essa vida acadêmica. Heloísa Vargas, você foi e sempre será fundamental nessa trajetória e espero fortemente poder estar contigo em todos os momentos pré e pós.

Agadeço por fim aos meus professores. Ao Prof. Dr. Flávio Elias, por ter me apoiado e me guiado a este projeto de ser Engenheiro, me dado oportunidade de trabalhar com pesquisas e monitorias. Ao meu orientador Prof. Dr. Georges Daniel, que sempre cobrou muito, mas fez o seu melhor para que todo este sonho pudesse virar realidade.

Se precisasse eu, faria tudo denovo...

RESUMO

A constância de ataques cibernéticos registrados nos últimos nos vem trazendo a tônica a fragilidade dos sistemas operacionais em face da evolução de uma prática muito comum chamada Engenharia Social. Os focos de intrusão em uma rede, que em hipótese alguma se tornou menos importante, vão abrindo espaços a intrusões baseadas em host, tendo em vista que os atacantes vem entendendo que a dinâmica de movimentação lateral atrelada as más práticas de configurações de proteção aplicadas por administradores de usuários, hosts e serviços possibilitam um ataque mais coordenado, ágil e certeiro. Este projeto propõe um estudo de como um atacante se comporta a partir do momento em que já se encontra em um cenário, tanto quanto como é possível perceber a presença de um invasor interno por meio de um sistema de detecção de intrusão baseado em host (HIDS). Para tal, utilizou-se de metodologia de experimento-teste e/ou PoC (*Proof of Concept*) concentrando se um sistema de ataque de intrusão em hosts multiplataformas e, como solução de detecção, utilizou-se do HIDS OSSEC, um sistema amplamente explorado por vários desenvolvedores de sistemas de detecção e resposta como um gerador de eventos de intusão e modificação de arquivos. Os resultados expõem a reação da ferramenta em face ao cenário proposto por meio da geração de constantes logs de evento

Palavras-chave: Multiplataformas, Ataques de intusão baseado em hosts, Ataques cibernéticos Gestão de incidentes, Mitigação e remediação de ataques, Segurança da informação.

ABSTRACT

The constancy of cyber attacks recorded in recent years has brought to the fore the fragility of operating systems in the face of the evolution of a very common practice called Social Engineering. The focuses of intrusion into a network, which under no circumstances has become less important, open spaces to host-based intrusions, given that attackers have understood that the dynamics of lateral movement tied to the poor practices of protection settings applied by administrators of users, hosts and services enable a more coordinated attack, agile and accurate. This project proposes a study of how an attacker behaves from the moment they are already in a scenario, as much as it is possible to perceive the presence of an internal attacker through a host-based intrusion detection system (HIDS). To this end, we used a test experiment methodology and/or PoC (Proof of Concept) focusing on an intrusion attack system on multiplatform hosts and, as a detection solution, HIDS OSSEC, a system widely exploited by several developers of detection and response systems as a generator of intusion and file modification events, was used. The results expose the tool's reaction to the proposed scenario by generating constant event logs

Keywords: Cross-platform, Host-based intuition attacks, Cyber attacks Incident management, Mitigation and remediation of attacks, Information security.

SUMÁRIO

LISTA DE FIGURAS

LISTA DE TABELAS

LISTA DE ABREVIATURAS E SÍMBOLOS

Siglas

API	*Application Programming Interface*
APT	*Adavanced Persistant Threat*
CIA	*Confidentiality, Integrity and Availability*
CIDR	*Classless Inter-Domain Routing*
CKC	*Cyber Kill Chain*
CPU	*Central Processing Unit*
CVE	*Common Vulnerabilities and Exposure*
DNS	*Domain Name System*
DoS	*Denial of Service*
EDR	*Endpoint Detection and Response*
GDPR	*General Data Protection Regulation*
GLP	*General Public License*
GNU	*GNU's Not Unix*
GPG	*GNU Privacy Guard*
HIDS	*Host-based Intrusion Detection System*
HTTP	*Hypertext transfer protocol*
HTTPS	*Hypertext transfer protocol* sobre TLS
IDS	*Intrusion Detection System*
IPS	*Intrusion Protection System*
IP	*Internet Protocol*
MAC	*Medium Access Control*
NAT	*Network Address Translation*
NIDS	*Network Intrusion Detection System*
POC	*Proof of Concept*
RAM	*Random Access Memory*
RDP	*Remote Desktop Protocol*
RHEL	*Red Hat Enterprise Linux*
SO	Sistema Operacional
SSH	*Secure Shell*
SSL	*Secure Socket Layer*
TCP	*Transport Control Protocol*
TLS	*Transport Layer Security*
UDP	*User Datagram Protocol*
VM	*Virtual machine*

1 INTRODUÇÃO

Com a constante evolução da comunicação em mídias digitais, se tornou comum ler, ouvir e comentar sobre os ataques cibernéticos em torno de todo o Globo Terrestre. Em uma breve pesquisa na [CNN Brasil | Ataque cibernético - Notícias e tudo sobre] é possível verificar notícias relacionadas a investigações de vazamento de dados, ataques cibernéticos internacionais, crescentes riscos de ataques devido a pandemia do novo Coronavirus, apagão de dados em áreas governamentais consideradas vitais, como áreas de Justiça e Saúde, por exemplo.

De acordo com [Machado et al. 2019], estes vazamentos não são oriundos de dias atuais e vem ocorrendo a um período considerável para serem percebidos com muita ênfase somente agora. Por meio deste estudo, é possível se destacar vazamentos de dados desde a aproximadamente 7 anos (vide Tabela 1.1) e em escalas milionárias ou até mesmo bilionárias de informações, como o caso do **Esquema Nacional de Indentificação Aadhaar**.

Tabela 1.1: Os 5 Maiores vazamentos de dados entre 2014 e 2019

Índice	2014	2015	2016	2017	2018	2019
1º	145 milhões	78,8 milhões	5 milhões	145,5 milhões	1,1 bilhão	773 milhões
2º	2,6 milhões	25 milhões	2,2 milhões	5,5 milhões	500 milhões	200 milhões
3º	1,3 milhões	15 milhões	1,5 milhões	2,2 milhões	340 milhões	24 milhões
4º	774 mil	11 milhões	950 mil	1,8 milhões	150 milhões	12 milhões
5º	550 mil	10 milhões	320 mil	1,6 milhões	100 milhões	7,7 milhões

O *Aadhar* é considerado o maior esquema nacional de identidade do mundo. Implementado na Índia pelo governo local, o projeto buscava criar um banco de dados centralizado (localmente conhecido pela sigla UIDAI) com informações como dados biométricos e demográficos dos residentes [Agrawal, Banerjee e Sharma 2017].

Contudo, por meios desconhecidos, houve um vazamento de dados que impactou aproximadamente 1,1 bilhão de pessoas. Estes dados vazados vão desde os nomes dos residentes indianos como as informações de serviços conectados, conta bancária e documentos de identidade de 12 dígitos [Biggest Data Breaches of 2018].

É interessante trazer para este estudo a interpretação apresentada pela [Alarcão 2021] em seu trabalho, a qual destaca que, com o avanço da facilidade de conexões através das redes e da internet, o problema de ameaças persistentes ou APTs (*Advanced Persistent Threats*) e a tática de movimento lateral que se confunde com o tráfego normal da rede são os principais ocasionadores de vazamento e perda de dados sensíveis, o que pode ser entendido através do estudo de seus conceitos.

Os movimentos laterais possibilitam aos atacantes a manutenção do acesso bem como o aumento dos privilégios de execução e atuação naqueles sistemas. Neste ponto há a atuação dos ATPs, haja vista que um movimento lateral de um hacker em um sistema crítico é uma ameaça:

- **avançada** devido a seus ataques de dia zero (Uma arquitetura *Zero-Trust* é um conceito de cibersegurança corporativa que é baseada em princípios de confiança a nível zero e justamente projetada para que sejam evitadas violações de dados e limitar o movimento lateral interno);

- **persistentes** devido ao nível de interesse da organização criminosa em face das etapas alcançadas.

É de suma importância o entendimento de que o movimento lateral é apenas um ciclo de uma APT, tendo esta um *Lifecycle* de incríveis doze etapas agrupadas em três estágios, sendo eles a **infiltração**, a **expansão** e a **execução**.

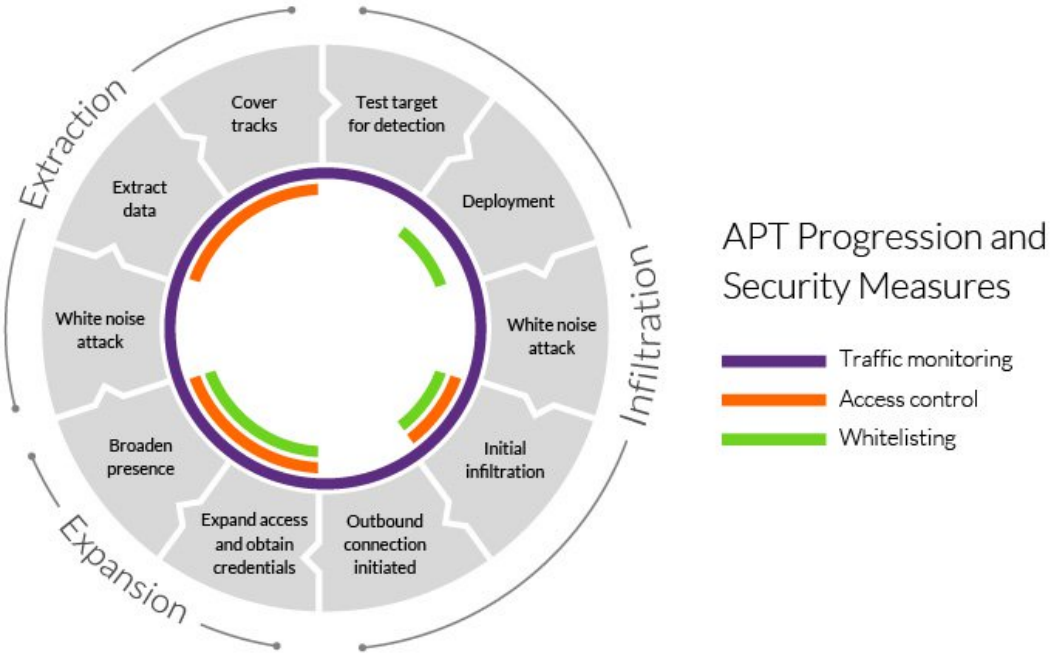

Figura 1.1: Gráfico a qual é possível observar as 12 etapas e os 3 estágios de um APT que possibilita a inclusão do movimento lateral. Fonte: [APT: Advanced persistent threats 2020]

1.1 PROBLEMÁTICA

Toda infraestrutura atualmente, seja ela educacional ou corporativa, está diretamente conectada a internet sendo que, no "melhor dos mundos", se encontra minimamente protegida. Esta mínima proteção pode ser, por exemplo, um *Firewall* de camada 2 ou um controle de acesso físico e lógico ou ainda simplesmente um Antivírus em um dispositivo final.

Entretanto, as ameaças estão cada vez mais capacitadas, consolidadas e silenciosas, se alojando e atuando em muitos caso de forma coerente com padrões de segurança reconhecidos e desenvolvidos para, teoricamente, impedi-las.

Com este cenário, há a necessidade de mudar o conceito de que o investimento em segurança da informação é um gasto desnecessário e adiável e torná-lo de igual importância de uma infraestrutura de rede bem arquitetada, até porque isto permite uma maior confiabilidade de operação no ambiente.

Não somente investimento tecnológico, é necessário investimento social, isto é, capacitar analistas de seguranças a utilizarem ferramentas responsivas ao mais baixo nível da comunicação, possibilitando assim

a criação de um corpo eficiente que seja capaz de monitorar e atuar de forma eficaz em um incidente de segurança, reforçando assim as bibliotecas de boas práticas e condutas. [Alarcão 2021]

1.2 OBJETIVOS

1.2.1 OBJETIVOS GERAIS

O objetivo geral deste trabalho é criar um ambiente controlado para estudo, implementação e análise de resultados de um sistema de código aberto que seja capaz de realizar funcionalidades de detecção de intrusão em *endpoints* por meio da realização de ataques coordenados de varredura de portas e quebra forçada de autenticação por meio de portas abertas de forma devida ou indevida, tendo em vista a finalidade de cada dispositivo para a composição do cenário proposto.

As análises e resultados serão coletados da ferramenta em questão e comparadas com trabalhos correlatos, que tiveram como fim a análise de vulnerabilidades e detecção de intrusões utilizando sistemas de código aberto.

1.2.2 OBJETIVOS ESPECÍFICOS

Os objetivos específicos tem como teor dar sentido ao objetivo geral deste trabalho. Todos os requisitos, processos e descrições a seguir visão a implementação, coleta e análise de resultados comportamentais do sistema HIDS (*Host-based Intrusion Detection System*) OSSEC, de código aberto e disponível em https://www.ossec.net/download-ossec/. Desta forma, objetiva-se:

1.2.2.1 Dar continuidade ao estudo elaborado pela [Alarcão 2021] e implementar um ambiente controlado de testes local para Sistemas de Detecção de Intrusão baseados em *Host*, aplicando neste ambiente os ferramentais de segurança que provenham a satisfação a qual esta POC (*Proof of Concept*) está destinada;

1.2.2.2 Realizar a abertura de portas estratégicas de forma devida ou indevida, que permitam um acesso não autorizado de modo a instigar uma intrusão em prol da análise comportamental da ferramenta proposta;

1.2.2.3 Realizar ataques coordenados no ambiente minimamente seguro.

1.2.2.3.1 Ataques de *Port Scanning* via NMAP;

1.2.2.3.2 Ataques de intrusão via porta SSH;

1.2.2.3.3 Exposição de *Kernel-Mode Rootkit*.

1.2.2.4 Configurar, preparar e acionar, de forma apurada, o software HIDS OSSEC para que o mesmo detecte os ataques propostos;

1.2.2.5 Analisar o desempenho dos resultados obtidos pelo HIDS OSSEC e fazer um estudo comparativo de eficiência e eficácia com o estudo feito pela [Alarcão 2021] a respeito do EDR Wazuh, verifi-

cando se seguem os mesmos padrões de análise e detecção de intrusões, penetrações e ameaças ou se há divergências de funcionamento e quais divergências, dado que a solução estudada é um HIDS em face de um EDR (*Endpoint Detection and Response*).

1.3 ESTRUTURA DOCUMENTAL

Este documento visa o estudo, implementação e análise de resultados de ferramenta de detecção de intrusão para proteção de *endpoints* em ambiente controlado e se encontra segmentado em sete capítulos sendo que:

- O primeiro capítulo se refere a introdução, que tem cunho principal de realizar uma abordagem descritiva das motivações, primeiros estudos e conceitos e objetivos do trabalho;
- O segundo capítulo se refere a fundamentação teórica e tem cunho principal realizar um estudo mais a fundo bem como relacionar os trabalhos correlatos que serviram como base teórica necessária para toda a concepção deste estudo;
- O terceiro capítulo é dedicado a descrição das ferramentas utilizadas para a produção da PoC, detalhando cada serviço implementado juntamente com suas funcionalidades e configurações de operação e ajuste técnico;
- O quarto capítulo retrata a arquitetura do projeto, o desenho topológico físico e lógico e a ideia por traz dos serviços e tecnologias implementadas;
- O quinto capítulo apresenta os resultado obtidos após a aplicação dos ataques coordenados no ambiente configurado de forma detectiva como descrito na seção 1.2.2;
- O sexto capítulo apresenta uma análise conclusiva sobre os resultados comportamentais obtidos após a aplicação dos ataques coordenados descritos pela seção 1.2.2, de modo a fundamentar este estudo inicial;
- O sétimo e último capítulo visa apresentar sugestões de trabalhos futuros relacionados a este estudo, mas não sendo este o limitante para novas contribuições.
- Este projeto conta, além dos sete capítulos apresentados, quatro anexos que detalham o processo de instalação das ferramentas utilizadas.

2 FUNDAMENTAÇÃO TEÓRICA

A descrição deste capítulo tem como foco nortear toda a teoria das tecnologias adjacentes a proposta que serão abordadas em todo este projeto. Portanto, visa por meio deste realizar um revisão gradual de conceitos, termologias e fundamentos teóricos que serão necessários à execução de todo escopo da solução proposta.

2.1 EDUCAÇÃO EM SEGURANÇA DA INFORMAÇÃO

De acordo com [Amankwa, Loock e Kritzinger 2014] e autores citados, a segurança da informação se trata da **prevenção de acesso, uso, divulgação, interrupção, modificação, fiscalização, gravação ou destruição de sistemas de informação e informação não autorizados**.

Tópicos como prevenção, fiscalização e destruição são altamente comuns em meios de discurso de proteção de dados e instrução, sendo eles grande enfoque da educação cibernética. Em um mundo altamente globalizado, a Engenharia Social se torna um fator primordial no que se refere a responsabilidade de ataques cibernéticos.

A educação em segurança do trabalho deve ter atributos essenciais para sua boa implementação, isto é:

- deve ter foco no desenvolvimento da capacidade e da visão das pessoas para realizar atividades multidisciplinares complexas e habilidades necessárias para prover a segurança da informação;
- deve ter propósito em acompanhar as ameaças e mudanças tecnológicas;
- Deve ser um conjunto de controles técnicos e gerenciais para garantir a confidencialidade, integridade, autenticidade, disponibilidade e utilidade dos sistemas de informação e da informação.

Apesar das definições apresentadas terem foco e/ou propósito, a educação da segurança da informação é ausente. Isto se dá muito pela ineficácia em se promover métodos de abordagens de Engenharia Social em muitos âmbitos da segurança da informação. Se torna amplamente necessário que cada profissional que esteja direta ou indiretamente ligado a informações sensíveis tenha conhecimento de normas e documentos de segurança da informação, pois estes fornecem informações e compreensão do cenário quando impactado por um vazamento de dados.

Assim sendo, a segurança da informação é composta por , regulamentações (como a ISO 27001), princípios (como a tríade CIA), e tecnologias lógicas de comunicação segura (protocolos, criptografia e sistemas de monitoramento, detecção e reposta), sendo estes componentes elementares da segurança da informação e tópicos necessários para uma "boa"Engenharia Social em qualquer âmbito onde haja compartilhamento, armazenamento e tratamento de dados sensíveis.

2.2 REGULAMENTAÇÕES: ISO 27001

A ISO/IEC 27001 é o padrão mais utilizado no campo de segurança da informação, sendo usado por organizações que gerenciam informações em nome de terceiros e é aplicado para garantir a proteção de informações críticas do cliente.

De acordo com [ITGovernance 2019], as duas principais razões para o crescente interesse pela adequação à ISO 27001 são as proliferações de ameaças à informação bem como a crescente gama de requisitos regulatórios e estatutários relacionados à proteção das informações, como o caso atual da Lei Geral de Proteção de Dados.

As ameaças à segurança da informação são de natureza global e visam indiscriminadamente todas as organizações e indivíduos que possuem ou usam (principalmente) informações eletrônicas. Essas ameaças são automatizadas e soltas na Internet. Os dados também estão expostos a muitos outros perigos, como atos da natureza, ataque externo e corrupção interna e roubo.

Nos últimos anos, houve o surgimento de um corpo crescente de legislação e regulação em torno da segurança da informação e dos dados e, algumas dessas regulamentações se concentram na proteção de dados individuais, enquanto outras visam sistemas financeiros, operacionais e de gestão de riscos corporativos.

Um sistema formal de gestão de segurança da informação que fornece orientação para a implantação de melhores práticas é cada vez mais visto como uma necessidade em termos de conformidade e a ISO 27001 é cada vez mais exigida de organizações (e governos) antes que estejam envolvidos em quaisquer transações comerciais significativas.

2.3 PRINCÍPIOS - TRÍADE CIA

Normalmente, de acordo com [Poulsen et al. 2005], os objetivos gerais (princípios) de segurança dos sistemas genéricos de TI são especificados como o tríade de confidencialidade, integridade e disponibilidade (inglês *availability*, deste modo, CIA) conforme definido na ISO-17799, sendo que estes princípios podem descrever os objetivos de um sistema de controle de processos.

Tradicionalmente e neste quesito, antigamente, a disponibilidade e a integridade destes sistemas eram os principais princípios a serem seguidos haja vista que, não tendo muitas informações para se proteger e apenas os tendo como detentores de acesso os operadores imediatos, não se fazia muito sentido garantir a confidencialidade dos dados. Todavia, com os sistemas mais robustos, sendo eles sistemas de gerenciamento de informação que potencialmente disponibilizam essas informações para as pessoas a fora – assim como sobre – as instalações da organização, a confidencialidade pode se tornar relevante.

Como já abordado pela [Alarcão 2021], em cada ataque, vulnerabilidade ou ameaça que surge, é sempre possível identificar qual ou quais desses três princípios foram violados. Desta forma, se faz necessário o entendimento de cada conceito defendido pela tríade, portanto:

- **Confidencialidade:** A confidencialidade é diretamente ligada a privacidade e a exclusividade da informação, se tornando um fator primordial para a proteção da informação (isto é, a qualidade do que

é confidencial). Atualmente, além dos ataques *man-in-the-middle* (que de acordo com [Mallik 2019], é um ataque onde o atacante se transfere furtivamente e muda a correspondência entre duas partes que confiam que estão se comunicando diretamente entre si), a Engenharia Social vem sendo um grande fator de preocupação para geração de confidencialidade. Sendo a Engenharia Social, descrita por [Rosa 2012], como a aplicação de conhecimentos empíricos e científicos de um modo sociável de acordo com as necessidades humanas para obter informações confidenciais, esta se torna algoz de toda o investimento em busca da confidencialidade do projeto haja vista que é limitada a ação humana. As ações protetoras que visam promover a confidencialidade partem desde a redundância da ação humana segura, como a participação de dois agentes para atuação em um sistema a qual ambos autentiquem de forma unilateral e criptografada até a criptografia em tunelamento para transmissão de informações e protocolos;

- **Integridade:** A integridade é correlata a manutenção da originalidade de determinado arquivo, processo ou projeto, isto é, é o ato de preservar a autenticidade de tal maneira a se gerar confiabilidade. Ataques de intrusão à raiz do sistema ou a um código-fonte de uma aplicação com propósito de danificá-lo ao ponto de causar indisponibilidade ou até mesmo uma arquitetura de movimentação lateral é o grande exemplo de uma violação de integridade. Não somente, há o claro exemplo citado pela [Alarcão 2021], a qual se aplica uma ação de ocultação de eventos sistêmicos ao se alterar registros de logs e/ou modificação de sistemas de registros de eventos, monitores de alterações, HIDS's e afins. As ações protetoras que visam promover a integridade partem da adoção de medidas de detecção de intrusão, como é o foco deste projeto até a integração com um sistema responsivo que permita a correção e bloqueio de vulnerabilidades e/ou conexões indevidas e não homologadas pelo sistema.

- **Disponibilidade:** Como apresentado por [Santos 2014], a disponibilidade é o ato de garantir que usuários **autorizados** obtenham acesso à informação e aos ativos correspondentes sempre que necessário. A disponibilidade esperada de um sistema de controle deve ser descrita, uma vez que pode realmente ser medida. Uma vez que pode ser medido, também pode ser gerenciado. Quando os sistemas de controle se tornam diretamente ligados aos objetivos de "negócios", um nível acordado de disponibilidade, por exemplo, deve ser fornecido a um cliente, e esse contrato eventualmente será formalizado em um contrato de nível de serviço, mesmo dentro das organizações [Poulsen et al. 2005]. Os ataques mais comuns para negação de um serviços está implícito em seu próprio nome. Como descrito por [Shurman, Khrais e Yateem 2020], os ataques de negação de serviço (distribuídos (DDoS: *Distributed Denial-of-Service*) ou não distribuídos (DoS: *Denial-of-Service*)) visam interromper serviços de organizações ou de usuários específicos em troca de uma quantidade de dinheiro ou recompensa específica. Existem muitos tipos de ataques DDoS onde a identidade do invasor permanece oculta usando componentes legítimos de terceiros, como o caso do DDoS baseado em reflexão, em que os invasores definem o endereço IP da vítima como uma fonte IP de destino desejável e pacotes de transferência para servidores refletores para dominar a vítima com pacotes de resposta. As ações protetoras que visam promover a disponibilidade vão desde a configuração de um sistema altamente disponível, com a realização de backups e redundância de recursos [Alarcão 2021] desde a implementação de sistemas de detecção de negação de serviço e correlatos.

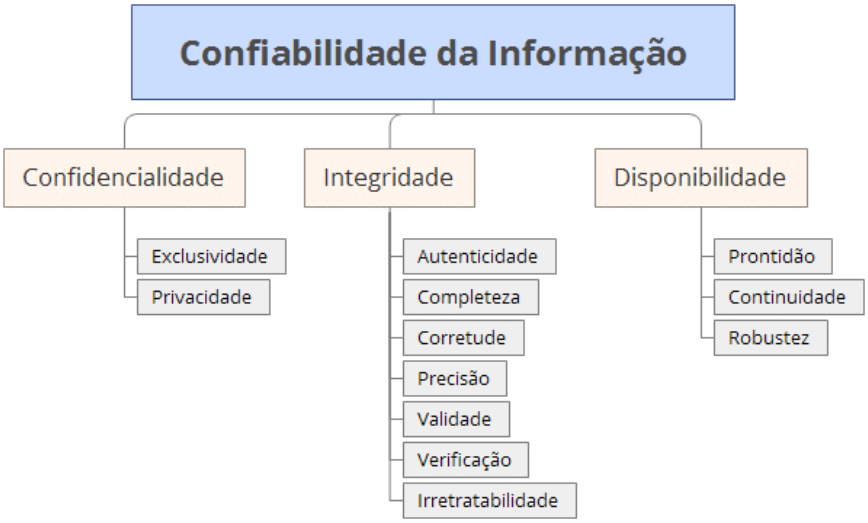

Figura 2.1: Confiabilidade da Informação: TRÍADE CIA - *Confiabilidade, Integridade e Disponibilidade* - e pilares de cada princípio. Fonte: [Freund, Sembay e Macedo 2019]

2.4 MODELAGEM *CYBER KILL CHAIN*

Desenvolvida em 2011 pelos analistas Eric M. Hutchins, Michael J. Clopptert e Rohan M. Amin da Lockheed Martin e baseado no modelo de sistema de defesa cibernética militar, a modelagem *Cyber Kill Chain* tem como principal objetivo auxiliar na tomada de decisão de analistas de cibersegurança nas melhores ações em resposta a invasões adversárias [Assante e Lee 2015].

Originalmente, a modelagem CKC era entendida como uma modelagem de intrusão que descrevia fases de invasão a redes mas que, com o alavancar dos processos coordenados de ataque, se atualizou e começou a definir passos para ataques cibernéticos como um todo, sendo a importância do uso desta dado pela sua capacidade de realizar um mapeamento da estrutura de uma intrusão, isto é, a modelagem auxilia nas tomadas de ações apropriadas pois atua de forma diária no planejamento de uma defesa cibernética, que pode ser guiada, por exemplo, pela matriz do Curso da Ação disponibilizado pela Lockheed Martin (*Lockheed Martin Couse of Action (CoA)*) [Dargahi et al. 2019].

Contudo, antes de entendermos o funcionamento dessa matriz, se faz importante entender as fases de operação da modelagem CKC, que é segmentada em 7 passos, apresentados a seguir:

- **Fase 1 - Reconhecimento (*Reconnaissance*):** Etapa na qual os ataques são direcionados ao reconhecimento do cenário a qual se deseja realizar a intrusão, realizando coleta de informações necessárias para formalização de um ataque robusto bem como para tomada de decisão de qual ferramenta de ataque utilizar. Durante esta etapa, é comum que se tente coletar informações como portas abertas e vulnerabilidades de aplicações e serviços, não sendo estas as únicas informações possíveis de se apartar.
- **Fase 2 - Armamento (*Weaponization*):** Etapa na qual o atacante disfarça sua técnica maliciosa com

meios seguros para que ele consiga se adentrar ao alvo, isto é, para que acessem o ambiente de forma segura, disfarçada e que não desperte um alarde na rede. Com este processo objetiva-se, naquele *host*, realizar abertura de vulnerabilidades locais, como abertura de acesso remoto, desativação de proteções básicas de segurança (antivírus, *Endpoint Detection*, HIDSs entre outros).

- **Fase 3 - Entrega (*Delivery*):** Etapa na qual o invasor encontra meios de implementar a carga maliciosa a qual ele desenvolveu na etapa anterior que, independente de quão sofisticada seja, de algum modo ela tem que adentrar ao ambiente. Como busca uma entrada de modo sutil aos olhos dos analistas de cibersegurança, os invasores buscam técnicas que não despertem um alarde na rede, como unidades USBs, e-mail *phishing*, *spam* dentre outros.

- **Fase 4 - Exploração (*Exploitation*):** Etapa na qual o invasor executa sua carga de *malware* destinada ao host na etapa anterior, isto é, a aplicação do que já vinha sendo trabalhado pelas etapas de reconhecimento, armamento e entrega. Um cenário desta etapa por exemplo, é quando na etapa de reconhecimento se encontra uma vulnerabilidade de sistema que possa ser explorada, na etapa de armamento se desenvolve uma técnica de se explorar esta vulnerabilidade e com base nisto, se blinda esta técnica de recursos de detecção rápida e pela etapa de entrega, se adentra ao cenário, a rede ou a host a qual aquela vulnerabilidade se encontra.

- **Fase 5 - Instalação (*Instalation*):** Etapa na qual o invasor busca a expansão do ataque a outros nós da rede, fazendo com que assim se descubra novos conteúdos a qual ele possa utilizar a seu favor. Nesta etapa é comumente utilizado ferramentas de administração remotas, *Tojans* ou *Backdoors* para essas ações [Dargahi et al. 2019].

- **Fase 6 - Comando e Controle (*Command and Control* (C2)):** Etapa na qual é estabelecido o comando e o controle da máquina atacada. Nesta etapa, o atacante já possui ferramental suficiente para filtrar dados do ambiente de destino final, tendo poder de ação tal qual administrador para poder extrair, criptografar ou qualquer que seja seu objetivo dentro daquele cenário.

- **Fase 7 - Ações e Objetivos (*Actions and Objectives*):** Após a instalação bem sucedida do *malware*, a posse de controle de acesso remoto e todo cenário sob seu controle, é chegada a hora de por em prática seus mais variados anseios. Neste momento, o atacante concretiza sua ação, seja ela qual for em prol dos seus objetivos iniciados na Fase 1.

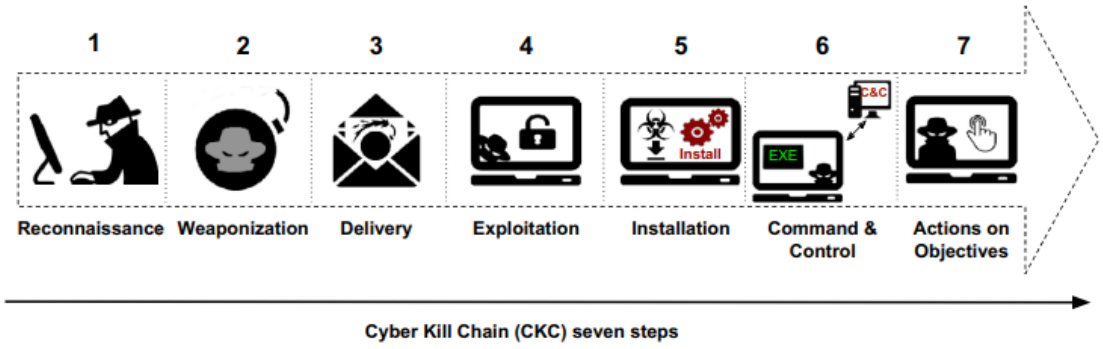

Figura 2.2: 7 fases da modelagem *Cyber Kil Chain* por Lockheed Martin. Fonte: [Dargahi et al. 2019] (Adaptado)

A CoA (*Courses of Action*), citada anteriormente, nada mais é do que uma matriz de instruções de ações inteligentes contra cada uma das fazes acima, isto é, ao mapear um possível movimento malicioso, o analista de segurança pode decidir, para cada fase do ataque, que ação irá tomar, tendo dentre elas a **Detecção** (*Detect*), a **Negação** (*Deny*), a **Interrupção** (*Disrupt*), **Degradação** (*Degrade*), **Enganação** (*Deceive*) e a **Destruição** (*Destroy*).

Phase	Detect	Deny	Disrupt	Degrade	Deceive	Destroy
Reconnaissance						
Weaponization						
Delivery						
Exploitation						
Installation						
C&C						
Actions on Objectives						

Figura 2.3: 6D's - Matriz do Curso de Ação segundo a Lockheed Martin . Fonte: [Dargahi et al. 2019]

2.5 SISTEMA DE DETECÇÃO DE INTRUSÃO BASEADO EM *HOST* (HIDS)

Como apresentado por [Vokorokos e Baláž 2010], o objetivo de uma detecção de intrusão é identificar todas as tentativas de intrusão corretamente e reconhecer atividades que não devem ser marcadas como intrusão. Nesse contexto, entende-se que a intrusão é uma violação de integridade ou acessibilidade de recursos, o que significa violação da política de segurança. Sistemas que detectam tal atividade são apresentados como sistema de detecção de intrusões – IDS (*Intrusion Detection System*).

Quando o IDS utiliza recursos ou dados de sistema de computador e rede para uma resolução subsequente sobre a existência ou processo de ataque, ele é considerado um HIDS (*Host-based Intrusion Detection System*).

[Denning 1987] cita duas abordagens conceituais para análise de intrusões que se trata da **Detecção por Padrão** e da **Detecção baseado em comportamento anormal**. Para cada um destas abordagens, há uma explicação lógica por trás, apresentada a seguir:

- **Detecção por Padrão:** De acordo com [Vokorokos e Baláž 2010], na detecção por padrão, são inferidos os padrões comportamentais do sistema com relação a ataques conhecidos. Este método utiliza-se da avaliação de quadros bem como dados neles encontrados pelo qual se torna possível determinar um comportamento de um possível ataque. Não somente, é possível detectar padrões de comportamento do sistema correspondentes a ataques conhecidos de trilhas de auditoria, logs (registros de status) ou alterações no sistema atacado.

- **Detecção baseado em comportamento anormal:** Conforme abordado por [Bace 2000], quando se trata de uma detecção baseada em anomalia, isto é, em um comportamento anormal, cada comportamento anormal atrelado ao comportamento esperado pelo sistema é identificada (este comportamento esperado pelo sistema é predeterminado de forma manual ou automática, a depender de cada usuário/sistema). Ressalta-se que, em caso de elaboração automática de comportamento, é utilizado um sistema de monitoramento de perfil que coleta e processa os dados que caracterizam o comportamento do sistema a partir de um dado tempo, formando assim um padrão comportamental estatisticamente admissível posteriormente. No entanto, a abordagem estatística tem suas desvantagens, como por exemplo quando um invasor decifra e aprende o comportamento dos usuários elegíveis (infectados ou com permissões laterais), o que pode levar a um ataque não detectado. Eventualmente, se este usuário elegível mudar seu comportamento, ocorrerá um possível falso-positivo.

Por fim, conforme salientado por [Vokorokos e Baláž 2010], uma das principais questões de um projeto HIDS é a seleção de prioridade. O mesmo elenca alguns fatores que podem se considerados e verificados pelo HIDS, como por exemplo:

- Mecanismo de disco;
- Utilização dos recursos de Hardware (processador, memória e disco rígido);
- Integridade de Dados;
- Utilização da bateria;
- Velocidade média de transferência de dados na rede;
- Tamanho médio de quadros na rede;
- Duração de conectividade externa.

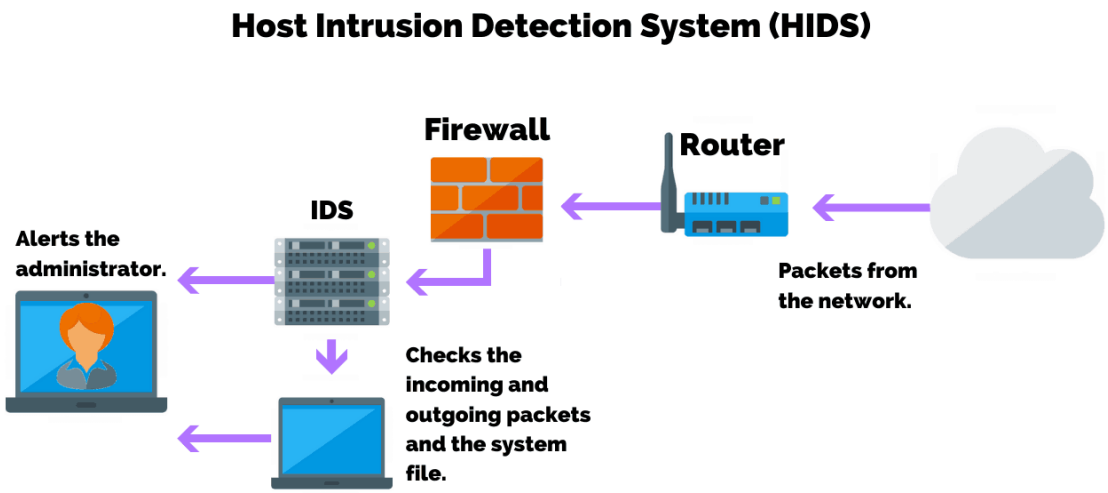

Figura 2.4: Exemplo de topologia de funcionamento de um HIDS - Fonte:[Swanagan 2021]

2.6 VULNERABILIDADES E ATAQUES DE INTRUSÃO

De acordo com a CERT.br, o número de incidentes reportados de 2014 a 2020 no Brasil não foi menor do que 647 mil incidentes. Em um cálculo rápido, temos que, no melhor cenário em 7 anos, o Brasil sofria aproximadamente 1,8 mil ataques por dia, ou seja, aproximadamente 74 ataques por hora. Isto é mais que suficiente para verificar que a todo minuto alguém em nosso continente está sendo atacado e que estes ataques são oriundos de vulnerabilidades sistêmicas.

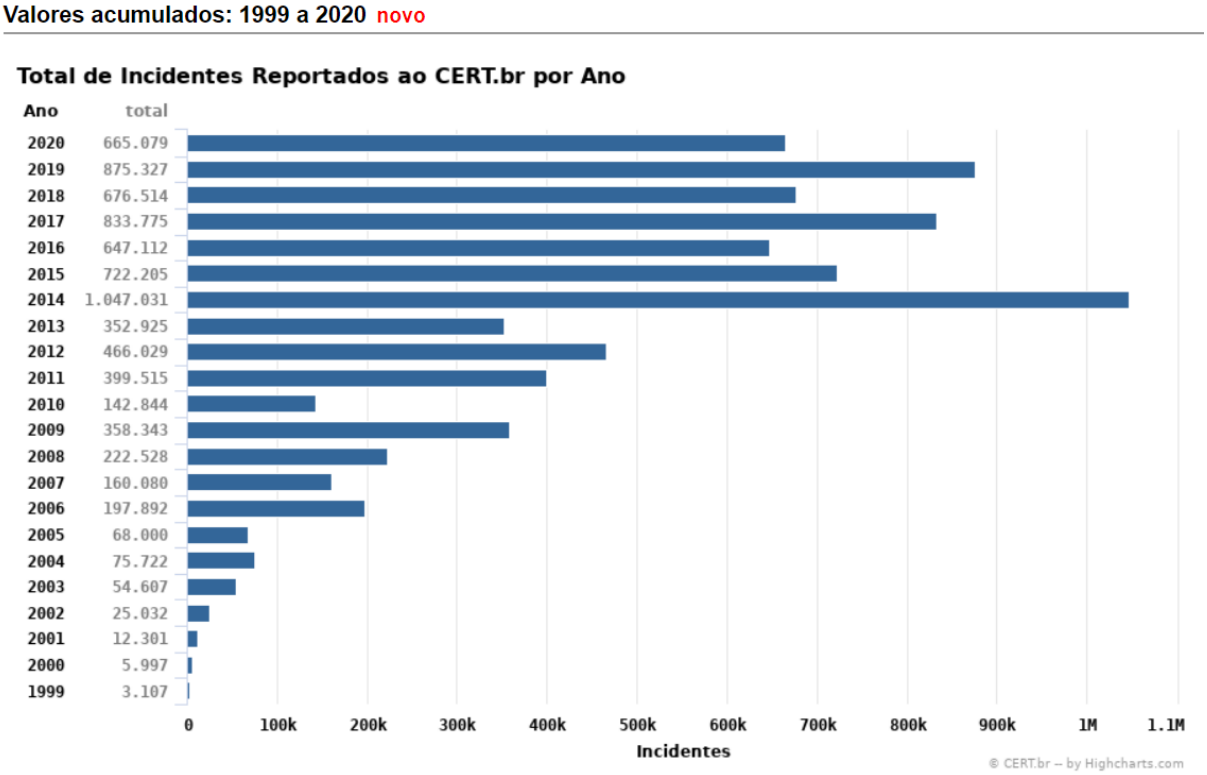

Figura 2.5: Total de Incidentes reportados ao CERT.br por Ano desde 1999 - Fonte: CERT.br

Como citado por [Ferreira et al. 2012], Vulnerabilidade é o mal comportamento (falha ou bug) apresentado pela segurança dos sistema, aplicação ou correlato que permite a um usuário e/ou ferramenta mal intencionada deturpar o comportamento esperado deste elemento, sujeitando assim o sistema a diversos problemas que afetam a sua disponibilidade, integridade e confidencialidade.

Estas vulnerabilidades geram um ato de *Exploit*, que definido também por [Ferreira et al. 2012], se refere ao ato ou ao desenvolvimento ferramental para se tomar vantagem de uma vulnerabilidade presente e/ou potencialmente detectável em um dado sistema.

Muitos dos eventos de ataques cibernéticos iniciam-se com uma varredura de portas e vulnerabilidades, isto é, os atacantes tem por padrão analisar o cenário que irá atacar para que assim possa ser definido a melhor estratégia para a progressão do mesmo, sendo que uma das formas mais conhecidas da consumação destes ataques se dá pela **intrusão**.

Uma intrusão pode ser definida como uma ação ou um conjunto de ações que visam violar uma ou mais políticas de segurança, afetando principalmente a integridade do host, confidencialidade dos dados e

a disponibilidade de serviços, explorando as vulnerabilidades no procedimento de segurança e na implementação do sistema monitorado por um sistema de intrusão. [Anwar et al. 2017].

Entende-se assim que a manutenção de um sistema de detecção de intrusão é, além de necessário, fundamental para o bom desempenho de um ou vários sistemas e serviços. Apesar de, um sistema de detecção de intrusão não apresentar resultados suficientes para uma proteção completa, assim como as demais ferramentas de segurança, quando combinados com boas políticas e uma Engenharia Social alinhado à boas práticas, fornece um desempenho satisfatório.

2.7 SISTEMA OPERACIONAL MICROSOFT WINDOWS

Em Meados de 1981, a Microsoft identificou a necessidade de desenvolver uma interface gráfica (ou na época, um gerenciador de interface) e baseou-se no conceito de padrão de janelas (daí a ideia do nome *Windows*, que em tradução livre, significa Janelas) que exibem informações e recebem respostas dos utilizadores através de um teclado ou de cliques do mouse. [Silveira]

Ainda de acordo com [Silveira], o Microsoft Windows só foi verdadeiramente reconhecido e chamado com Sistema Operacional em 1983. Isso pelo fato de que o que se havia antes na construção do "Windows"eram sistemas gráficos que eram executados sobre alguma versão dos sistemas compatíveis com DOS (*Disk Operating System*), como MS-DOS (*MicroSoft Disk Operating System*), PC-DOS (*Personal Computer Disk Operating System* ou DR-DOS (*Digital Research's Disk Operating System*). Como perceptível, apenas o MS-DOS era um sistema produzido pela própria Microsoft.

Desde 1983, o Windows começou a produzir versões de seus sistemas com um Sistema Operacional. As curiosidades (embasando-se no estudo feito por [Silveira]) das principais versões estão listadas a seguir:

- **Windows 3.x:** Lançado em maio de 1990, foi o primeiro sistema operacional robusto lançado pela Microsoft que resultou em um amplo sucesso. Apesar de um sistema gráfico de 16 bits, ainda era necessário ativar o MS-DOS para assim iniciar o Windows. Dentre as principais curiosidades desta versão, a principal delas foi quebra da barreira de disco do MS-DOS de 1MB em aplicações. Este sistema possibilitou a utilização de 16MB em aplicações, sendo naquela época também o único possível de compatibilizar todos os programas das versões anteriores.

- **Windows 95:** Lançado em agosto de 1995, foi o primeiro sistema operacional de 32 bits. Na época, foi o grande salto da Microsoft dado que em nada se parecia com o seu sistema antecessor. Inseriu ferramentas como o Menu Iniciar e a Barra de Tarefas. Não somente, nesta versão o Windows já ativa-se sem a dependência prévia do MS-DOS tanto quanto as limitações de memória já não mais existiam. Foi adotado como sistema de arquivos o VFAT (FAT-16), possibilitando assim os mesmos a terem 255 caracteres de nome (mais uma extensão de três caracteres que indica o programa que abre o arquivo).

- **Windows 98:** Lançado em junho de 1998, trouxe como principal vantagem a a completa integração do sistema operacional com a Rede Mundial (Internet) via *browser* Internet Explorer 4. Não somente,

introduziu-se o sistema de arquivos FAT-32. Apesar das novidades, apresentou-se como um sistema lento e instável, sendo necessário alguns *patchs* de correção.

- **Windows XP (NT 5.1):** Lançado em outubro de 2001, foi considerado por muitos a melhor versão de sistema operacional da Microsoft para usuários domésticos, tendo seu *End Of Suporte* anunciado pela empresa somente em abril de 2014. Baseou-se no antigo OS/2 da IBM, cujos direitos foram comprados pela Microsoft, e, seguindo a linha OS/2-NT-2000-XP, a partir deste Windows, surgiu uma nova interface, abandonando o antigo formato 3D acinzentado. Com isso, apresentou melhorias em relação a velocidade, em especial em sua inicialização. Contudo, pecou, na época, em seu alto consumo da qual só pode ser instalado em sistemas com capacidade maior que 128MB de memória.

- **Windows Vista (NT 6.0):** Lançado em janeiro de 2007, foi um sistema duramente criticado pela sua instabilidade, sendo um dos sistemas com mais rápida descontinuação pela Microsoft (abril de 2017) dentre os lançados na era 2000, com menos de 10 anos de suporte. Em seu lançamento, apresentou seis versões, indo do a *Starter Edition* ao *Ultimate Edition*.

- **Windows 7:** Lançado em outubro de 2009, o Windows 7 se tornou um dos sistemas de maior sucesso para a Microsoft, tendo sido descontinuado somente apenas em janeiro de 2020. O sistema inovou em design e nos recursos e ferramentas, trazendo maior compatibilidade com ferramentas e aplicações. É um sistema de 32 ou 64bits. Em comparação ao seu antecessor, o mesmo recebeu maiores recursos de navegação (*Aero Shake*, *Aero Peek* e *Aero Snaps*). Apesar destes recursos, o Windows 7 não veio, de forma nativa, com editor de imagens, editor de filmes e nem mesmo um cliente de correio. [InfoEscola 2006]

- **Windwos 8:** Lançado em outubro de 2012 e tendo seu *End Of Suporte* em meados de 2016, foi considerada a versão mais abrupta em termos de revolução de design ao retirar o botão iniciar da tela principal (retornado com a versão 8.1 do sistema) e organizar os ícones em blocos. A revisão gráfica foi considerada bem sucedida por muitos usuários, reforçando a ideia da Microsoft em colocar um sistema que fosse funcional para usuários de desktop e de dispositivos *touchscreen*, com uma única interface. [TechTudo 2014]

- **Windows 10:** Lançado em setembro de 2014, marcou o retorno da Microsoft as "origens", trazendo de volta o menu iniciar e mais equilíbrio para usuários de computadores de PCs tradicionais. Apesar do foco em *desktops*, o Windows 10 traz uma arquitetura que propõe unificação entre celulares e *tablets*, tendo consigo uma loja de aplicativos (*Windows Store*) que permite a instalação de aplicativos multiplataformas, acabando com a divisão *Windows Phone* e Windows RT. O Windows 10 será a distribuição utilizada neste projeto para os testes propostos.

2.8 DISTRIBUIÇÕES LINUX

Como citado por [Campos 2006], uma distribuição Linux é *um sistema operacional Unix-like incluindo o kernel Linux e outros softwares de aplicação, formando um conjunto*. As distribuições linux podem ser mantidas por organizações comerciais ou grupos de desenvolvedores a depender da sua finalidade.

Atualmente, as distribuições comerciais apresentam maior estabilidade tendo em vista o fornecimento de constantes atualizações e suporte técnico especializado, com o fornecimento de certificações profissionais e reconhecidas no cenário de desenvolvimento Linux. Este projeto fez uso de duas distribuições Linux comerciais, sendo elas o *CentOS Project* da **Red Hat Enterprise Linux (RHEL)** e o Ubuntu LTS da **Canonical**. Suas peculiaridades e curiosidades serão apresentadas a seguir.

- **Red Hat Enterprise Linux CentOS Project:** O CentOS (abreviação para ***C**ommunity) **ENT**erprise **O**perating **S**ystem* nasceu de um projeto comunitário em meados de 2003 e desde Março de 2004 vem sendo apoiada por uma comunidade de fontes fornecidas pela RHEL a qual promete fornecer a distribuição maior estabilidade, previsibilidade, reprodutibilidade e gerenciabilidade, sendo um sistema de código aberto a nível corporativo a qual se empenha em manter níveis de treinamento, suporte e futuramente certificações.

- **Canonical Ubuntu LTS:** O Ubuntu é um sistema operacional desenvolvido com base no Sistema Debian lançado em Outubro de 2004 pela Canonical, sendo atualmente conhecido como a distribuição mais popular entre usuários Linux. A Canonical, no desenvolvimento da Distribuição Ubuntu LTS adotou o discurso de projetar um sistema sempre gratuito, sempre intuitivo e de fácil utilização e com tecnologias assistidas e traduzidas, sendo este seu principal objetivo em face das limitações de linguagens e acessibilidade que as distribuições Linux na época.

Last 12 months		
1	MX Linux	3277▼
2	EndeavourOS	2953▲
3	Mint	2107▲
4	Manjaro	2099▼
5	Pop!_OS	1518▼
6	Ubuntu	1323–
7	Debian	1228–
8	Garuda	1165–
9	Fedora	1018▲
10	elementary	1004▼
11	Zorin	910▲
12	openSUSE	797▼
13	KDE neon	670▲
14	antiX	602–
15	Slackware	562▼
16	Solus	544▼
17	Lite	537–
18	PCLinuxOS	456–
19	Arch	454▼
20	Kali	436–
21	ArcoLinux	402–
22	Kubuntu	386–
23	Puppy	385–
24	SparkyLinux	380▲
25	FreeBSD	361▼
26	Q4OS	360▲
27	CentOS	359–
28	Artix	358▲
29	Void	325▲
30	Alpine	319▼

Last 6 months		
1	MX Linux	3378▼
2	EndeavourOS	3156▼
3	Mint	2281▲
4	Manjaro	1965▼
5	Pop!_OS	1498▼
6	Ubuntu	1330▼
7	Garuda	1185–
8	Debian	1178▼
9	Fedora	1070▲
10	Zorin	929▼
11	elementary	879▼
12	openSUSE	784–
13	Slackware	666–
14	antiX	654▼
15	KDE neon	640▲
16	Lite	625–
17	Solus	484▼
18	PCLinuxOS	478–
19	Kubuntu	474▲
20	Kali	459–
21	Super Grub2	446▲
22	Arch	419▼
23	ArcoLinux	415–
24	SparkyLinux	405▼
25	FreeBSD	397▲
26	Void	380▲
27	Q4OS	371–
28	CentOS	369–
29	Puppy	369▼
30	Linuxfx	359–

Last 3 months		
1	MX Linux	3085▼
2	EndeavourOS	2849▼
3	Mint	2298▲
4	Manjaro	1781▼
5	Pop!_OS	1407▼
6	Ubuntu	1372▲
7	Debian	1148▼
8	Garuda	1094▲
9	Fedora	1090▲
10	Zorin	929▼
11	openSUSE	803–
12	elementary	776▼
13	Slackware	711▼
14	Lite	649▼
15	KDE neon	606▲
16	antiX	602▼
17	Linuxfx	526–
18	Kali	478–
19	Solus	439–
20	SparkyLinux	431–
21	PCLinuxOS	421▼
22	Void	417▲
23	Kubuntu	408▼
24	ArcoLinux	402–
25	FreeBSD	400▲
26	Arch	395–
27	EasyOS	376▼
28	Puppy	363–
29	Q4OS	361▲
30	Peppermint	359▲

Last 1 month		
1	MX Linux	3060–
2	EndeavourOS	2819▲
3	Mint	2492▼
4	Manjaro	1571▼
5	Ubuntu	1437▲
6	Pop!_OS	1395▼
7	Fedora	1349▲
8	Debian	1227▼
9	Garuda	1065▼
10	Zorin	904▼
11	Void	864▲
12	openSUSE	736▼
13	elementary	699▼
14	KDE neon	625▲
15	FreeBSD	569–
16	antiX	535▼
17	Lite	527▼
18	Alpine	499–
19	Slackware	488▼
20	EasyOS	483▲
21	Q4OS	455–
22	Parrot	437▲
23	Solus	422▼
24	Kali	403▼
25	PCLinuxOS	401▲
26	Kubuntu	383▲
27	Arch	363–
28	Puppy	356–
29	ArcoLinux	353▼
30	SparkyLinux	329▼

Figura 2.6: Relação das 30 distribuições Linux mais utilizadas segmentadas pela listagem do último mês, dos últimos 3 meses, dos últimos 6 meses e do último ano. Fonte: DistroWatch [DistroWatch 2022]

3 FERRAMENTAS UTILIZADAS

A descrição deste capítulo tem como foco detalhar as ferramentas utilizadas para o desenvolvimento deste projeto. Neste, será possível entender o funcionamento e as principais funcionalidades e aplicabilidade com base no cenário de segurança da informação proposto.

3.1 ORACLE VIRTUALBOX

O Oracle Virtual Box é uma ferramenta de virtualização multiplataforma baseada em arquitetura x86 e AMD64/Intel64 disponível como software de código aberto sob os termos da segunda versão da *GNU General Public License (GLP)*.

Na prática, um virtualizador tal como o **Oracle VM VirtualBox** amplia a capacidade de execução de um *host*, independente do Sistema Operacional base (Windows, Mac OS, Linux ou Oracle Solaris), de um para vários Sistemas Operacionais. Isto é possível por meio do que chamamos de instâncias virtuais, que são sistemas iguais ou diferentes que operam "sobre recursos dedicados de um único dispositivo"[Hat 2018] de forma emulada e delimitada à disponibilidade de hardware como espaço em disco, poder de processamento e memória RAM. [VirtualBox 2022]

Os recursos fornecidos pelo Oracle VM VirtualBox podem ser utilizados em vários tipos de contextos e estes podem ser consultados através do link https://www.virtualbox.org/manual/ch01.htmlvirt-why-useful. Para fins deste projeto e dentre os contextos apresentados pelo Manual do Virtual Box citado acima, foram coletados e listados as seguintes funcionalidades:

- **Execução de vários sistemas operacionais de forma simultânea:** Como um virtualizador, o Oracle VM VirtualBox permite a execução de simultâneos sistemas operacionais utilizando-se do hardware hospedeiro da aplicação. Esta virtualização pode ser feita mediante a inserção de imagem de disco (Imagens ISO (extensão de arquivo de imagem obtido através da antiga ISO9660, comumente utilizada para padrões de CD-ROM), por exemplo) ou extensões de discos rígidos virtuais já operacionais. Todo o processo de importação e configuração de uma máquina virtual no Oracle VM VirtualBox pode ser acessado através do link https://www.virtualbox.org/manual/ch01.htmlgui-createvm

- **Testes e recuperação de desastres:** Por este projeto realizar um ataque de intrusão em *hosts* guiado, buscando desta forma testar o comportamento da ferramenta a ser estudada, é de vital importância que seja mantido a possibilidade de recuperação da máquina virtual que será atacada, haja vista será apresentado um cenário real de intrusão o deturpação de arquivos de sistema. No Oracle VM VirtualBox, uma vez que uma máquina virtual foi instalada, ela, juntamente com seus discos virtuais assumem o comportamento de um contêiner que arbitrariamente pode ser pausado, parado, iniciado copiado e transportado entre *hosts*. Este recurso, juntamente com a funcionalidade no mundo da virtualização conhecido como *Snapshots* (funcionalidade em que é permitido salvar um estado

específico de uma máquina virtual a fim de reverter seu estado atual a este estado específico em ocorrência de mau funcionamento) possibilita uma maior segurança para um ambiente controlado.

Desta forma, o Oracle VM VirtualBox se torna a ferramenta base deste projeto, sendo o mantenedor de toda a estrutura virtualizada que permite a operação de instâncias virtuais tais como a VM (*Virtual Machine*) do emulador de software de rede GNS-3, discutido a seguir.

3.2 GRAPHICAL NETWORK SIMULATOR 3

O GNS-3 (*Graphical Network Simulator 3* é um software de código aberto que permite emular, configurar, testar e solucionar problemas de redes reais e virtuais distribuídos em uma pequena topologia que pode ser composta de *hosts*, passivos e ativos de redes, estejam eles hospedados em vários servidores ou mesmo na nuvem.

Originalmente, o GNS-3 "imitava"dispositivos Cisco usando um software denominado Dynamips, mas atualmente o software suporta outros dispositivos de outros fabricantes, que podem ser consultados através do site https://docs.gns3.com/docs/.

O GNS-3 consiste basicamente em dois componentes de software, descritos a seguir:

- **GNS3-all-in-one software (GUI: *Graphical User Interface*):** Consiste na parte cliente do GNS-3 e na interface gráfica de usuário juntas em um arquivo de instalação a ser executado de forma local no *host* hospedeiro (justamente por isso do conceito *all-in-one*). Uma vez instalado, o software permite a criação de topologias de rede, desenho de soluções e afins, tendo como servidor o próprio *host* hospedeiro, i.e., a comunicação cliente-servidor passa a ser entre a interface gráfica de usuário (cliente) e o *host* de a qual o software se encontra instalado (servidor).
- **GNS-3 VM:** O GNS-3 VM é uma Virutal Machine que pode ser utilizada em algum virtualizador (no caso deste projeto, o VirtualBox mas não limitante. O GNS-3 VM oferece suporte a outros virtualizadores que podem ser encontrados através do site https://docs.gns3.com/docs/ a qual servirá como um servidor virtual para a comunicação com a interface gráfica de usuário. Usualmente, esta máquina virtual dispõe de dois adaptadores de rede virtuais, a qual uma é utilizada para estabelecer a comunicação entre cliente-servidor e outra para estabelecer a comunicação com a internet.

Por fim, é importante salientar que o GNS-3 suporta dispositivos emulados e simulados. Os conceitos de dispositivos emulados passam pela limitação do hardware de um dispositivo possibilitando a execução de imagens reais. Um exemplo desta emulação, citado pelo site do GNS-3 Docs é a possibilidade de se importar uma imagem de um determinado Roteador real e físico e executá-lo em um roteador virtual e emulado em GNS-3.

Já os conceitos de dispositivos simulados passam pela simulação dos recursos e funcionalidades de um dispositivo, como por exemplo, um *switch*, dispositivo de camada 2, em que não se executa um sistema operacional real, mas sim, as funcionalidades de um dispositivo simulado e desenvolvido pelo próprio GNS-3.

3.3 FIREWALL PFSENSE

Iniciado em meados de 2004, o Projeto Firewall PfSense foi desenvolvido como uma customização FreeBSD adaptado para utilizar funcionalidades de Firewall e Roteamento da qual é inteiramente gerenciável via GUI Web.

Apesar de oferecer soluções embarcadas via Netgate, o pfSense é uma solução de código aberto focada em instalações simplificadas, porém completas e em Desktops. O PfSense utiliza-se de um único arquivo XML (*e**X**tensible Markup Language*, comumente denominado **config.xml**, sendo este o responsável por armazenar a configuração de todos os serviços disponíveis na máquina pfSense. Já seu desenvolvimento é todo baseado na linguagem PHP (acrônimo recursivo para *Hypertext Preprocessor*) o que facilita a ampliação da base de código atual, melhorando os recursos existentes ou adicionando novos [Patel e Sharma 2017]. Os principais recursos disponíveis pela ferramenta são:

- Roteamento;
- Firewall;
- QoS (*Quality of Service*);
- NAT *Network Address Translation*;
- Balanceamento de Carga;
- VPN (*Virtual Private Network*);
- Redundância, dentre outros.

3.4 HIDS OSSEC

O OSSEC (***O**pen **S**ource **SEC**urity*) é um sistema baseado em software livre e pretende, de acordo com a Atomicorp, sua desenvolvedora, permanecer assim pelos próximos longos anos. Isso concede o direito, com base na Licença Pública Geral GNU da FSF (*Free Software Foundation*) modificá-lo e redistribuí-lo, como foi o caso do Wazuh. [Bray, Cid e Hay 2008]. Não somente, é baseado na arquitetura agente-servidor a qual o agente estabelece comunicação com o servidor mediante chave autenticadora gerada pelo servidor para com base no endereçamento IP do agente.

A partir desta identificação, é feito todo o monitoramento referente a qualquer ato que possa caracterizar uma intrusão. O conceito de agente OSSEC e servidor OSSEC são apresentados a seguir, conforme descrito por [Bray, Cid e Hay 2008]

- **OSSEC SERVER:** O papel do Servidor OSSEC é basicamente coletar todos os alertas das instalações dos Agentes implantados em seu escopo de organização. O servidor, quando integrado ao OSSEC Web Interface, fornece uma visão geral do que está sendo exportado por todas as instalações de agentes implantados na organização. A figura 3.1 detalha como funciona essa recepção.

- **OSSEC AGENT:** O Agente OSSEC permite que seja implantado a segurança e a proteção oferecidas pela OSSEC no *host*, centralizando suas informações e enviando alertas de volta para um único servidor OSSEC. Esta instalação elimina a sobrecarga de registro no agente implantado e garante que os alertas gerados não sejam mantidos no sistema.

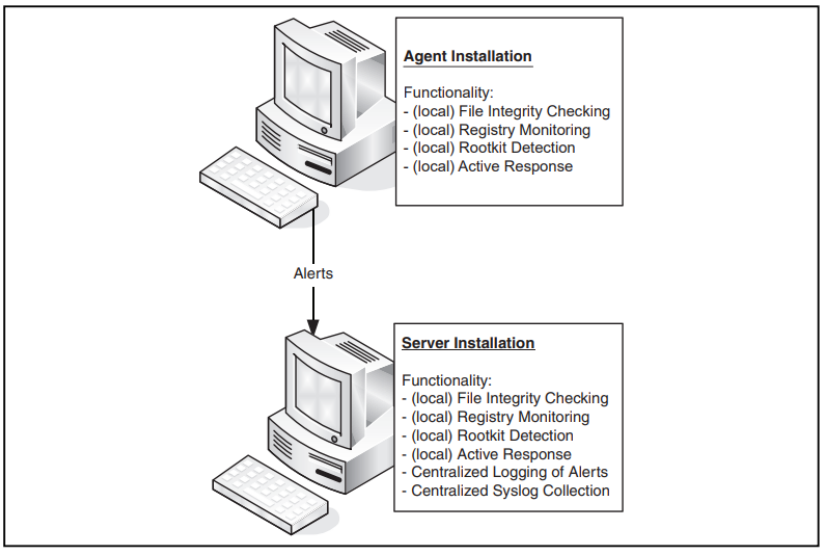

Figura 3.1: Configuração Agente-Servidor do OSSEC, a qual se estabelece o envio por parte do Agente e o recebimento e centralização de logs por parte do Servidor. Fonte: *OSSEC host-based intrusion detection guide* [Bray, Cid e Hay 2008]

A ferramenta OSSEC é um HIDS de código aberto escalável e multiplataforma que conta com uma média de 5.000 downloads por mês, possuindo:

- um mecanismo de correlação e análise;
- uma integração de análise de logs;
- verificação de integridade de arquivos
- monitoramento de registro (no caso de plataformas Windows);
- uma aplicação centralizada de políticas;
- detecção de *rootkit*;
- alerta em tempo real e resposta ativa.

Não somente, o OSSEC pode ser integralizado para operar com funcionalidades a mais do que o um HIDS (funcionalidades de um HIDS foram citadas na seção 2.5), tais como monitoramento e análise comportamental de *firewalls*, IDSs, servidores Web e *Log Authentication*, alem de poder operar simplesmente como um analisador de Logs.

Por ser um sistema multiplataforma, o OSSEC pode ser executado na maioria dos sistemas operacionais disponíveis no mercado, tais como Windows, Mac OS X, SunSolaris, Linux (em várias distribuições como Ubuntu, Debian, CentOS entre outros), OpenBSD, FreeBSD.

3.4.1 Motivação para a escolha do HIDS OSSEC

O HIDS OSSEC foi escolhido como tema central deste projeto baseando-se no projeto apresentado por [Alarcão 2021] a qual refere-se no EDR Wazuh, uma ferramenta de detecção e resposta aprimorada com base no HIDS OSSEC.

O OSSEC foi e ainda é um projeto utilizado como ponto de partida de alguns cenários de ferramentas *Open-Source* de detecção de intrusão e/ou detecção e resposta e, por ter sido desenvolvido e mantido pela **Trend Micro**, umas das principais fabricantes em termos de soluções de detecção e resposta segundo o *Gartner* (*Gartner Peer Insights Trend Micro*), tem o respaldo global de ser uma solução confiável e responsiva para o início de um projeto, seja de aprimoramento de um HIDS ou confecção de um MDR (*Managed Detection and Response*), EDR (*Endpoint Detection and Response*) ou XDR (*Extended Detection and Response*) [Insights 2022].

Objetiva-se com essa escolha entender o que esta ferramenta oferece de agregador a estes projetos e onde a mesma se diferencia de projetos de detecção e resposta, como o caso do EDR Wazuh apresentado por [Alarcão 2021].

Assim sendo, a escolha desta ferramenta se embasa no projeto desenvolvido por [Alarcão 2021] a qual faz uso da mesma de forma interna e melhorada através do EDR Wazuh em que, por meio deste projeto, será possível verificar onde a ferramenta atua e até onde se limita.

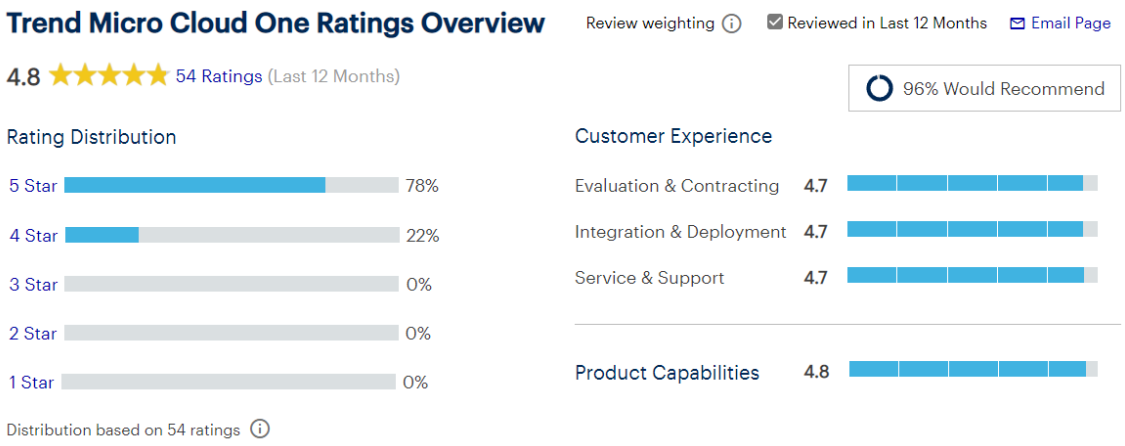

Figura 3.2: Visão geral do *Trend Micro One* segundo *Gartner* nos últimos 12 meses. Esta solução é um exemplo de solução de proteção de dados disponibilizadas para nuvem e contêiner (solução SaaS (*Software as a Service*)) que faz utilização do OSSEC, conforme sua documentação disponibilizada em Cloud One Documentation. Fonte: [Insights 2022]

4 ARQUITETURA PROPOSTA

Este capítulo visa descrever toda a arquitetura proposta bem como a metodologia de ataque utilizada para condução dos experimentos realizados. É possível por meio deste capítulo verificar as topologia lógica e de ataque propostas e desenvolvidas em ambiente controlado, que objetiva relacionar os conceitos de uma detecção de intrusão baseada em host.

Por fim, espera-se que a partir do conteúdo apresentado por este, seja possível o entendimento de como as tecnologias envolvidas na solução proposta podem ser utilizadas tanto quanto como se dá uma construção, por mais sucinta que seja, de um ambiente controlado para testes.

4.1 METODOLOGIA

O Projeto, como citado anteriormente, tem como finalidade estudar, coletar e avaliar eventos de segurança relacionados ao a uma tentativa acesso não autorizado (intrusão) em host, sejam eles a nível de usuário ou a nível de serviços que serão produzidos via ataque interno, isto é, um ambiente controlado já infectado em que o host atacante já se encontra alojado e comunicável na rede, contudo sem um agente OSSEC instalado.

Isto ocorre tendo em vista que o intuito é realizar o teste no host que se encontra sendo atacado e não no host que se encontra atacando e por meio do host atacado, coletar informações essenciais sobre o host atacante que, inicialmente é desconhecido pela rede.

O sistema OSSEC de detecção de intrusão instalado em *hosts* foi a ferramenta avaliada, sendo esta avaliação direcionada a sua capacidade de detecção de eventos de segurança relacionados a intrusão em múltiplos sistemas. Esta propriedade foi avaliada de forma individual e independente, simulando-se ataques selecionados em *hosts* de forma independente, assim como a análise dos resultados desta detecção.

O projeto proposto utiliza técnicas de simulação e emulação baseada e hospedagem local, em que sistemas reais são executados em ambientes virtualizados e ativos de redes são simulados para dar dinamismo ao projeto [GUSTEDT e QUINSON 2009].

O sistema real foi o HIDS OSSEC executado em multiplataformas hospedadas em ferramenta local de emulação e simulação GNS-3, virtualizado pelo sistema de virtualização Oracle VirtualBox. É interessante ressaltar o que este projeto visa uma contribuição ao que já foi apresentado pela [Alarcão 2021] em seu projeto.

Aproveitando-se disto, [Alarcão 2021] cita algumas vantagens em se basear uma pesquisa na emulação, tais como a **segurança**, que proporciona o isolamento a nível de rede da máquina virtual dado que a mesma se encontra isolada em um *host* e virtualizada, permitindo uma maior "tranquilidade"na execução dos ataques e a **Reiteratividade**, que permite a repetição dos testes de forma incansável até que seja satisfatório para validação.

Neste projeto, o atacante posicionado internamente irá direcionar seu tráfego malicioso via rede a ambientes específicos do cenário proposto, cujas interfaces estão atreladas a rede privada do cenário que já é de conhecimento do atacante. A configuração topológica do cenário se encontra apresentada na figura 4.1

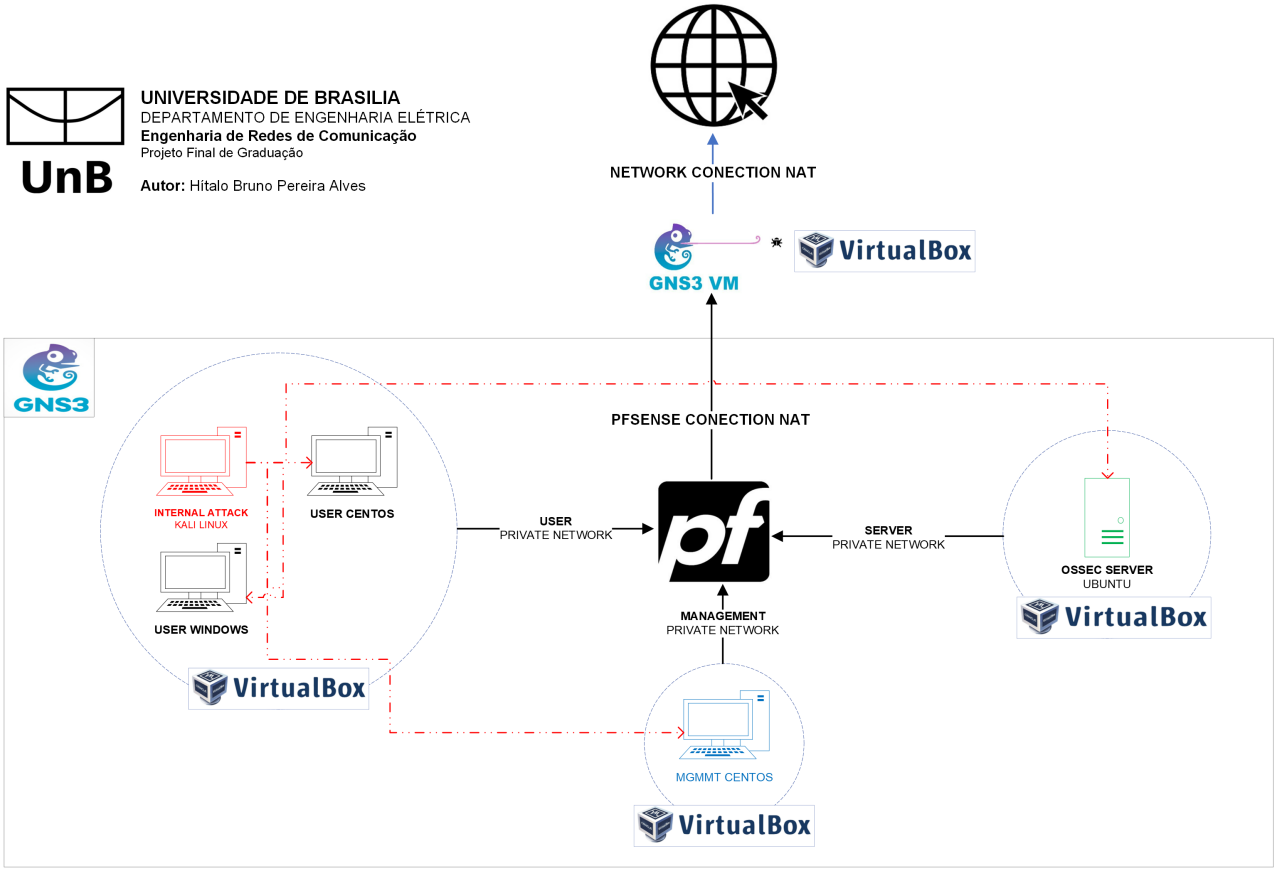

Figura 4.1: Metodologia de Ataque em ambiente emulado e virtualizado via ferramenta GNS-3 contendo uma segmentação de rede em 3 ambientes sendo eles Gerência, Usuários e Servidores, controlados via Firewall PfSense. Fonte: autor

A implementação e execução do projeto foi confeccionada a permitir um entendimento passo-a-passo do ambiente, como demonstrado a seguir:

- **PASSO 1 → Instalação do sistema de virtualização Oracle VirtualBox:** Todo o sistema emulado pela Aplicação GNS-3 é dependente de um ambiente virtualizado quando se utiliza o GNS-3 VM. Todo o procedimento necessário para instalação será detalhado a pela seção 4.2.1
- **PASSO 2 → Instalação da aplicação GNS-3 e da máquina virtual GNS-3 VM:** O GNS-3 é o sistema emulador e simulador utilizado para este projeto. Ele acompanha a máquina virtual GNS-3 VM, que faz o link das máquinas virtuais do sistema de virtualização com a GUI do GNS-3, possibilitando funcionalidades que serão melhor descritas na seção 4.2.2
- **PASSO 3 → Instalação das máquinas virtuais e importação para o sistema GNS-3:** Como o

sistema-mãe do projeto é o GNS-3, ele deve receber todas as máquinas virtuais criadas no virtualizador Oracle VirtualBox. Este processo é descrito pela seção 4.2.3

- **PASSO 4 → Projeto de segmentação da rede:** O projeto prevê a estruturação de três ambientes de rede a qual o atacante possua acesso via algumas falhas controladas de configuração. Mais detalhes serão expostos na seção 4.2.4
- **PASSO 5 → Instalação do OSSEC:** Na seção 4.2.5 será demonstrado o procedimento de instalação do HIDS OSSEC em sua funcionalidade Server e em sua funcionalidade Agent.
- **PASSO 6 → Ossec Web User Interface:** O OSSEC WUI é uma ferramente bastante auxiliadora no quesito de leitura de logs gerados pelos Agentes ao Servidor OSSEC. Detelhes desta ferramenta e como opera sua coleta de logs serão passadas na seção 4.2.6 .
- **PASSO 7 → Sistema Invasor** O Kali Linux é uma dos sistemas operacionais de código aberto mais comum para realização de testes de segurança tendo em vista seu vasto arsenal de ferramentas de análises e até mesmo de ataques. Detalhes de quais ferramentas serão utilizadas estarão descritas na seção 4.2.8.
- **PASSO 8 → Testes de Detecção e coleta de resultados:** Após todos cenário configurado e entendido, é necessário aplicar as técnicas de ataques descritas pela seção 4.2.8 e colher os resultados apresentados pelos logs gerados pelo HIDS OSSEC (a qual seu comportamento e configuração serão detalhados nas seções 4.2.5 e 4.2.7). Estes testes e resultados serão melhor detalhados no capítulo 5.

4.2 CONFIGURAÇÃO E DESENHO DA ARQUITETURA

4.2.1 INSTALAÇÃO DO VIRTUALIZADOR ORACLE VIRTUALBOX

O Virtualizador Oracle VirtualBox, como descrito pela seção 3.1, é uma ferramenta de virtualização multiplataforma baseada em arquitetura x86 e AMD64/Intel64 disponível como Software de Código Aberto sob os termos da segunda versão *GNU General Public License (GLP)*.

Para o desenho deste projeto e visto a necessidade de utilização de recursos de hardware, é importante informar as especificações técnicas da máquina a qual todo o ambiente foi configurado, haja vista se tratar, em situações de reprodução, de requisitos que são suficientes para este projeto, mas não necessariamente os requisitos mínimos.

Desta forma, para o desenvolvimento do projeto proposto, os recursos utilizados foram:

Tabela 4.1: Recursos computacionais utilizados para virtualização do ambiente proposto pela Figura 4.1

Recurso	**Tipo/Modelo**	**Recurso Disponível**	**Recurso Utilizado**
Armazenamento	SSD M.2 NVMe	235GB	218MB (apenas virtualizador)
Memória RAM	DDR-4 (*Double Data Rate*)	16GB	8GB (estimativa)
Processamento	Intel Core i5 1135G7	Até 4.2GHz	2.8 GHz (estimativa)

Em complemento da tabela 4.1, apresenta-se pela tabela 4.2 os recursos de software que foram utilizados tal como suas respectivas versões. Estas versões, em caso de uma futura reprodução e/ou contribuição podem estar desatualizadas, recomendando-se realizar testes de disponibilidade e integração dos ambientes.

Tabela 4.2: Recursos de software para virtualização do ambiente proposto pela Figura 4.1

Recurso	**Tipo/Modelo**	**Versão**
Sistema Operacional	Microsoft Windows	11 Pro
Virtualizador	Oracle VirtualBox	6.1.32 149290

Todo o processo de instalação do virtualizador Oracle VirtualBox é descrito pelo Anexo I.

4.2.2 INSTALAÇÃO DA APLICAÇÃO GNS-3 E GNS-3 VM

Como abordado pela seção 3.2, o GNS-3 é um software de código aberto que permite emular, configurar, testar e solucionar problemas de redes reais e virtuais distribuídos em uma pequena topologia que pode ser composta de *hosts*, passivos e ativos de redes, estejam eles hospedados em vários servidores ou mesmo na nuvem.

O GNS-3 opera em uma arquitetura Cliente-Servidor, como também abordado pela seção 3.2. No caso deste projeto, foi adotado como Servidor não só a máquina local, mas também a máquina virtual GNS-3 VM, que servirá como um servidor virtual para a comunicação com a interface gráfica de usuário das máquinas virtuais utilizadas neste projeto e descrito pela seção 4.2.3.

As versões utilizadas pelo Emulador e Simulador GNS-3 foram as descritas pela tabela 4.3. Da mesma forma que para os recursos listados pela tabela 4.2, as versões aqui apresentadas podem estar desatualizadas em caso de uma futura reprodução e/ou contribuição deste projeto, recomendando-se realizar testes de disponibilidade e integração dos ambientes.

Tabela 4.3: Recursos de software para virtualização do ambiente proposto pela Figura 4.1 - Complemento

Recurso	**Tipo/Modelo**	**Versão**
Sistema Operacional	Microsoft Windows	11 Pro
Virtualizador	Oracle VirtualBox	6.1.32 149290
Emulador/Simulador	Graphicalll Network Simulator 3	2.2.31
Emulador/Simulador Server VM	Graphicalll Network Simulator 3 VM	2.2.31 for VirtualBox

A tabela 4.4 apresenta os requisitos mínimos para instalação do GNS-3 a partir da versão 2.2.20 em diante. Todo processo de instalação o Emulador GNS-3 bem como do processo de importação do GNS-3 VM ao virtualizador VirtualBox se encontra disponível no Anexo II.

Observações:

- *Virtualização:* Você pode precisar habilitar este recurso através do BIOS do seu computador.
- *Armazenamento:* Você pode precisar de armazenamento adicional para seu sistema operacional e imagens do dispositivo.

Tabela 4.4: Recursos mínimos de instalação do GNS-3 para Microsoft Windows de acordo com [GNS3]

Item	Requerimento/Especificação
Sistema Operacional	Microsoft Windows 7 (64 bit) ou superior
Processador	2 ou mais cores lógicos
Virtualização	Extensão de virtualização necessária
Memória	4GB
Armazenamento	1GB de espaço disponível

4.2.3 CRIAÇÃO DE MÁQUINAS VIRTUAIS E IMPORTAÇÃO AO GNS-3

O Oracle VirtualBox funcionará como o emulador de *appliances virtuais* do nosso projeto de rede. As importações (com exceção do GNS-3 VM, explicado na seção 4.2.2) seguirá os padrões de importação do Oracle Virtual Box por meio de instalações de arquivos **.iso**. Para instalação dessas máquinas virtuais, serão disponibilizados tutoriais para cada uma das versões e distribuições utilizadas, de forma a montar o cenário apresentado pela figura 4.2.

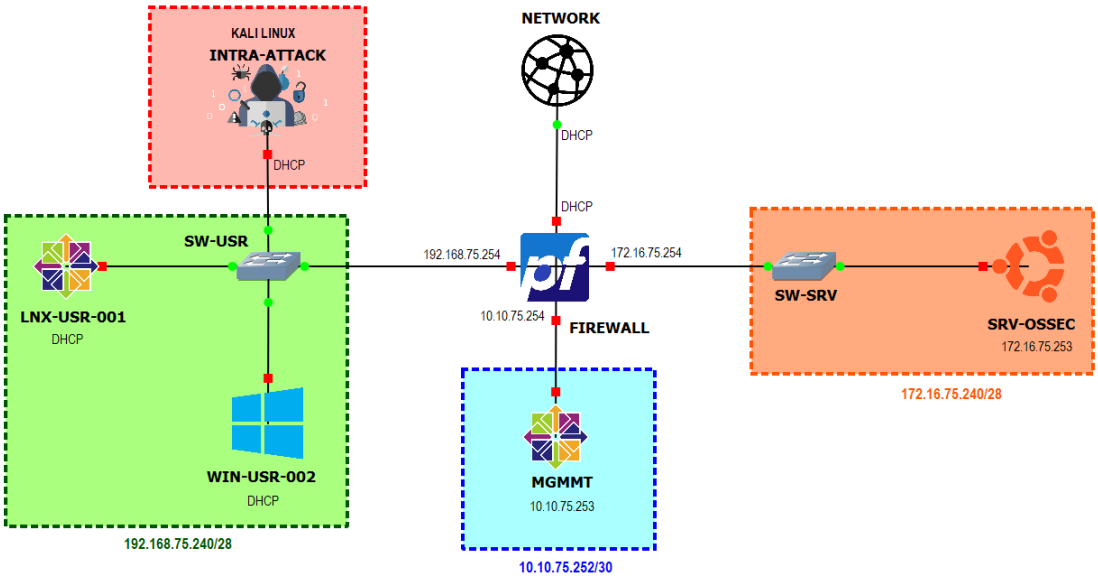

Figura 4.2: Cenário final do projeto desenhado o Emulador e Virtualizador GNS-3, a qual se encontra com a distribuições emuladas e os ativos de rede simulados. Fonte: autor

A tabela 4.5 a seguir apresenta os tutoriais utilizados para instalação de cada distribuição usada neste projeto. É importante ressaltar que a indicação destes tutoriais não são limitantes e podem ser acrescidos de novos tutoriais desde que bem executados.

Todo o processo de importação de máquinas virtuais ao GNS-3 pode ser visto no Anexo III.

Tabela 4.5: Tutoriais de instalação dos Sistemas Operacionais utilizados no Cenário da figura 4.2

Distribuição/S.O. e Versão	Tutorial de Instalação
Microsoft Windows 10 Pro 64-bit	No-hassle way to install Windows 10 with VirtualBox
Canonical Ubuntu 20.04	How to Install Ubuntu 20.04 LTS on VirtualBox in Windows 10
RHEL CentOS 7	How to Install CentOS 7 on VirtualBox in Windows 10
Netgate PfSense 2.6.0	How To Install PfSense on VirtualBox?
OffSec Kali Linux 2022.1	How to Install Kali Linux on VirtualBox

4.2.4 SEGMENTAÇÃO DE REDE

O projeto idealizou um cenário controlado mais parecido possível a uma organização, isto é, segmentado logicamente e com uma proteção mínima do tráfego de entrada e saída da rede. Para isso, criou-se três zonas de passivos, sendo que uma abriga a rede de usuários, outra a rede de servidores e por fim a rede de gerência, que dispõe de um *host* (que por sinal, se torna crucial para o projeto, haja vista que por ele se é possível acessar todas as plataformas) permissivo para controle da rede.

Assim sendo, a segmentação da rede foi feita conforme a tabela 4.6 a seguir:

Tabela 4.6: Segmentação de rede por zona, conforme ilustrado pela figura 4.2

Zona	Qtde. de Dispositivos	Endereço de rede
Gerência	1 dispositivo *host*	10.10.75.252/30
Rede Servidores	1 dispositivo server	172.16.75.240/28
Rede Usuários	3 dispositivos *host*	192.168.75.240/28

4.2.4.1 FIREWALL PFSENSE - Configurações Básicas

O Firewall PfSense, como abordado pela seção 3.3, é uma solução OpenSource utilizada em larga escala como uma forma alternativa de se substituir um *appliance* de firewall, além de ser um dispositivo que também opera como roteador ao disseminar suas rotas pelas interfaces ativas, possibilitando assim o mesmo ser um interconector de redes, sendo esta uma das principais funcionalidades utilizadas neste projeto.

Assim sendo, é importante salientar como é o processo de instalação e configuração deste dispositivo em referência a proposta adotada. A principal referência de instalação e configuração do pfSense pode ser vista na tabela 4.5, sendo seu processo de importação idêntico ao processo das demais *appliances* destacadas na seção anterior.

A configuração das redes de cada interface foi feita conforme figura 4.3 e tabela 5.1, apresentados a seguir:

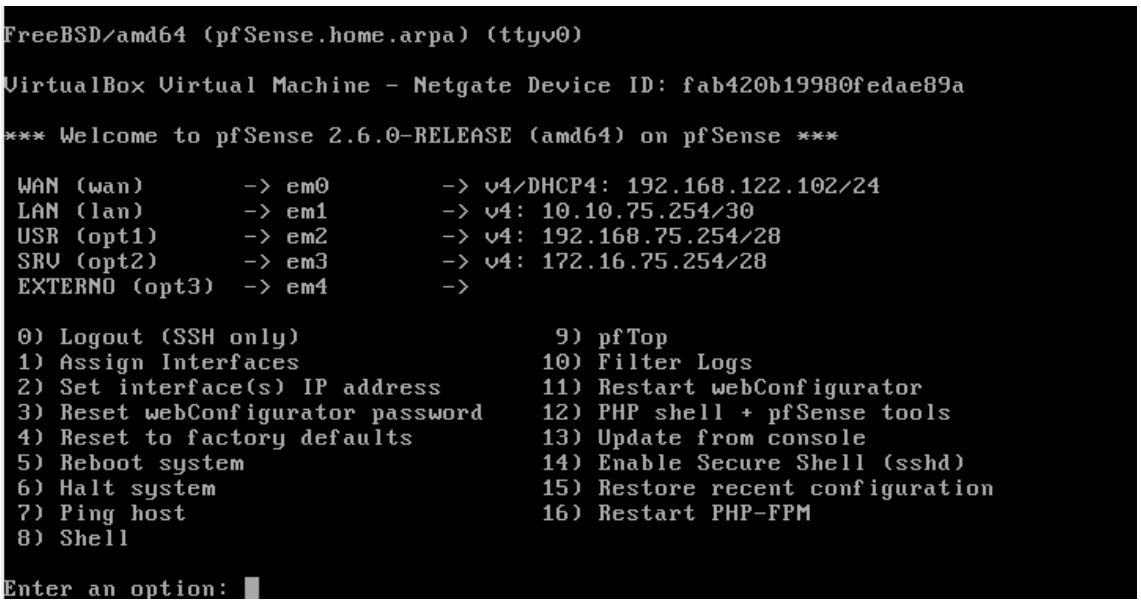

Figura 4.3: Console do Firewall pfSense com a descriminação das interfaces e IPs configurados, com atalhos para configurações rápidas. Fonte: autor

Tabela 4.7: Segmentação de rede por zona, conforme ilustrado pela figura 4.2

Interface	IP da Interface	Zona	Endereço de rede
WAN	DHCP: 192.168.122.102	Internet	192.168.122.0/24
LAN	Static: 10.10.75.254	Gerência	10.10.75.252/30
USR	Static: 192.168.75.254	Rede Usuários	192.168.75.240/28
SRV	Static: 172.16.75.254	Rede Servidores	172.16.75.240/28

4.2.4.2 FIREWALL PFSENSE - Regras

O pfSense disponibiliza via interface gráfica a possibilidade de atribuir regras de liberação e de bloqueio de tráfego através de redes, IPs específicos ou até mesmo portas ou protocolos. Essas configurações são altamente recomendadas para que não se libere tráfego de protocolos e portas desnecessários a zona de rede.

Orientações sobre essas configuraçõe são passadas pelo top 5 da OWASP (*Open Web Application Security Project*) a qual trata do tópico de ***Security Misconfiguration*** e pode ser acessado via site A05:2021 – Security Misconfiguration.

Para este projeto, não foram feitas nenhuma configuração de regras na rede Usuários que possa bloquear o tráfego. Na interface LAN, foi bloqueada saída para Internet, tendo em vista que não se fazia necessário o dispositivo de Gerência acessar a Internet sendo sua finalidade apenas o controle da rede interna. Na interface SRV foi permitido todo tipo de tráfego entre as interfaces SRV <> LAN e SRV <> USR, mas também não foi permitido saída para internet, sendo que também não se faz necessário a comunicação do OSSEC com uma rede externa, sendo seu objetivo um monitoramento interno.

Esta opção foi adotada para expor todas as vulnerabilidades possíveis de serem exploradas dentro de

cada zona, não sendo o foco deste fazer o bloqueio de qualquer tráfego oriundo da rede interna. Esta opção foi adotada, em complemento ao já abordado, para que não seja dificultado o ataque, isto é, dando liberdade ao atacante e dando o centro de atenção ao HIDS OSSEC. Desta forma, as configurações de cada interface ficou conforme figuras 4.4, 4.5, 4.6 e 4.7, a seguir:

- A regra descrita pela figura 4.4 são regras padrões das interfaces WAN do pfSense para que não seja permitido a utilização de endereços privados ou bogon no lado da WAN, conforme RFC 1918.

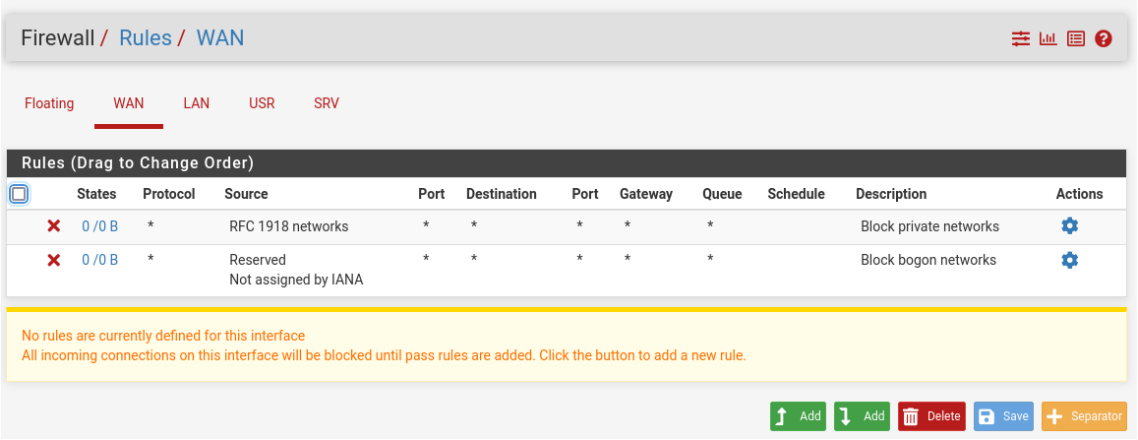

Figura 4.4: Regras aplicadas para permissividade de todo o tráfego pela interface WAN do pfSense. Fonte: autor

- A regra apresentada pela Figura 4.5 refere-se a interface LAN do pfSense, que foi destinada a rede de Gerência como apresentado pelas figuras 4.2 e 4.1. Dado que sua finalidade é acessar os ativos da rede e gerenciar o parque de funcionalidades previstas, não se fazia necessário que a mesma tivesse acesso a internet. Esta tomada de decisão também se embasa em conceitos de segurança, haja vista que os ataques direcionados a uma rede de gerência vindos da rede externa podem causar maiores danos em contrapartida a um ataque a host usuário comum, a depender claro dos privilégios deste host. Precavendo-se de acessos vindo da interface USR (onde se encontra o cenário de ataque) para que usuários comuns não acessem a gerência dos ativos da rede, não se permitiu tráfego entre estas interfaces.

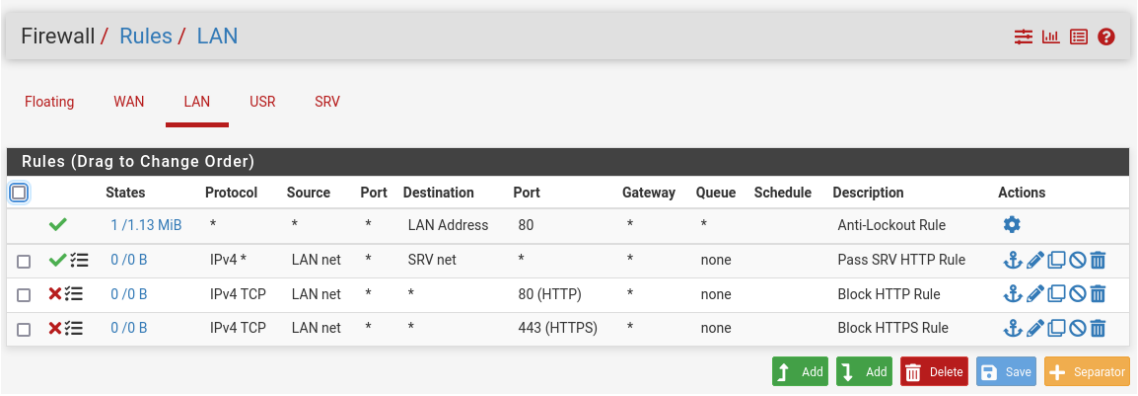

Figura 4.5: Regras aplicadas para permissividade de todo o tráfego menos HTTP e HTTPS pela interface LAN do pfSense. Fonte: autor

- A regra apresentada pela Figura 4.6 refere-se a interface USR do pfSense, que foi destinada a rede de Usuários como apresentado pelas figuras 4.2 e 4.1. Por se tratar de uma rede considerada aberta, decidiu-se por não aplicar regras de restrição a esta rede. Tomou-se esta atitude tendo em vista a geração de vulnerabilidades e facilidades de comunicação, uma vez que o cenário de ataque se encontra nela.

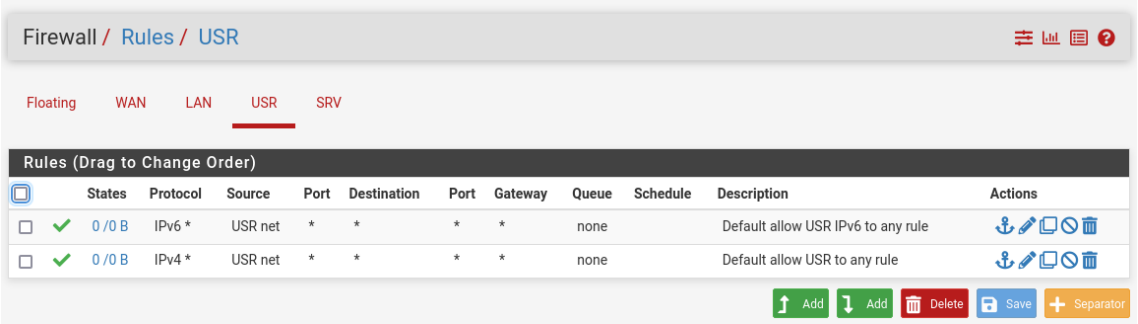

Figura 4.6: Regras aplicadas para permissividade de todo o tráfego pela interface USR do pfSense. Fonte: autor

- A regra apresentada pela Figura 4.7 refere-se a interface SRV do pfSense, que foi destinada a rede de Servidores como apresentado pelas figuras 4.2 e 4.1. A rede Servidores, assim como a rede Gerência, tem como objetivo prover atuação interna no projeto. Desta forma, não se faz necessário que a rede permita tráfego para a Internet, sendo necessário somente que a mesma mantenha comunicação com as demais interfaces do cenário (LAN e USR).

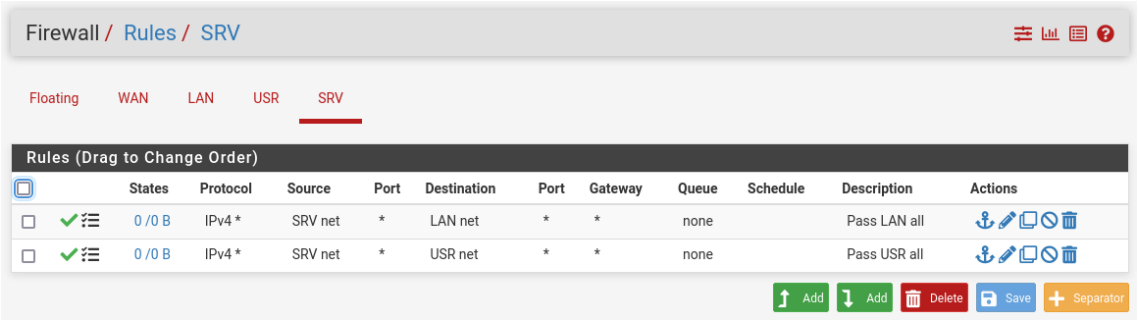

Figura 4.7: Regras aplicadas para permissividade de todo o tráfego pela interface SRV do pfSense. Fonte: autor

Apresentada todas as regras e configurações por parte do Firewall pfSense, a segmentação de rede, com incremento dos usuários e demais dispositivos ficou da seguinte forma:

Tabela 4.8: Segmentação de rede por dispositivo, conforme apresentado pela figura 4.2

Dispositivo	**S.O.**	**Endereço IP**	**Zona**	**Endereço de rede**
INTRA-ATTACK	Kali Linux	DHCP: 192.168.75.246	Atacante	192.168.75.240/28
LNX-USR-001	RHEL CentOS 7	DHCP: 192.168.75.244	Rede Usuários	192.168.75.240/28
WIN-USR-002	MS Windows 10 Pro	DHCP: 192.168.75.245	Rede Usuários	192.168.75.240/28
MGMMT	RHEL CentOS 7	Static: 10.10.75.253	Gerência	10.10.75.252/30
SRV-OSSEC	Can. Ubuntu 20.04	Static: 172.16.75.253	Rede Servidores	172.16.75.253/28

4.2.5 INSTALAÇÃO DO HIDS OSSEC

Como pode ser demonstrado via seção anterior, o HIDS OSSEC é um sistema multiplataforma e isto faz com que ele obtenha vários cenários de instalação. Pensando nisso, a própria Atomicorp, detentora e desenvolvedora do OSSEC disponibiliza em seu site vários tutoriais para instalação do OSSEC para todas as multiplataformas a qual ele é compatível. Para demais distribuições que não sejam as listadas nesse projeto e as que inclusive se encontram listadas, orienta-se seguir os passos recomendados pelo site https://www.ossec.net/download-ossec/.

Entretanto, depois de algumas leituras e testes, os tutoriais apresentados pelo site acima não foram suficientes para conclusão com êxito do HIDS OSSEC. Isso se dá pelo fato da implementação também da interface web do OSSEC (OSSEC WUI) para gestão dos logs bem como bibliotecas das distribuições Linux que eram necessárias para complementar o projeto, tendo em vista a instalação via arquivo ISO, limpa e sem bibliotecas padrões de algumas imagens de disco já prontas, como o caso das imagens fornecidas pla OSBOXES.

Vale ressaltar também que o HIDS OSSEC fornece apenas o Agent para distribuições Windows, sendo necessário que o servidor seja instalado em uma distribuição Linux.

O OSSEC Server desempenhará o papel de coletor de logs e gestor do alimentador do OSSEC WUI, isto é, será o ponto regente de toda arquitetura de detecção de intrusão e obviamente, o dispositivo mais visado pelo sistema de intrusão. Todo seu processo de instalação, para a Distribuição Canonical Ubuntu 20.04, se encontra disponível via Anexo IV.1.

O OSSEC Agent desempenhará o papel de informante ativo ao OSSEC Server, isto é, além de tomar atitudes tais como tomadas pelo OSSEC Server (exceto no caso do MS Windows, que não detém da resposta ativa), reportará logs dos eventos ocorridos em seu terreno ao OSSEC Server, para que assim o administrador do ambiente possa ter ciência das ocorrências destinadas à aquele *host* e assim, por exemplo, mitigar as vulnerabilidades ali presentes e de forma reativa, replicar essas ações aos *hosts* semelhantes do seu parque de usuários.

Semelhante ao processo de instalação descrito pelo Anexo IV.1, o processo de instalação do OSSEC Agent para distribuições Linux RHEL CentOS 8 é descrito pela seção IV.2 bem como as peculiaridades deste processo. Já o processo de instalação do OSSEC Agent em distribuições MS Windows 10 são guiadas, como todos a grande maioria dos programas deste sistema, via executável e seu processo de instalação é descrito pelo tutorial vide Anexo IV.3.

4.2.6 OSSEC WEB USER INTERFACE (WUI)

O OSSEC Web User Interface é uma interface web centralizadora de logs e informações sobre o HIDS OSSEC, tanto servidor quanto agentes, considerado atualmente na comunidade de desenvolvimento como um projeto-morto dado que a muito tempo não se trabalha em novas versões e atualizações da plataforma haja vista novos e robustos integradores de logs.

Contudo, para este projeto, vislumbrou-se a oportunidade de explorar esta ferramenta tendo em vista com por ela é possível se verificar alguns cenários de ataques de intrusão de forma mais intuitiva que a

console do OSSEC via arquivo de logs.

A seguir (pela figura 4.8), é possível ver, com alguns detalhes como nome, endereço IP, data da última vez em que estava ativo e sistema operacional, os agentes disponíveis apresentados na tela inicial do OSSEC WUI.

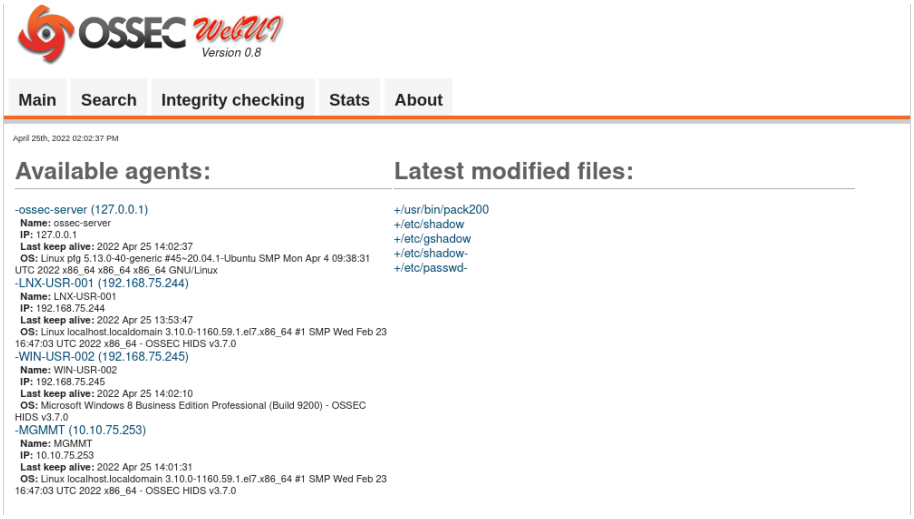

Figura 4.8: Tela inicial resumida do OSSEC WUI, onde é possível se verificar informações de Agentes Disponíveis bem como alguns ferramentais de exploração do HIDS OSSEC. Fonte: autor

Não somente, a ferramenta apresenta algumas abas em sua configuração inicial, explorada a seguir:

- **Search:** A aba **Search** tem como principal objetivo facilitar a busca dos alertas de acordo com alguns parâmetros estabelecidos pelo usuário, que vai desde um intervalo simples de data e hora até o formato do log que se deseja filtrar.

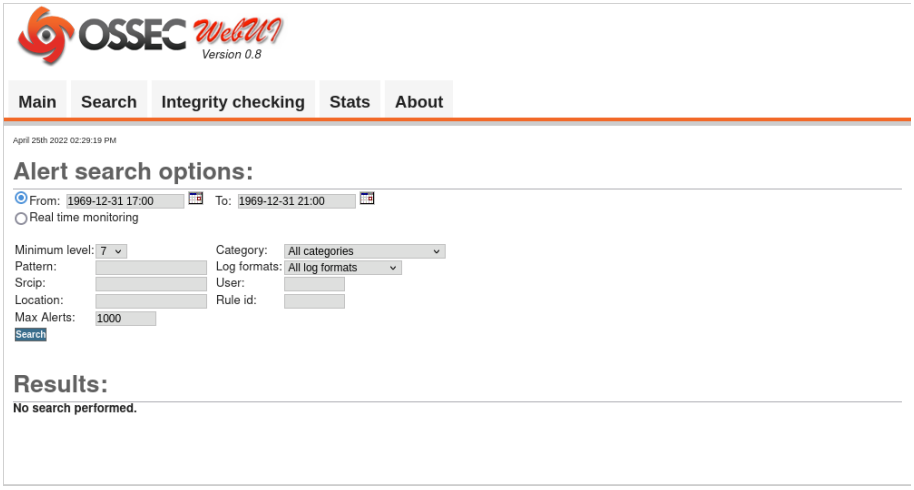

Figura 4.9: OSSEC WUI - Aba Search e seus parâmetros. Fonte: autor

- **Integrity Checking:** Na aba **Integrity Checking** é possível verificar a integridade dos arquivos via um banco de dados de alterações que foram feitas desde o início do monitoramento do agente naquele

determinado *host*. Este campo é bastante útil para se verificar quais arquivos foram alterados durante a operação do sistema e quais *checksum* foram modificados e o tamanho do arquivo modificado, se tornando um importante aliado em auditorias de cibersegurança, por exemplo.

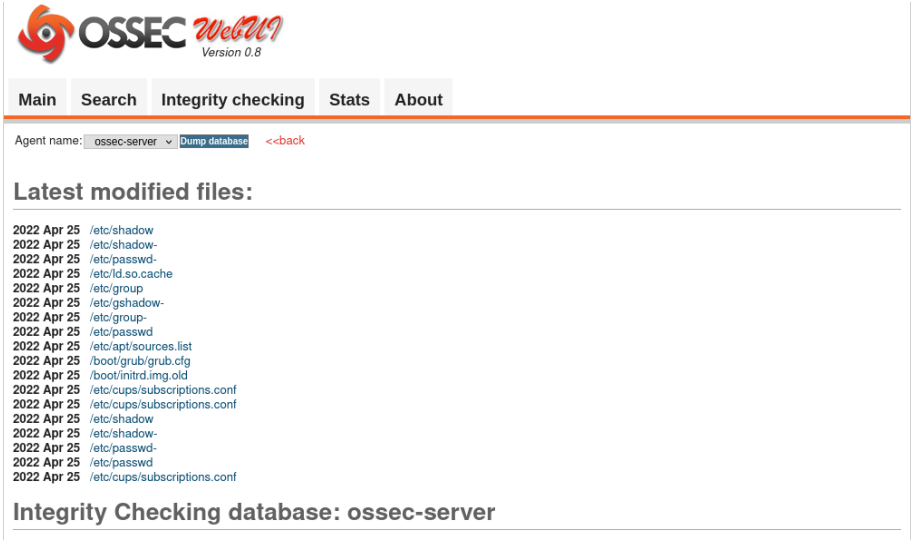

Figura 4.10: OSSEC WUI - Aba Checksum e o exemplo de varredura realizada para o OSSEC Server. Fonte: autor

A aba **Stats**, devido a falta de atualização da OSSEC WUI, não oferece parâmetros para geração de estatísticas dentro de um intervalo de datas válidas. A última aba, **About**, apresenta informações sobre a interface gráfica e sobre seus mantenedores e desenvolvedores, que curiosamente são nacionais, assim como o HIDS OSSEC.

4.2.7 LOGS

Os Logs do HIDS OSSEC são reportados via arquivo denominado `ossec.conf` e se localiza no diretório `/var/ossec/etc`. Este arquivo recebe todas as regras que são baseadas em arquivos xml, como o caso das regras de configuração (`rules\_config.xml`), regras de SSH (`sshd\_rules.xml`), regras de Telnet (`telnetd\_rules.xml`), regras de Syslog (`syslog\_rules.xml`) entre várias outras, inclusive personalizadas, isto é, o HIDS OSSEC permite que o administrador crie um arquivo xml de regras próprias e o insira no arquivo `ossec.conf` para serem executadas pelo mesmo.

Algumas informações necessárias para este projeto estão listadas dentro do arquivo `ossec.conf` tais como:

- **REGRA** `<SYSCHECK>|` **:** A regra `<syscheck>` estabele a lista de diretórios que deverão ser verificados dentro de cada *host*. Esta regra se encontra presente tanto para o OSSEC Server quanto para os OSSEC Agentes e é exposta pela figura 4.11. Comumente, a regra `<syscheck>` tem como padrão ignorar verificação de arquivos da raiz **/etc** e em aplicações ou arquivos específicos que não se fazem necessário validar a integridade. Do arquivo padrão de configuração, removemos da lista de ignorados o arquivo *host.deny* tendo em vista que nosso objetivo, como será mostrado a seguir, é que ele não seja alterado e caso seja, seja de fato notificado.

```
<syscheck>
  <!-- Frequency that syscheck is executed - default to every 22 hours -->
  <frequency>79200</frequency>

  <!-- Directories to check  (perform all possible verifications) -->
  <directories check_all="yes">/etc,/usr/bin,/usr/sbin</directories>
  <directories check_all="yes">/bin,/sbin,/boot</directories>

  <!-- Files/directories to ignore -->
  <ignore>/etc/mtab</ignore>
  <ignore>/etc/mnttab</ignore>
  <ignore>/etc/hosts.deny</ignore>
  <ignore>/etc/mail/statistics</ignore>
  <ignore>/etc/random-seed</ignore>
  <ignore>/etc/adjtime</ignore>
  <ignore>/etc/httpd/logs</ignore>
  <ignore>/etc/utmpx</ignore>
  <ignore>/etc/wtmpx</ignore>
  <ignore>/etc/cups/certs</ignore>
  <ignore>/etc/dumpdates</ignore>
  <ignore>/etc/svc/volatile</ignore>

  <!-- Windows files to ignore -->
  <ignore>C:\WINDOWS/System32/LogFiles</ignore>
  <ignore>C:\WINDOWS/Debug</ignore>
  <ignore>C:\WINDOWS/WindowsUpdate.log</ignore>
  <ignore>C:\WINDOWS/iis6.log</ignore>
  <ignore>C:\WINDOWS/system32/wbem/Logs</ignore>
  <ignore>C:\WINDOWS/system32/wbem/Repository</ignore>
  <ignore>C:\WINDOWS/Prefetch</ignore>
  <ignore>C:\WINDOWS/PCHEALTH/HELPCTR/DataColl</ignore>
  <ignore>C:\WINDOWS/SoftwareDistribution</ignore>
  <ignore>C:\WINDOWS/Temp</ignore>
  <ignore>C:\WINDOWS/system32/config</ignore>
  <ignore>C:\WINDOWS/system32/spool</ignore>
  <ignore>C:\WINDOWS/system32/CatRoot</ignore>
</syscheck>
```

Figura 4.11: Regras padrões do Syschceck para Agentes e Servidores OSSEC. Fonte: autor

- **REGRA** `<ROOTCHECK>|` **:** A regra `<rootcheck>` faz referência verificação de rootkits ativada por meio dos processos de inserção de valores descritos pela seção IV.1.1. O `ossec.conf` tem em seu banco os principais caminhos e arquivos de onde se encontram os principais rootkits para cada sistema operacional e/ou distribuição, como apresentado pela figura 4.12

```
<rootcheck>
  <rootkit_files>/var/ossec/etc/shared/rootkit_files.txt</rootkit_files>
  <rootkit_trojans>/var/ossec/etc/shared/rootkit_trojans.txt</rootkit_trojans>
  <system_audit>/var/ossec/etc/shared/system_audit_rcl.txt</system_audit>
  <system_audit>/var/ossec/etc/shared/cis_debian_linux_rcl.txt</system_audit>
  <system_audit>/var/ossec/etc/shared/cis_rhel_linux_rcl.txt</system_audit>
  <system_audit>/var/ossec/etc/shared/cis_rhel5_linux_rcl.txt</system_audit>
</rootcheck>
```

Figura 4.12: Listas dos principais caminhos e arquivos onde se encontram os rootkits citados na regra <rootcheck> para Agentes e Servidores OSSEC. Fonte: autor

- **REGRA** `<ACTIVE-RESPONSE>|` **:** A Regra `<active-response>` dita o processo de execução de bloqueio de uma atividade de intrusão classificada como nível 6 em diante (por padrão) pelo HIDS OSSEC (como o caso de um brute-force). Esta regra faz a chamada de dois scripts já estabelecidos dentro do `ossec.conf`, a quais são:

  - **firewal-drop.sh:** Script que realiza o *DROP* do *host* que esteja realizando algum comportamento malicioso. O *DROP* é o ato de derrubar uma conexão, isto é, este script visa criar uma

regra no IPTABLES do *host* (quando Distribuição Linux) para que seja imediatamente derrubada aquela tentativa de conexão possivelmente maliciosa. O tempo *default* de **DROP** é de 600 segundos, isto é, a conexão permanece derruba para aquele endereço de IP por 10 minutos. Contudo, está valoração é uma variável e pode ser configurada de acordo com os propósitos desejados pelo administrador.

– `host-deny.sh|`: Script que realiza a negação de acesso de forma temporária do *host* que possivelmente esteja realizando alguma atividade não autorizada em direção ao possível alvo. Note que, o host-deny é diferente do firewall-drop dado que ele não estabelece uma derrubada de serviço e sim uma negação de acesso, isto é, por mais que a regra seja removida neste período que, assim como o firewall-drop é de 600 segundos, do IPTABLES do *host*, o acesso do IP do possível atacante ao *host*-alvo continuará indisponível. Daqui parte a justificativa de se tirar o host-deny da lista de não monitorados pela regra **<syscheck>**, tendo em vista que caso haja essa remoção, o administrador, a partir do <syscheck>, deverá ser notificado da alteração da integridade do arquivo e assim prover as devidas providências de forma ágil.

```
<!-- Active Response Config -->
<active-response>
  <!-- This response is going to execute the host-deny
     - command for every event that fires a rule with
     - level (severity) >= 6.
     - The IP is going to be blocked for  600 seconds.
    -->
  <command>host-deny</command>
  <location>local</location>
  <level>6</level>
  <timeout>600</timeout>
</active-response>

<active-response>
  <!-- Firewall Drop response. Block the IP for
     - 600 seconds on the firewall (iptables,
     - ipfilter, etc).
    -->
  <command>firewall-drop</command>
  <location>local</location>
  <level>6</level>
  <timeout>600</timeout>
</active-response>
```

Figura 4.13: Regras e chamadas do Active Response com os parâmetros de firewall-drop e host-deny, como descritos acima. Fonte: autor

4.2.8 SISTEMA INVASOR: KALI LINUX

O Kali Linux é uma distribuição Linux baseada em Debian criada principalmente para fins de auditoria de cibersegurança, isto é, tem como seu principal, para não dizer único foco, em testes de penetração e invasão de *hosts* e demais dispositivos, mantido atualmente pela Offensive Security.

Para este projeto, o Kali desempenhará a função de invasor na arquitetura, proporcionando ataques de força bruta (BRUTE FORCE) após analisar o cenário que encontrará em cada host, propositalmente criado para este ambiente e se desenhará em basicamente duas portas: SSH e FTP.

Espera-se inicialmente que estes ataques sejam automaticamente identificados e bloqueados pela resposta ativa do HIDS OSSEC, mesmo que posteriormente, o atacante altere seu endereço de rede em busca de novas investidas.

Para realização do ataque, utilizará-se de uma ferramenta bastante conhecida no meio da cibersegurança quando se trata de ataques de intrusão baseados em força bruta, que é o HYDRA.

O Hydra é um cracker de login paralelizado ágil e flexível, que suporta vários protocolos e que possibilita que atacantes tanto quanto consultores de segurança explorem vulnerabilidades de configuração expostas em dispositivos cruciais para a operação de uma rede como *hosts*, servidores, roteadores e até mesmo firewalls

Este cracker busca por meio do uso da técnica de tentativa e erro a força bruta realizar um acesso remoto não autorizado via um ou vários usuários e uma ou várias senhas por uma ou várias portas, o que demonstra seu poderoso arsenal de fogo em face das possíveis vulnerabilidades de configurações presentes em tais dispositivos.

Todo o processo de utilização do Hydra passa atualmente pelo xHydra, a interface gráfica que facilita ainda mais as formas de realização do teste de intrusão, tendo em vista que a mesma deixa as informações mais intuitivas. Um exemplo desta tela é apresentado a seguir, pela figura 4.14 e será mais detalhada na execução dos ataques coordenados previstos neste projeto, explorados e apresentado na seção 5.

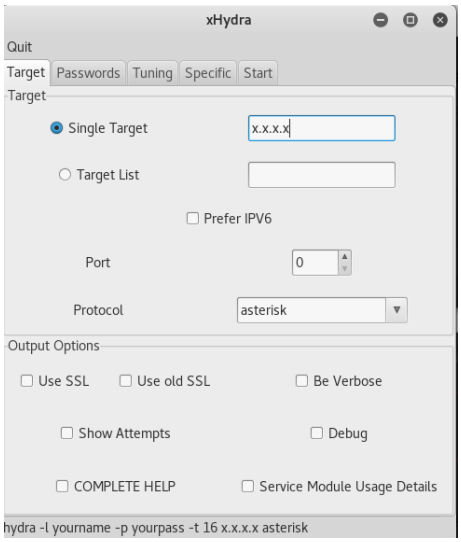

Figura 4.14: Exemplo de tela inicial do xHydra, interface gráfica facilitadora do Cracker Hydra, disponível via Kali Linux. Fonte: [QWERTYGUY 2018]

Os passos a seguir serão destinados aos testes bem como a obtenção dos resultados baseados no comportamento da ferramenta HIDS OSSEC, embasando-se em todo cenário construído e descrito por essa seção.

5 TESTES E RESULTADOS

Para apurar o desempenho, eficiência e eficácia da ferramenta OSSEC HIDS, foram feitos alguns ataques coordenados e comumente conhecidos, que será retratado a seguir na forma de **panoramas**, a qual cada um destes irá abordar a forma de execução e o resultado obtido.

Os ataques aqui apresentados seguiram o modelo *Cyber Kill Chain* conforme análise feita pela seção 2.4 e foram escolhidos com base na última Estatística apresentada pela CERT.br em dois cenários:

1. **NMAP:** Incidentes Reportados ao CERT.br no período de Janeiro a Dezembro de 2020 com relação aos Tipos de Ataque: Optou-se pela realização da feature de Port Scanning tendo em vista ser o tipo de scan mais utilizado no início de um processo de ataque;

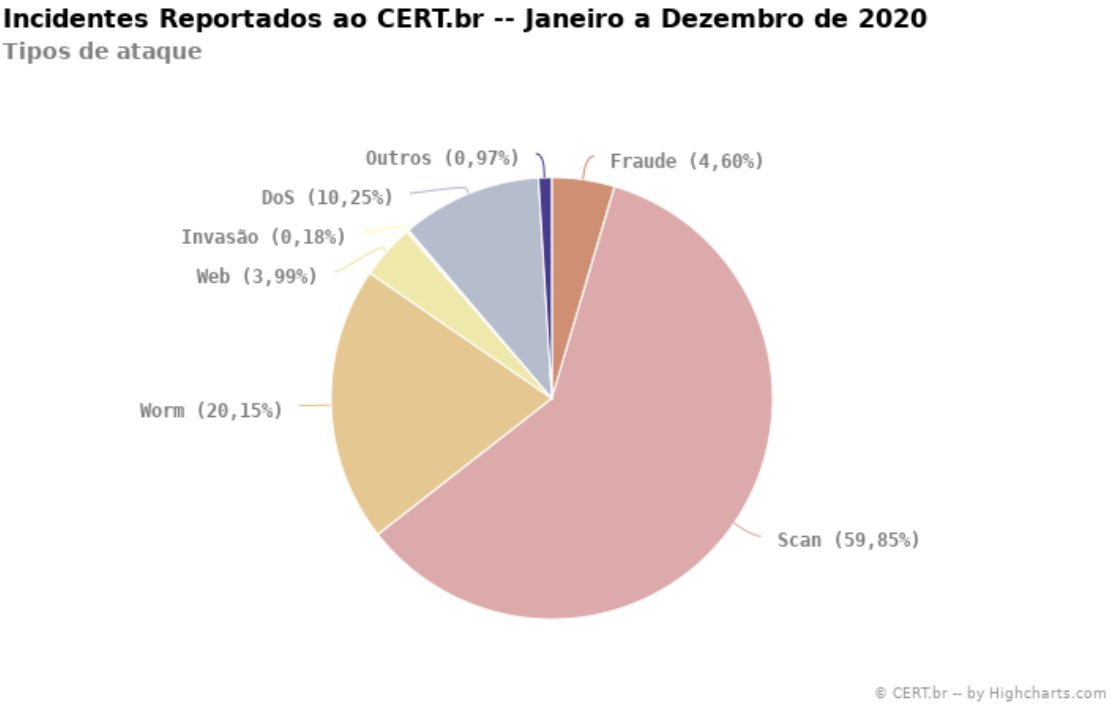

Figura 5.1: Incidentes Reportados ao CERT.br no período de Janeiro a Dezembro de 2020 com relação aos Tipos de Ataque. Fonte [CERT.br 2020]

2. **BRUTE FORCE SSH (22):** Incidentes Reportados ao CERT.br no período de Janeiro a Dezembro de 2020 com relação aos Scans reportados por porta: Optou-se por utilizar a técnica de BRUTE FORCE em prol da geração de alto tráfego de acessos errôneos para que se fosse possível observar o comportamento da resposta ativa da ferramenta HIDS OSSEC.

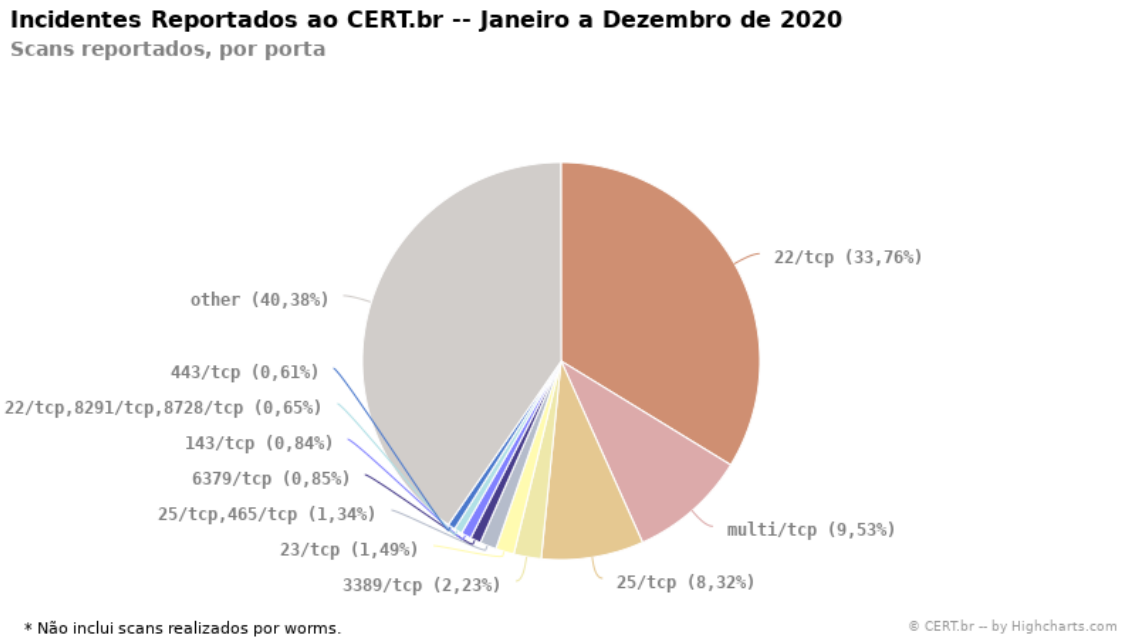

Figura 5.2: Incidentes Reportados ao CERT.br no período de Janeiro a Dezembro de 2020 com relação aos Scans reportados por porta. Fonte [CERT.br 2020]

3. **ROOTKIT MODE-KERNEL:** A escolha deste tipo de ataque tem como fundamentação realizar um teste comportamental na ferramenta HIDS OSSEC em comparativo com o projeto apresentado por [Alarcão 2021], tendo como objetivo visualizar o comportamento do HIDS em um mesmo cenário em face a um Endpoint Detection and Response.

Os testes, ataques, técnicas, estudos e resultados aqui expostos estão datados em uma janela de pouco mais de 3 meses de desenvolvimento e execução, sendo precisamente executados entre 17 de fevereiro de 2022 a 27 de abril de 2022.

É importante salientar que, na concepção dos ataques por parte do *host* **INTRA-ATTACK** há uma defasagem entre o host INTRA-ATTACK e OSSEC WUI de exatas 01h, sendo o horário correto da realização do panorama proposto o horário apresentado pelo OSSEC WUI. Este problema foi apresentado por uma falha constante de sincronização por parte do Kali Linux com os servidores NTP instalados.

Os Panoramas apresentarão cenário de execução sempre para o Servidor OSSEC e para um dos 3 *host* listados na tabela 5.1 tendo em vista testar a funcionalidade do agente OSSEC.

5.1 PANORAMA 1 - SCAN NMAP

O *Network Mapper* (NMAP) é uma poderosa ferramenta de código aberto desenvolvida por **Gordon Lyon** e tem como finalidade básica descobrir *host* e serviços em uma rede por meio do envio de pacotes a estes diversos *host* e diversos serviços e analisando suas respostas. Por meio destas respostas retorna-se um cenário de serviços ativos em cada *host* analisado, por exemplo, *host* que respondam pela porta TCP e/ou ICMP. [Lyon 2014].

Dentre as *features* inclusas no NMAP e listadas por [Lyon 2014] em seus estudos, temos interesse no ***Port scanning***, haja vista que para um ataque bem sucedido, é necessário realizar o scanning das portas abertas em cada *host*.

Por meio da máquina INTRA-ATTACK, operando via endereço IP 192.168.75.246 fornecido via DHCP pela interface USR do Firewall pfSense e posicionado na rede proposta pela figura 4.1 como um **Atacante Interno**, foi realizado o comando NMAP em direção ao Servidor OSSEC e ao *host* WIN-USR-002.

```
┌──(root㉿kali)-[~]
└─# nmap -A 172.16.75.253
Starting Nmap 7.92 ( https://nmap.org ) at 2022-04-26 01:10 EDT
Nmap scan report for 172.16.75.253
Host is up (0.0030s latency).
Not shown: 998 closed tcp ports (reset)
PORT   STATE SERVICE VERSION
22/tcp open  ssh     OpenSSH 8.2p1 Ubuntu 4ubuntu0.4 (Ubuntu Linux; protocol 2.0)
| ssh-hostkey:
|   3072 bb:50:fc:6d:85:ec:c4:32:5d:48:01:20:7b:63:f9:39 (RSA)
|   256 79:3d:69:78:58:53:39:26:08:c6:50:0d:bb:f0:af:09 (ECDSA)
|_  256 ec:ad:7b:36:4b:ca:f8:a5:5c:bd:cf:c4:67:ea:d9:a5 (ED25519)
80/tcp open  http    Apache httpd 2.4.41 ((Ubuntu))
|_http-server-header: Apache/2.4.41 (Ubuntu)
|_http-title: Apache2 Ubuntu Default Page: It works
No exact OS matches for host (If you know what OS is running on it, see https://nmap.org/sub
mit/ ).
TCP/IP fingerprint:
OS:SCAN(V=7.92%E=4%D=4/26%OT=22%CT=1%CU=34814%PV=Y%DS=2%DC=T%G=Y%TM=62677EF
OS:2%P=x86_64-pc-linux-gnu)SEQ(SP=F7%GCD=1%ISR=10F%TI=Z%II=I%TS=A)OPS(O1=M5
OS:B4ST11NW7%O2=M5B4ST11NW7%O3=M5B4NNT11NW7%O4=M5B4ST11NW7%O5=M5B4ST11NW7%O
OS:6=M5B4ST11)WIN(W1=FE88%W2=FE88%W3=FE88%W4=FE88%W5=FE88%W6=FE88)ECN(R=Y%D
OS:F=Y%T=40%W=FAF0%O=M5B4NNSNW7%CC=Y%Q=)T1(R=Y%DF=Y%T=40%S=O%A=S+%F=AS%RD=0
OS:%Q=)T2(R=N)T3(R=N)T4(R=N)T5(R=Y%DF=Y%T=40%W=0%S=Z%A=S+%F=AR%O=%RD=0%Q=)T
OS:6(R=N)T7(R=N)U1(R=Y%DF=N%T=40%IPL=164%UN=0%RIPL=G%RID=G%RIPCK=G%RUCK=G%R
OS:UD=G)IE(R=Y%DFI=N%T=40%CD=S)

Network Distance: 2 hops
Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel

TRACEROUTE (using port 587/tcp)
HOP RTT     ADDRESS
1   1.61 ms 192.168.75.254
2   2.96 ms 172.16.75.253

OS and Service detection performed. Please report any incorrect results at https://nmap.org/
submit/ .
Nmap done: 1 IP address (1 host up) scanned in 20.29 seconds
```

Figura 5.3: Ataque NMAP com *feature* de *Port Scanning* direcionado ao *host* OSSEC Server com endereço IP 172.16.75.253 a qual foi descoberta a abertura das portas SSH (22) e HTTP (80) bem como informações de sistema e rede. Fonte: autor

```
┌──(root㉿kali)-[~]
└─# nmap -A 192.168.75.245
Starting Nmap 7.92 ( https://nmap.org ) at 2022-04-26 01:57 EDT
Nmap scan report for 192.168.75.245
Host is up (0.0015s latency).
Not shown: 999 filtered tcp ports (no-response)
PORT   STATE SERVICE VERSION
22/tcp open  ssh     OpenSSH for_Windows_7.7 (protocol 2.0)
| ssh-hostkey:
|   3072 38:c0:64:4f:56:ac:63:57:05:85:74:b7:88:08:be:b7 (RSA)
|   256 cb:ca:fa:a0:68:31:d9:8c:29:f8:65:08:20:cd:f3:a8 (ECDSA)
|_  256 db:30:5d:ad:1a:ee:19:e3:cc:bd:ca:9b:9e:27:42:69 (ED25519)
MAC Address: 08:00:27:6E:7C:B9 (Oracle VirtualBox virtual NIC)
Warning: OSScan results may be unreliable because we could not find at least 1 open and 1 cl
osed port
Device type: general purpose|specialized
Running (JUST GUESSING): Microsoft Windows XP (92%), AVtech embedded (87%), FreeBSD 6.X|10.X
 (86%)
OS CPE: cpe:/o:microsoft:windows_xp::sp3 cpe:/o:freebsd:freebsd:6.2 cpe:/o:freebsd:freebsd:1
0.3
Aggressive OS guesses: Microsoft Windows XP SP3 (92%), AVtech Room Alert 26W environmental m
onitor (87%), FreeBSD 6.2-RELEASE (86%), FreeBSD 10.3-STABLE (85%), Microsoft Windows XP SP2
 (85%)
No exact OS matches for host (test conditions non-ideal).
Network Distance: 1 hop

TRACEROUTE
HOP RTT     ADDRESS
1   1.48 ms 192.168.75.245

OS and Service detection performed. Please report any incorrect results at https://nmap.org/
submit/ .
Nmap done: 1 IP address (1 host up) scanned in 12.27 seconds
```

Figura 5.4: Ataque NMAP com *feature* de *Port Scanning* direcionado ao *host* WIN-USR-002 com endereço IP 192.168.75.245 a qual foi descoberta a abertura da porta SSH (22) bem como informações de sistema e rede. Fonte: autor

5.1.1 RESULTADOS OBTIDOS

5.1.1.1 *HOST* OSSEC SERVER IP: 172.16.75.253

Para o *host* OSSEC-SERVER, o panorama 5.1 foi executado as 02:11 do dia 26/04/2022 e através dele foi possível identificar que o serviço SSH estava aberto na porta 22 tanto quanto o serviço HTTP na porta 80, sendo esta por meio do servidor HTTP Apache na versão 2.4.41.

Nesta identificação, para o serviço SSH foi possível verificar quais algoritmos são utilizados na chave de acesso SSH pelo *host* e qual sistema operacional este *host* opera. Foi possível verificar a quantidade de saltos necessários até que se chegue ao *host* e consequentemente seu gateway. Todos os parâmetros relatados são apresentados na figura 5.3

Observou-se que o comportamento do HIDS OSSEC por meio da sua interface gráfica OSSEC WUI para a varredura da porta SSH foi positiva e responsiva, retornando por meio da regra 5701, listada pelo arquivo sshd_rules.xml, a mensagem: *Possible attack on the ssh server (or version gathering)*.

O OSSEC WUI não apresentou a origem do possível ataque, mas retornou um alerta objetivo e de severidade 8, isto é, média para alta evidenciando que, independente de onde venha, está vindo para o *host* um possível ataque SSH ou uma varredura para coleta de versão de operação e que há a necessidade de uma atenção para este evento.

Por não apresentar uma vasta tentativa de conexão, mas sim um *broadcast* de pacotes no serviço, a reposta-ativa não acionou as regras de *firewall-drop* e/ou *host-deny* para a origem do serviço. Todos os parâmetros relatados são apresentados na figura 5.5.

Figura 5.5: Resposta do OSSEC WUI ao ataque NMAP direcionado ao *host* OSSEC Server com endereço IP 172.16.75.253 com relação a porta SSH (22). Fonte: autor

Observou-se que o comportamento do HIDS OSSEC por meio da sua interface gráfica OSSEC WUI para a varredura da porta HTTP foi positiva, responsiva e informativa, retornando por meio da regra 31101, listada pelo arquivo web_rules.xml, a mensagem: *Web server 400 error code*.

O OSSEC WUI apresentou um massivo de 8 eventos desta regra bem como a origem da tentativa de conexão que retornou o erro 400 na requisição HTTP (por meio do parâmetro *Src. IP*), facilitando assim a identificação das suscetíveis tentativas de processamento da solicitação de conexão.

Observa-se uma informação bastante relevante que se repete durante todo o massivo via retorno da regra 31101, seja para a tentativa de conexão via GET, POST, WISA ou PROPFIND, que é a seguinte mensagem: "...*Nmap Scripting Engine*...". O surgimento dessa informação evidencia a detecção de utilização de um *Port Scanning* para a varredura daquele serviço que, associado ao massivo que apresenta uma média de quase 1 requisição a cada 3 segundos, tendo em vista o período de operação do NMAP que foi de pouco mais de 20 segundos, uma configuração maliciosa ao *host*. Todos os parâmetros relatados são apresentados na figura 5.6

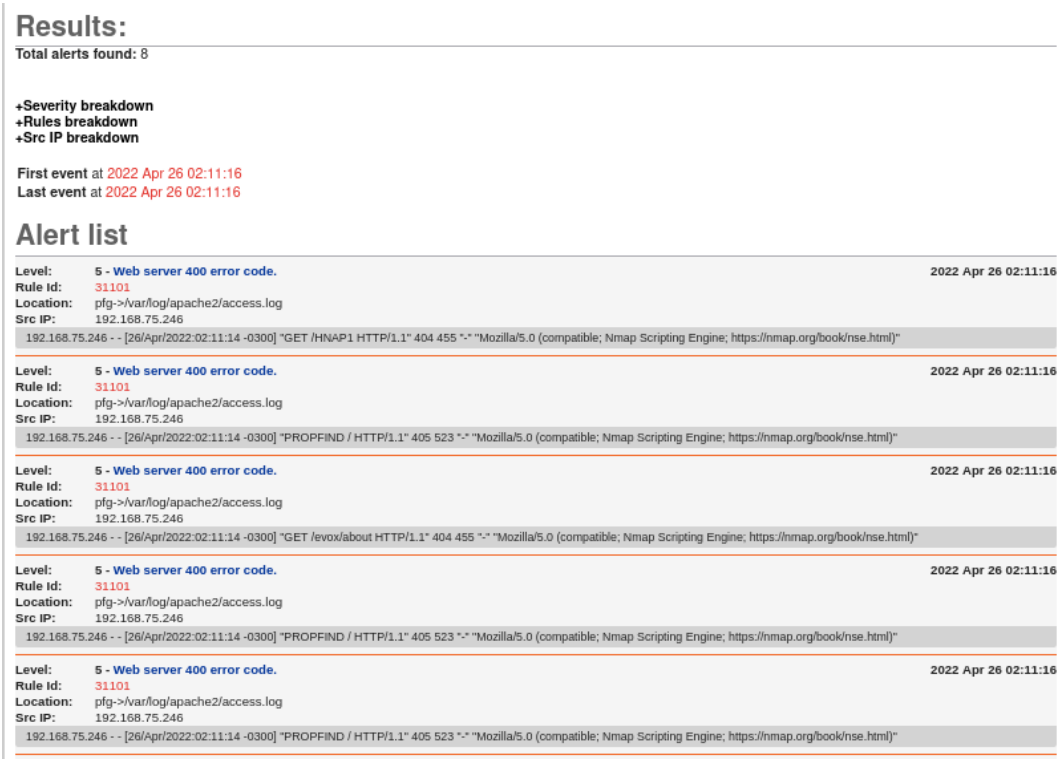

Figura 5.6: Resposta do OSSEC WUI ao ataque NMAP direcionado ao *host* OSSEC Server com endereço IP 172.16.75.253 com relação a porta HTTP (80). Fonte: autor

5.1.1.2 *HOST* WIN-USR-002 IP DHCP: 192.168.75.245

Para o *host* WIN-USR-002, o panorama 5.1 foi executado as 02:57 do dia 26/04/2022 e através dele foi possível identificar que o serviço SSH estava aberto na porta 22.

Nesta identificação foi possível verificar quais algoritmos são utilizados na chave de acesso SSH pelo *host* e qual sistema operacional este *host* opera. Foi possível verificar a quantidade de saltos necessários até que se chegue ao *host* e consequentemente seu gateway. Todos os parâmetros relatados são apresentados na figura 5.4

Observou-se que o comportamento do HIDS OSSEC por meio da sua interface gráfica OSSEC WUI para a varredura da porta SSH foi positiva, responsiva e informativa, retornando por meio da regra 18107 e 18149, listado pelo arquivo msauth_rules.xml, as mensagens: *Windows Logon Success* e *Windows User Logoff* respectivamente.

Houveram registros de 15 eventos (a figura 5.7 apresenta 16 eventos, contudo descarta-se da contagem o evento de regra 18113) envolvendo estas duas regras que apresentam registros de *login* e *logout*. Isto implica em um cenário onde tem-se um intervalo de mais de 1 evento/segundo, dado que a operação NMAP durou pouco mais de 12 segundos, tornando deveras suficiente para se suspeitar de um evento de *Port Scanning* tendo em vista que não foi uma massiva tentativa de login e senha e sim uma abertura e fechamento de conexão por meio de resposta a pacotes possivelmente de varredura da rede. Todos os parâmetros relatados são apresentados na figura 5.7.

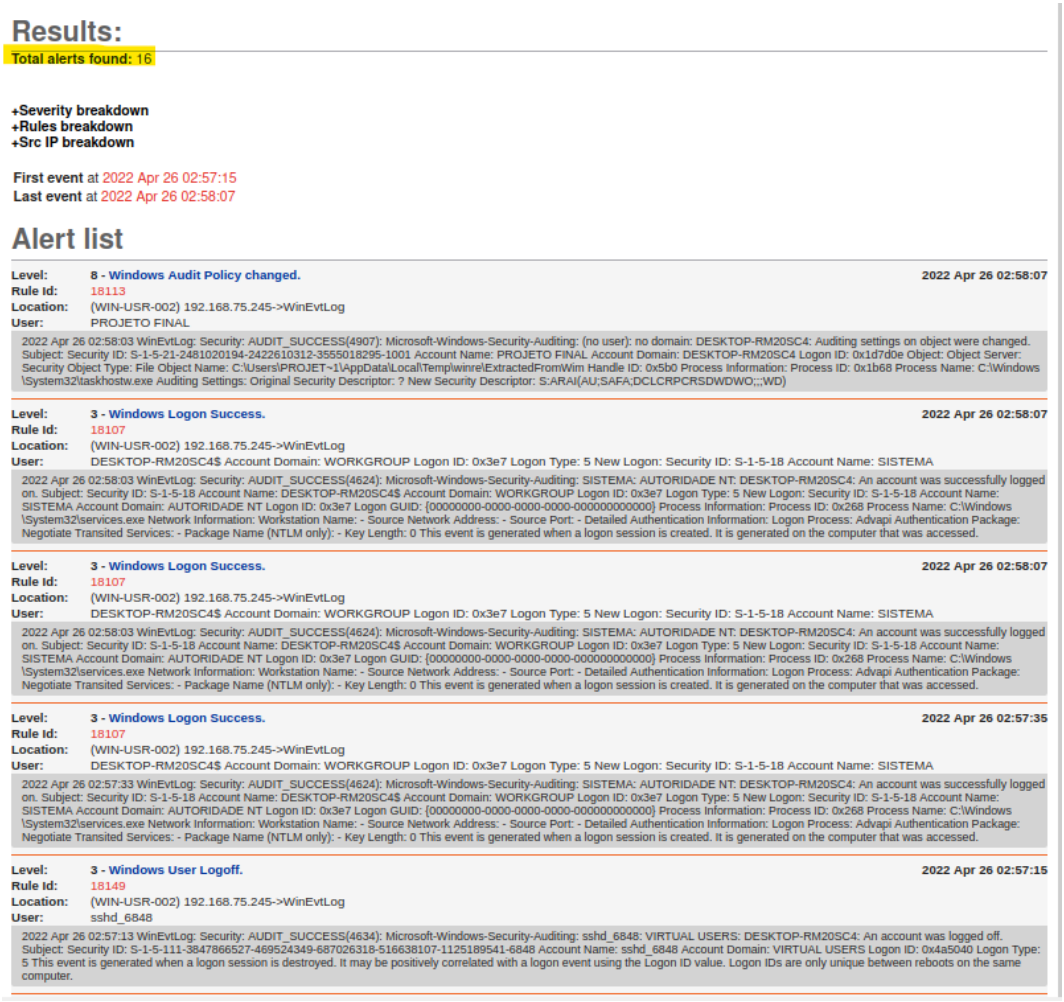

Figura 5.7: Resposta do OSSEC WUI ao ataque NMAP direcionado ao *host* WIN-USR-001 com endereço IP 192.168.75.245 com relação a porta SSH (22). Fonte: autor

5.2 PANORAMA 2 - SSH *BRUTE FORCE*

A tática de *Brute Force* é utilizada em processos iniciais de tentativa de intrusão de redes e *host* por usar técnicas básicas de tentativa e erro de forma exaustiva, até que se tenha uma reativa, seja de êxito, *drop* ou bloqueio por parte do *host*.

Por meio da máquina INTRA-ATTACK, operando via endereço IP 192.168.75.246 fornecido via DHCP

pela interface USR do Firewall pfSense e posicionado na rede proposta pela figura 4.1 como um **Atacante Interno**, realiza-se uma tentativa não autorizada de acesso via SSH por meio de técnica *brute force* em direção ao Servidor OSSEC e ao *host* MGMMT, listados pela tabela 5.1. Foi utilizado o recurso **Hydra** com auxílio da sua interface gráfica **xHydra** a qual funcionamento foi detalhado pela seção 4.2.8.

Os parâmetros repassados para a ferramenta são detalhados a seguir:

Tabela 5.1: Parâmetros necessários para operação da ferramenta Hydra para realização de ataque de intrusão via SSH, com base no cenário proposto pela figura 4.2

Dispositivo	IP	Protocolo	Porta	User	Arquivo de Senhas
OSSEC-SERVER	172.16.75.253	SSH	22	root	/usr/share/john/password.list
MGMMT	10.10.75.253	SSH	22	root	/usr/share/john/password.list

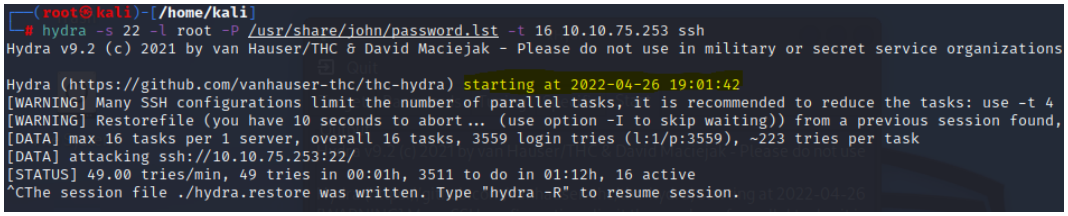

Figura 5.8: Ataque BRUTE FORCE destinado ao *host* MGMMP sob endereço IP: 10.10.75.253 via porta SSH (22) com volumetria de aproximadamente 49 ataques por minuto. Fonte: autor

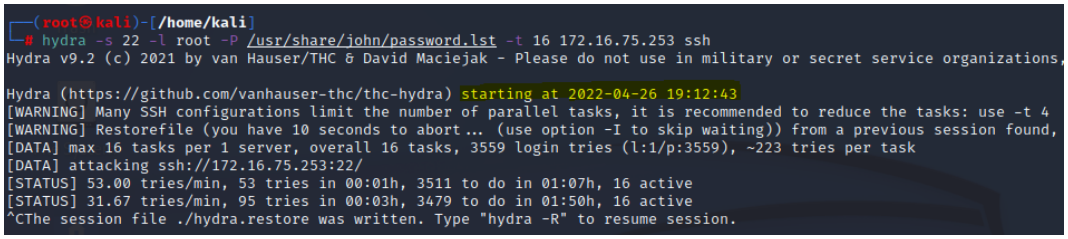

Figura 5.9: Ataque BRUTE FORCE destinado ao *host* OSSEC-SERVER sob endereço IP: 172.16.75.253 via porta SSH (22) com volumetria de aproximadamente 53 ataques por minuto em um dado momento e 32 ataques por minuto em um outro momento. Fonte: autor

5.2.1 RESULTADOS OBTIDOS

5.2.1.1 *HOST* MGMMT IP: 10.10.75.253

Para o host MGMMT, o panorama 5.2 foi executado as 20:01 do dia 26/04/2022 a qual foi realizado uma tentativa não autorizada de login via porta SSH (22) por meio do usuário **root** e de uma *password list* fornecida pela aplicação *John the Ripper*, disponível de forma nativa no Kali Linux pelo caminho /usr/share/john/password.lst, contendo 3559 linhas com senhas mais comuns utilizadas por usuários ao redor do globo.

Os parâmetros repassados ao hydra solicitavam que fossem feitas um máximo de 16 tarefas por servidor, isto é, como apenas operávamos com um único host, eram permitidas um máximo de 16 tarefas. Como citado acima, o arquivo *password.lst* possui 3559 entradas, o que resultou em aproximadamente 223 entradas por tarefa.

Obteve-se um ataque coordenado de pouco mais de um minuto com uma volumetria de 49 tentativas por minuto, resultando em pouco mais de 1 tentativa a cada 2 segundos. Todos os parâmetros relatados são apresentados na figura 5.8.

Observou-se que o comportamento do HIDS OSSEC por meio da sua interface gráfica OSSEC WUI foi postiva e informativa por meio das regras 5720 (listada pelo arquivo sshd_rules.xml, a mensagem: *Multiple SSHD authentication failures*) e 5551 (listada pelo arquivo pam_rules.xml, a mensagem: *Multiple failed logins in a small period of time*).

O OSSEC WUI registrou um total de 2 eventos da regra 5720 e 2 eventos da regra 5551, ambas classificadas com nível 10, alto risco e sempre acionadas quando há 6 ou mais tentativas falhas de autenticação na porta SSH (22) (no caso da regra 5720) e em um curto espaço de tempo (no caso da regra 551).

O OSSEC WUI apresentou informações relevantes para as duas regras, como IP de origem, *user*, porta e protocolo, o que facilita uma análise de que tipo de ocorrência está acontecendo. Neste caso por exemplo, é possível deduzir há um evento na rede originário do endereço de IP 192.168.75.246 relatado pelo *host* MGMMT de endereço IP 10.10.75.253 a qual há suscetivas tentativas de login via protocolo SSH (22) mal sucedidas via um usuário conhecido e existente naquele *host*, evidenciando um possível *brute force*. Todos os parâmetros relatados são apresentados nas figuras 5.10 e 5.11.

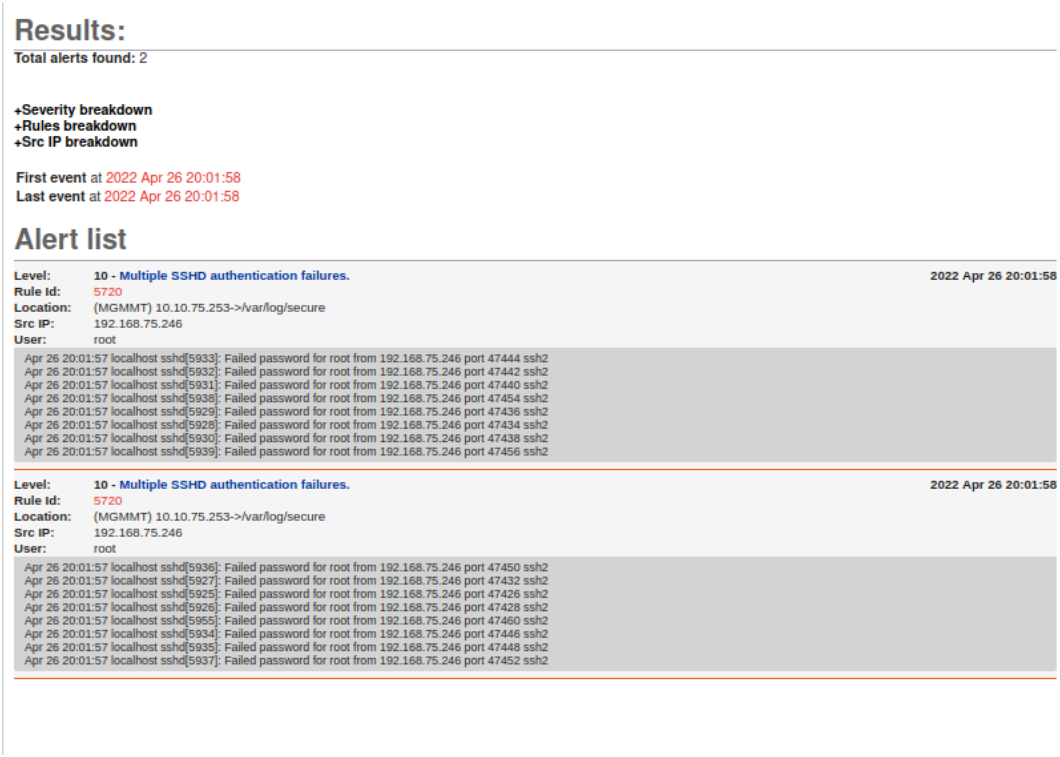

Figura 5.10: Resposta OSSEC WUI ao ataque BRUTE FORCE direcionado ao *host* MGMMT com endereço IP 10.10.75.253 via porta SSH (22) registrado sob ótica da regra 5720 regido pelo arquivo de regras pam_rules.xml
Fonte: autor

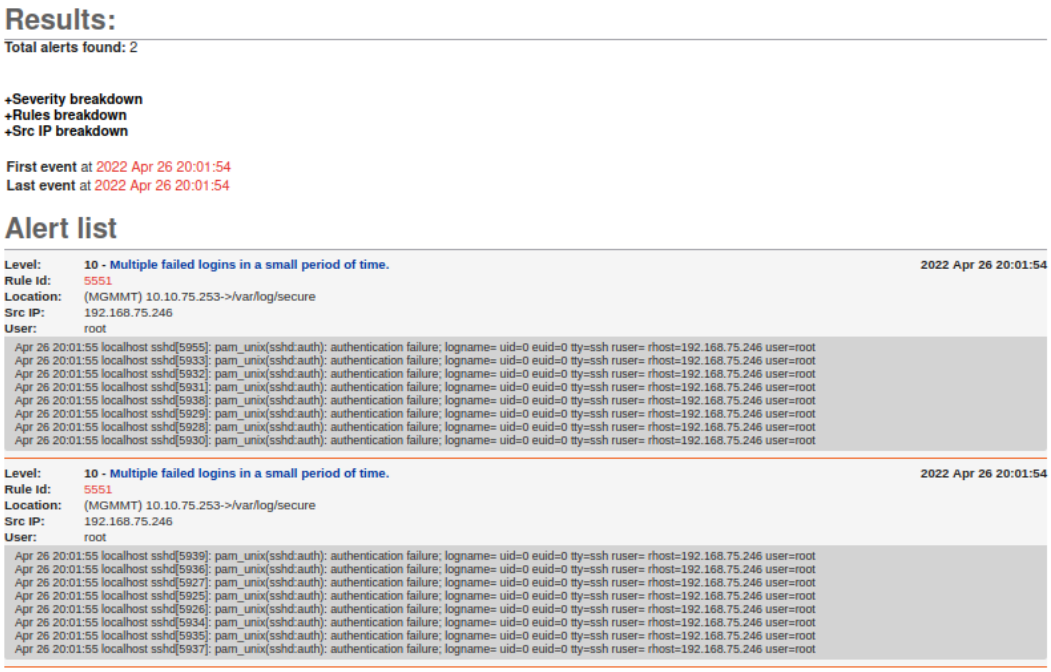

Figura 5.11: Resposta OSSEC WUI ao ataque BRUTE FORCE direcionado ao *host* MGMMT com endereço IP 10.10.75.253 via porta SSH (22) registrado sob ótica da regra 5551 regido pelo arquivo de regras sshd_rules.xml Fonte: autor

Por meio das regras da resposta ativa, o HIDS OSSEC se comportou de forma reativa em face a detecção do ataque, realizando o *DROP* do IP de origem via criação de regra pelo *firewall-drop.sh* no **IPTABLES** tanto quanto, como via de dupla seguridade, contextualizado na sessão 4.2.7, relizando a escrita via *hosts-deny.sh* do IP de origem do ataque no arquivo **hosts.deny**.

Essa ação prevista pela regra <*active-response*> tem duração padrão de 10 minutos mas podem ser alteradas de acordo com os interesses da do administrador da rede via arquivo `ossec.conf`. Todos os parâmetros relatados são apresentados nas figuras 5.12 e 5.13

Como teste da aplicação das regras, foi feito, as 20:04 do dia 26/04/2022 uma nova tentativa não autorizada de login via porta SSH (22) com as mesmas credencias apresentadas no primeiro cenário. Observou-se o retorno de *timeout* pela ferramenta hydra do atacante, concluindo assim a efetividade da resposta ativa por parte do agente OSSEC do *host* MGMMT e ratificando o apresentado pelas figuras 5.12 e 5.13. Todos os parâmetros relatados podem ser observados pela figura 5.14

```
[root@localhost etc]# iptables -L -n -v | grep "192.168.75.246"
  355 33652 DROP       all  --  *      *       192.168.75.246       0.0.0.0/0
    0     0 DROP       all  --  *      *       192.168.75.246       0.0.0.0/0
[root@localhost etc]#
```

Figura 5.12: Resposta da regra <ACTIVE-RESPONSE> com relação a ao ataque BRUTE FORCE direcionado ao *host* MGMMT com endereço IP 10.10.75.253 via porta SSH (22) sob ótica da chamada do script *firewall-drop.sh*. Fonte: autor

```
[root@localhost etc]# cat hosts.deny
#
# hosts.deny    This file contains access rules which are used to
#               deny connections to network services that either use
#               the tcp_wrappers library or that have been
#               started through a tcp_wrappers-enabled xinetd.
#
#               The rules in this file can also be set up in
#               /etc/hosts.allow with a 'deny' option instead.
#
#               See 'man 5 hosts_options' and 'man 5 hosts_access'
#               for information on rule syntax.
#               See 'man tcpd' for information on tcp_wrappers
#
ALL:192.168.75.246
[root@localhost etc]#
```

Figura 5.13: Resposta da regra <ACTIVE-RESPONSE> com relação a ao ataque BRUTE FORCE direcionado ao *host* MGMMT com endereço IP 10.10.75.253 via porta SSH (22) sob ótica da chamada do script *host-deny.sh*. Fonte: autor

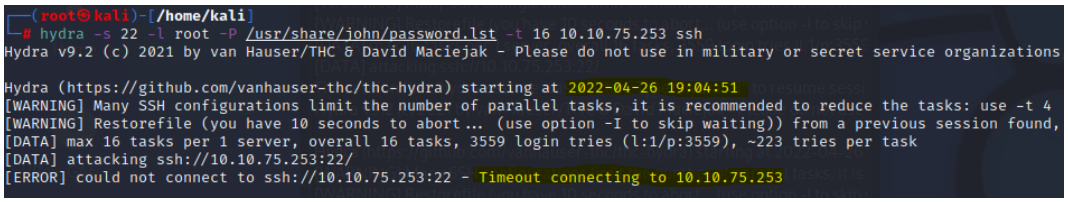

Figura 5.14: Retorno de *timeout* apresentado pela ferramenta HYDRA por parte do atacante quando direcionado ao *host* MGMMT. Fonte: autor

5.2.1.2 *HOST* OSSEC SERVER IP: 172.16.75.253

Para o host OSSEC-SERVER, o panorama 5.2 foi executado as 20:12 do dia 26/04/2022 a qual foi realizado uma tentativa não autorizada de login via porta SSH (22) por meio do usuário **root** e de uma *password list* fornecida pela aplicação *John the Ripper*, disponível de forma nativa no Kali Linux pelo caminho /usr/share/john/password.lst, contendo 3559 linhas com senhas mais comuns utilizadas por usuários ao redor do globo.

Os parâmetros repassados ao hydra solicitavam que fossem feitas um máximo de 16 tarefas por servidor, isto é, como apenas operávamos com um único host, eram permitidas um máximo de 16 tarefas. Como citado acima, o arquivo *password.lst* possui 3559 entradas, o que resultou em aproximadamente 223 entradas por tarefa.

Obteve-se um ataque coordenado de pouco mais de três minutos com uma volumetria de 53 tentativas por minuto no primeiro minuto e aproximadamente 32 tentativas por minuto, sinalizando uma possível atuação do HIDS OSSEC. Todos os parâmetros relatados são apresentados na figura 5.9

Observou-se que o comportamento do HIDS OSSEC por meio da sua interface gráfica OSSEC WUI foi postiva e informativa por meio das regras 5720 (listada pelo arquivo sshd_rules.xml, a mensagem: *Multiple SSHD authentication failures*) e 5551 (listada pelo arquivo pam_rules.xml, a mensagem: *Multiple failed*

logins in a small period of time).

O OSSEC WUI registrou um total de 2 eventos da regra 5720 e 2 eventos da regra 5551, ambas classificadas com nível 10, alto risco e sempre acionadas quando há 6 ou mais tentativas falhas de autenticação na porta SSH (22) (no caso da regra 5720) e em um curto espaço de tempo (no caso da regra 551).

O OSSEC WUI apresentou informações relevantes para as duas regras, como IP de origem, *user*, porta e protocolo, o que facilita uma análise de que tipo de ocorrência está acontecendo. Neste caso por exemplo, é possível deduzir há um evento na rede originário do endereço de IP 192.168.75.246 relatado pelo *host* OSSEC-SERVER de endereço IP 172.16.75.253 a qual há suscetivas tentativas de login via protocolo SSH (22) mal sucedidas via um usuário conhecido e existente naquele *host*, evidenciando um possível *brute force*. Todos os parâmetros relatados são apresentados nas figuras 5.15 e 5.16.

Figura 5.15: Resposta OSSEC WUI ao ataque BRUTE FORCE direcionado ao *host* OSSEC-SERVER com endereço IP 10.10.75.253 via porta SSH (22) registrado sob ótica da regra 5720 regido pelo arquivo de regras pam_rules.xml
Fonte: autor

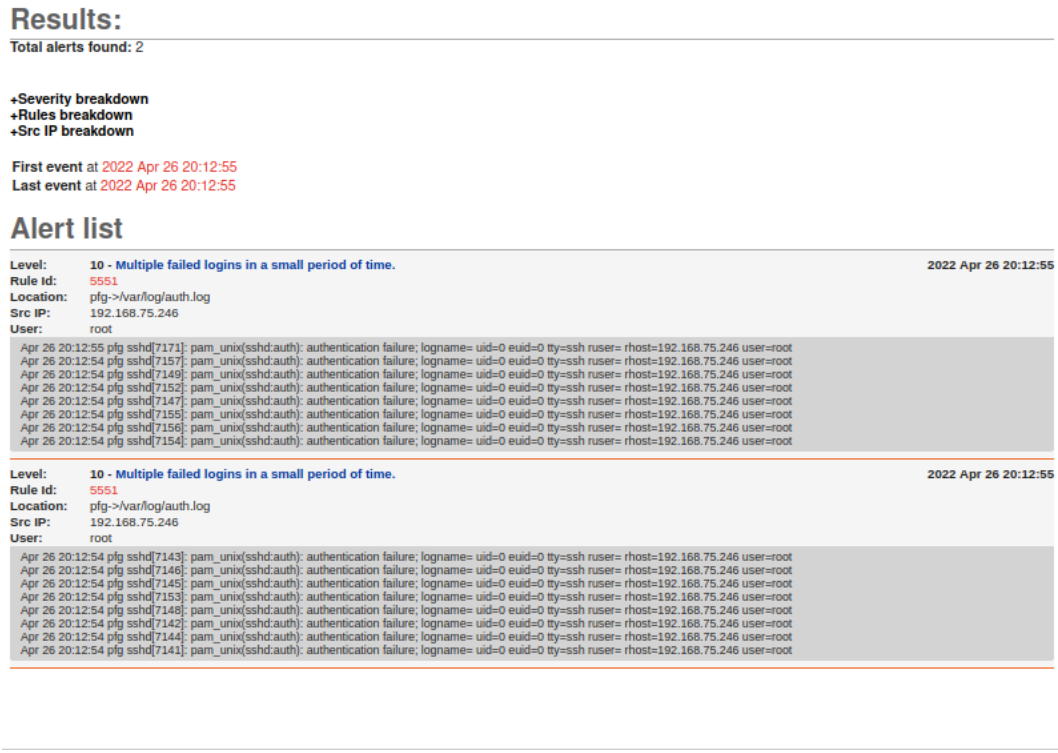

Figura 5.16: Resposta OSSEC WUI ao ataque BRUTE FORCE direcionado ao *host* OSSEC-SERVER com endereço IP 10.10.75.253 via porta SSH (22) registrado sob ótica da regra 5551 regido pelo arquivo de regras sshd_rules.xml
Fonte: autor

Por meio das regras da resposta ativa, o HIDS OSSEC se comportou de forma reativa em face a detecção do ataque, realizando o *DROP* do IP de origem via criação de regra pelo *firewall-drop.sh* no **IPTABLES** tanto quanto, como via de dupla seguridade, contextualizado na sessão 4.2.7, relizando a escrita via *hosts-deny.sh* do IP de origem do ataque no arquivo **hosts.deny**.

Essa ação prevista pela regra <*active-response*> tem duração padrão de 10 minutos mas podem ser alteradas de acordo com os interesses da do administrador da rede via arquivo `ossec.conf`. Todos os parâmetros relatados são apresentados nas figuras 5.17 e 5.18

Como teste da aplicação das regras foi feito, as 20:18 do dia 26/04/2022 uma nova tentativa não autorizada de login via porta SSH (22) com as mesmas credencias apresentadas no primeiro cenário. Observou-se o retorno de *timeout* pela ferramenta hydra do atacante, concluindo assim a efetividade da resposta ativa por parte do servidor OSSEC e ratificando o apresentado pelas figuras 5.17 e 5.18. Todos os parâmetros relatados podem ser observados pela figura 5.19

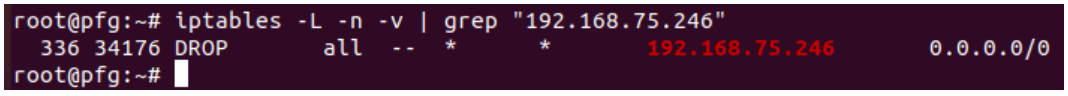

Figura 5.17: Resposta da regra <ACTIVE-RESPONSE> com relação a ao ataque BRUTE FORCE direcionado ao *host* OSSEC-SERVER com endereço IP 172.16.75.253 via porta SSH (22) sob ótica da chamada do script *firewall-drop.sh*.
Fonte: autor

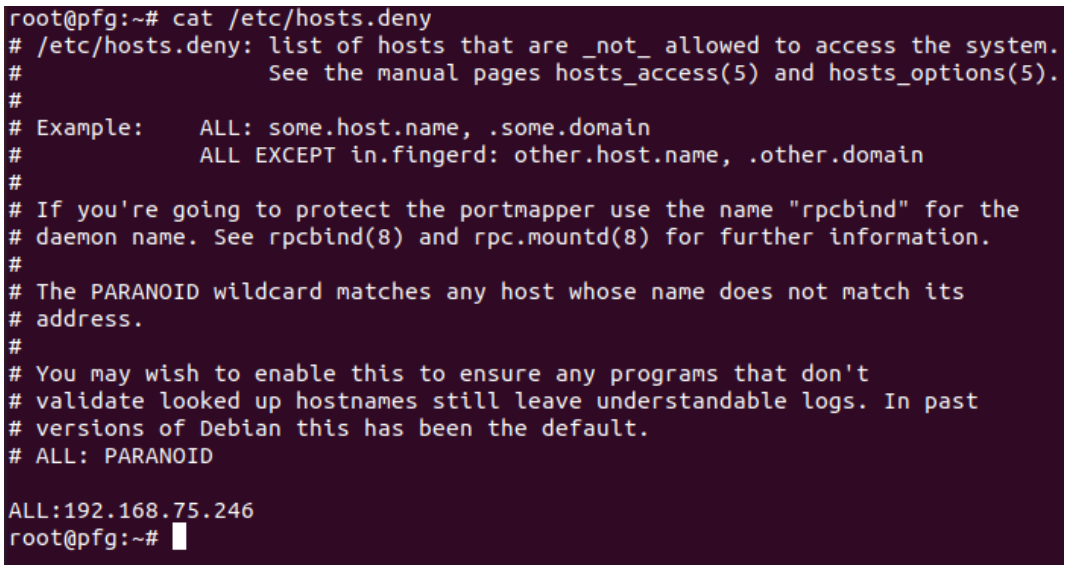

Figura 5.18: Resposta da regra <ACTIVE-RESPONSE> com relação a ao ataque BRUTE FORCE direcionado ao *host* OSSE-SERVER com endereço IP 172.16.75.253 via porta SSH (22) sob ótica da chamada do script *host-deny.sh*. Fonte: autor

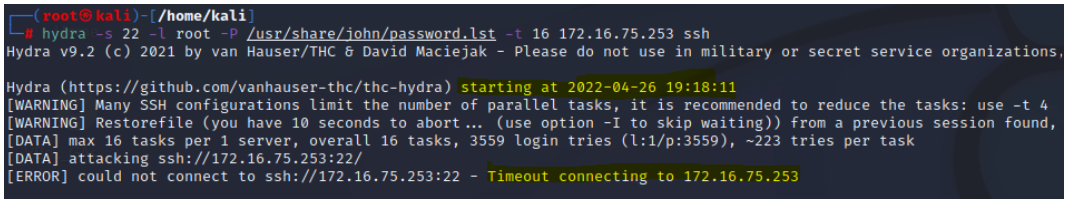

Figura 5.19: Retorno de *timeout* apresentado pela ferramenta HYDRA por parte do atacante quando executado em direção ao servidor OSSEC. Fonte: autor

5.3 PANORAMA 3 - DETECÇÃO DE *ROOTKITS*

Este panorama foi proposto pela Engenheira Ana Paula Alarcão em seu projeto [Alarcão 2021] e a sua utilização visa testar a capacidade de detecção de rootkits por parte HIDS OSSEC.

É importante trazer para este estudo a abordagem introduzida sobre a aplicabilidade de rootkits pela Engenheira Ana Paula Alarcão em seu projeto [Alarcão 2021], a qual a mesma cita que um *rootkit* é "comumente utilizado com a finalidade de se esconder e não somente, esconder a presença de arquivos, programas, acessos, serviços, *scripts* ou até mesmo malwares". Os principais tipos de *rootkits* são:

- *User-mode rootkits*;
- ***Kernel-mode rootkits***;
- *Firmware and hardware rootkits*;
- *Bootloader rootkit*;
- *Application rootkit*.

Para este panorama, o *host* OSSEC-SERVER sob endereço IP 172.16.75.253 será exposto a um *Kernel-mode rootkits*, que é um rootkit modo *kernel* que se esconde de listas de processos do *kernel* e também de processos selecionados do módulo nativo, módulo este que provê informações sobre quais processos se encontram em execução no sistema [Alarcão 2021].

Entenda este cenário como uma intrusão bem sucedida do atacante via porta SSH (que como visto pelos panoramas 5.1 e 5.2 se encontra exposta e explorável) ao Servidor OSSEC-SERVER.

Com essa intrusão bem sucedida, deseja-se esconder scripts de alteração de arquivos para causar uma indisponibilidade futura do servidor e de seus serviços por meio de criptografia de arquivos e/ou indisponibilidade destes, tendo em vista que o servidor apresenta portas HTTP (80) abertas dentro de sua estrutura.

Para o *rootkit*, foi utilizado o Diamorphine, um *rootkit Kernel-mode* para kernels Linux 2.6.X até 5.x tanto quanto ARM64. Este rootkit, como abordado por [Alarcão 2021], esconde e revela qualquer processo enviando um sinal 31 pelo comando kill() a nível de sistema operacional no linux.

Para carregá-lo no sistema, basta realizar um `git` do seu repositório padrão, situado em [m0nad 2021] através do site https://github.com/m0nad/Diamorphine e executar, dentro da pasta obtida pelo `git`, sua compilação via comando `make` e o load dos módulos via comando `insmod diamorphine.ko`.

5.3.1 RESULTADOS OBTIDOS

Realizou-se o carregamento do rootkit no *host* OSSEC-SERVER conforme descrito e proposto pelo panorama 5.3. Inicialmente manteve-se o *rootkit* escondido, como pode ser observado pela figura 5.20. O HIDS OSSEC **não foi capaz** de detectar a presença do malware em seu sistema, apresentando assim sua primeira falha em relação ao EDR Wazuh, proposto por [Alarcão 2021].

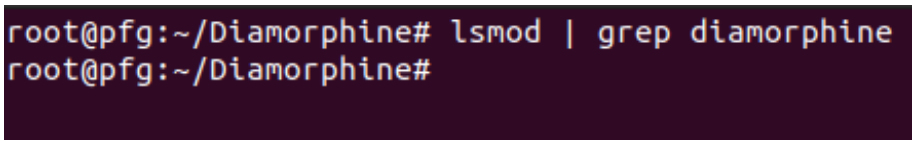

Figura 5.20: Rootkit kernel-mode Diamorphine operando de forma "invisível"dentro do host OSSEC-SERVER. Fonte: autor

Posteriormente, realizou-se a alteração do status do *rootkit* de invisível para visível. Para que isso ocorra, basta enviar um sinal 63 via comando `kill`, como pode ser observado pela figura 5.21. O HIDS OSSEC **não foi capaz** de detectar a presença do malware em seu sistema, apresentando assim sua segunda falha em relação ao EDR Wazuh, proposto por [Alarcão 2021].

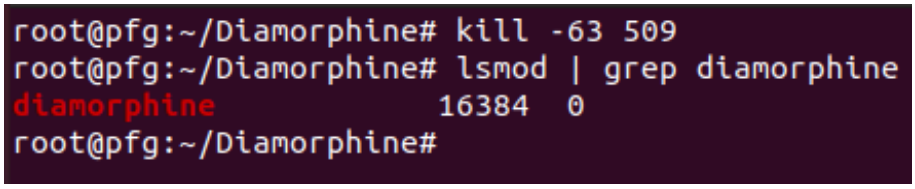

Figura 5.21: Rootkit kernel-mode Diamorphine operando de forma "visível"dentro do host OSSEC-SERVER. Fonte: autor

Apesar de se prever uma detecção por parte do HIDS OSSEC para *rootkits* tendo em vista que o mesmo utiliza-se de chamadas `getsid()` e `kill()` para verificar se algum **PID** está sendo utilizado ou não, foi constatado a incapacidade de reconhecimento **do Diamorphine, rootkit modo kernel**, mesmo sendo ele visível.

Isto ocorre pelo fato do HIDS trabalhar de forma reativa e não responsiva, isto é, somente é capaz de detectar malwares a qual ele já possui o conhecimento de operação e alojamento, o que não é o caso do Diamorphine e que pode ser observado pelo arquivo `/var/ossec/etc/shared/rootkit_files.txt`.

Por fim, os estudos propostos pelas ferramentas NMAP (panorama 5.1), BRUTE FORCE (panorama 5.2 e ROOTKITS (panorama 5.3) retrataram os cenários propostos pela *Cyber Kill Chain*, respectivamente, de Reconhecimento, Armazenamento e Entrega.

O NMAP, utilizado para realizar a varredura da rede por meio de feature Port Scanning faz o reconhecimento de vulnerabilidades do cenário, o BRUTE FORCE a tentativa de alojamento no hospedeiro e ROOTKIT o êxito da etapa e a entrega no hospedeiro do malware, *script*, conjunto de instruções etc.

Apesar dos panoramas constatarem apenas os cenários iniciais da Cyber Kill Chain, é evidente os perigos destas etapas, haja vista que alojado em uma máquina, o atacante pode realizar a movimentação lateral realizando assim de forma facilitada as demais etapas listadas pela mesma modelagem.

6 CONCLUSÃO

O projeto trouxe um cenário local, controlado e segmentado para avaliação de sistema de detecção de intrusão baseado em *host* de código aberto, a qual foi possível averiguar seu comportamento e resultados na função a qual está destinado a desempenhar. A solução proposta foi o HIDS OSSEC e o ambiente desenvolvido utilizou de um cenário multiplataforma, simulado e emulado por ferramenta aberta e livremente distribuída de simulação denominado GNS-3 com auxílio de um virtualizador de mesmo parâmetro, o VirtualBox.

O ambiente foi desenvolvido de forma a proporcionar a simulação de um cenário infectado onde o atacante já se encontra alojado dentro da rede via um determinado *host* a qual o mesmo possui total controle. A partir deste *host*, o atacante direciona ataques a *hosts* estratégicos na tentativa de obter o mesmo controle de acesso que já obteve anteriormente. Objetiva-se assim analisar o comportamento da ferramenta HIDS OSSEC, tendo em vista que por mais que o atacante se encontra alojado na rede interna, ele não possui conhecimento de credenciais internas para que possa tomar posse de outros *hosts* e realizar os ataques em busca dessas informações.

Como validador do reconhecimento e tratativas do ataque, foi utilizado a ferramenta gráfica de análise de logs do próprio HIDS OSSEC, o OSSEC WUI. Não somente, utilizou-se de artefatos da plataforma Linux como explorado na sessão anterior, podendo então este autor registrar os eventos ocorridos a cada estilo de ataque direcionado a cada host envolvido na rede.

Assim sendo, objetivando exercer um acesso não autorizado nos *hosts* do ambiente, foram propostos três cenários de intrusão ao HIDS a qual permitiu avaliar seu comportamento e que tipo de retorno reativo o mesmo tomou quando acionado.

Como faz parte de uma contribuição ao estudo desenvolvido pela Engenheira Ana Paula Alarcão via [Alarcão 2021], pôde-se concluir que o cenário proposto **valida a funcionalidade da ferramenta OSSEC com um HIDS mas não o valida como um EDR**. Isto pois os sistemas HIDS's são ferramentas baseadas em assinaturas e agem de forma reativa, isto é, somente quando acionadas. Já os sistemas EDRs são ferramentas responsivas que combinam múltiplos indicadores para atingir uma resposta mais conclusiva sobre a ocorrência de um fato.

Por fim, vale ressaltar que o projeto OSSEC é a base do sistema de detecção de intrusão do projeto EDR Wazuh e isto **não afirma que os mesmos são equivalentes**, mas sim que o EDR Wazuh utiliza-se das funcionalidades propostas pelo HIDS OSSEC para compor a sua estrutura de atuação. A comprovação deste fato é a capacidade de detecção de *malwares* desconhecidos por parte do EDR Wazuh que não foi possível pelo HIDS OSSEC, vide panorama 5.3.

7 TRABALHOS FUTUROS

O HIDS OSSEC permite a integração com alguns plugins que podem facilitar a sua operacionalidade, como por exemplo o *TickStream.KeyID*, que é uma solução de autenticação do Windows, o *Slack* que é uma aplicação de mensagens para local de trabalho a qual a ferramenta poderia encaminhar *log events* e o *Rule Generator*, um gerador de regras com base em funcionalidade de aplicações. É importante salientar, em complemento, que todas essas integrações podem ser encontradas no site do OSSEC em sua aba de extensões (OSSEC Extensions).

Esta integrações, apresentadas anteriormente, podem ser bastante exploradas em trabalhos futuros para que possam encorpar ainda mais as funcionalidades presentes no HIDS OSSEC. Não somente, sugere-se como trabalhos futuros:

- Testes de outras ferramentas HIDS de código aberto como efeito comparativo ao comportamento do HIDS OSSEC, realizando este ou outro ambiente, como por exemplo, um ambiente em nuvem;
- Testes neste ou em outro ambiente da ferramenta HIDS OSSEC com ataque coordenado via rede externa, realizando este ou outro ambiente, como por exemplo, um ambiente em nuvem;
- Teste de outros tipos de intrusão via outras portas como Telnet (23) e FTP (21) e testes de Negação de serviço (*Denial-of-Service*), interceptação de comunicação (*Main-in-the-Middle*) dentre outros;
- Testes em cenário com ambiente controlado por domínio, seja *Active Directory* ou LDAP;
- Expansão de funcionalidades do OSSEC com o OSSEC+ que traz *features* mais robustas, como a aprendizagem de máquina, um conjunto de milhares de novas regras e integração com o ELK.

REFERÊNCIAS BIBLIOGRÁFICAS

Agrawal, Banerjee e Sharma 2017 AGRAWAL, S.; BANERJEE, S.; SHARMA, S. Privacy and security of aadhaar a computer science perspective. EPW, 2017.

Alarcão 2021 ALARCÃO, A. P. A. Implementação e analise de resultados de ferramenta de detecção e resposta para proteção de endpoints em ambiente controlado. In: DEPARTAMENTO DE ENGENHARIA ELETRICA, UNIVERSIDADE DE BRASILIA. *Monografia de Projeto Final de Graduação*. [S.l.], 2021.

Amankwa, Loock e Kritzinger 2014 AMANKWA, E.; LOOCK, M.; KRITZINGER, E. A conceptual analysis of information security education, information security training and information security awareness definitions. In: *The 9th International Conference for Internet Technology and Secured Transactions (ICITST-2014)*. [S.l.: s.n.], 2014. p. 248–252.

Anwar et al. 2017 ANWAR, S.; ZAIN, J. M.; ZOLKIPLI, M. F.; INAYAT, Z.; KHAN, S.; ANTHONY, B.; CHANG, V. From intrusion detection to an intrusion response system: fundamentals, requirements, and future directions. *Algorithms*, Multidisciplinary Digital Publishing Institute, v. 10, n. 2, p. 39, 2017.

APT: Advanced persistent threats 2020 APT: Advanced persistent threats. 2020. <https://www.hscbrasil.com.br/apt/>. (Accessed on 05/12/2022).

Assante e Lee 2015 ASSANTE, M. J.; LEE, R. M. The industrial control system cyber kill chain. *SANS Institute InfoSec Reading Room*, v. 1, 2015.

Bace 2000 BACE, R. G. *Intrusion detection*. [S.l.]: Sams Publishing, 2000.

Biggest Data Breaches of 2018 BIGGEST Data Breaches of 2018. <https://www.businessinsider.com/data-hacks-breaches-biggest-of-2018-2018-12>. (Accessed on 02/09/2022).

Bray, Cid e Hay 2008 BRAY, R.; CID, D.; HAY, A. *OSSEC host-based intrusion detection guide*. [S.l.]: Syngress, 2008.

Campos 2006 CAMPOS, A. O que é uma distribuição linux. *BR-Linux. Florianópolis*, 2006.

CERT.br 2020 CERT.BR. *CERT.br - Centro de Estudos, Resposta e Tratamento de Incidentes de Segurança no Brasil*. 2020. <https://www.cert.br/>. (Accessed on 04/29/2022).

Club 2021 CLUB, H. *Step by Step Guide to Install OSSEC HIDS on Ubuntu*. 2021. <https://www.hackerxone.com/2021/09/19/step-by-step-guide-to-install-ossec-hids-on-ubuntu-20-04-lts/>. (Accessed on 04/22/2022).

CNN Brasil | Ataque cibernético - Notícias e tudo sobre CNN Brasil | Ataque cibernético - Notícias e tudo sobre. <https://www.cnnbrasil.com.br/tudo-sobre/ataque-cibernetico/>. (Accessed on 02/09/2022).

Dargahi et al. 2019 DARGAHI, T.; DEHGHANTANHA, A.; BAHRAMI, P. N.; CONTI, M.; BIANCHI, G.; BENEDETTO, L. A cyber-kill-chain based taxonomy of crypto-ransomware features. *Journal of Computer Virology and Hacking Techniques*, Springer, v. 15, n. 4, p. 277–305, 2019.

Denning 1987 DENNING, D. E. An intrusion-detection model. *IEEE Transactions on software engineering*, IEEE, n. 2, p. 222–232, 1987.

DistroWatch 2022 DISTROWATCH. *DistroWatch.com: Put the fun back into computing. Use Linux, BSD.* 2022. <https://distrowatch.com/dwres.php?resource=popularity>. (Accessed on 04/15/2022).

Ferreira et al. 2012 FERREIRA, M.; ROCHA, T. de S.; BREVES, G.; MARTINS, E. F.; SOUTO, E. Análise de vulnerabilidades em sistemas computacionais modernos: Conceitos, exploits e proteções. *in Minicursos XII Simpósio em Segurança da Informação e de Sistemas Computacionais, 1st ed., Porto Alegre: Sociedade Brasileira de Computação*, p. 1–50, 2012.

Freund, Sembay e Macedo 2019 FREUND, G.; SEMBAY, M.; MACEDO, D. Proveniência de dados e segurança da informação: relações interdisciplinares no domínio da ciência da informação. *Revista Ibero-Americana de Ciência da Informação*, v. 12, p. 807–825, 09 2019.

GNS3 GNS3. *GNS3 Windows Install | GNS3 Documentation*. <https://docs.gns3.com/docs/getting-started/installation/windows/>. (Accessed on 04/20/2022).

GUSTEDT e QUINSON 2009 GUSTEDT, E. J. J.; QUINSON, M. Experimental methodologies for large-scale systems: A survey. *Parallel Processing Letters*, v. 19, n. 3, p. 399–418, 2009.

Hat 2018 HAT, R. *Virtualização*. 2018. <https://www.redhat.com/pt-br/topics/virtualization>. (Accessed on 05/12/2022).

InfoEscola 2006 INFOESCOLA. *Windows 7 - Windows Seven - Sistemas operacionais - InfoEscola*. 2006. <https://www.infoescola.com/informatica/windows-7/>. (Accessed on 04/16/2022).

Insights 2022 INSIGHTS, G. P. *Trend Micro Cloud One Reviews, Ratings, and Features - Gartner 2022*. 2022. <https://www.gartner.com/reviews/market/cloud-workload-protection-platforms/vendor/trend-micro/product/trend-micro-cloud-one>. (Accessed on 05/01/2022).

ITGovernance 2019 ITGOVERNANCE. *IT GOVERNANCE GREEN PAPER*. 2019. <https://www.itgovernance.co.uk/files/Infosec_101v1.1.pdf>. (Accessed on 03/28/2022).

Lyon 2014 LYON, G. Nmap security scanner. *línea] URL: http://nmap. org/[Consulta: 8 de junio de 2012]*, 2014.

m0nad 2021 M0NAD. *GitHub - m0nad/Diamorphine: LKM rootkit for Linux Kernels 2.6.x/3.x/4.x/5.x (x86/x86_64 and ARM64)*. 2021. <https://github.com/m0nad/Diamorphine>. (Accessed on 04/27/2022).

Machado et al. 2019 MACHADO, R.; KREUTZ, D.; PAZ, G.; RODRIGUES, G. Vazamentos de dados: Histórico, impacto socioeconômico e as novas leis de proteção de dados. In: SBC. *Anais da XVII Escola Regional de Redes de Computadores*. [S.l.], 2019. p. 154–159.

Mallik 2019 MALLIK, A. Man-in-the-middle-attack: Understanding in simple words. *Cyberspace: Jurnal Pendidikan Teknologi Informasi*, v. 2, n. 2, p. 109–134, 2019.

Notes_Wiki 2019 NOTES_WIKI. *CentOS 7.x Install OSSEC agent - Notes_Wiki*. 2019. <https://www.sbarjatiya.com/notes_wiki/index.php/CentOS_7.x_Install_OSSEC_agent>. (Accessed on 04/25/2022).

Patel e Sharma 2017 PATEL, K. C.; SHARMA, P. A review paper on pfsense–an open source firewall introducing with different capabilities & customization. *IJARIIE*, v. 3, p. 2395–4396, 2017.

Poulsen et al. 2005 POULSEN, S. et al. Best practices in the design of a secure control system. 2005.

QWERTYGUY 2018 QWERTYGUY. *Basic of xHydra. Set up target ip and choose the... | by QWERTYGUY | Medium*. 2018. <https://medium.com/@qwertystory/basic-of-xhydra-cf294e9c39d>. (Accessed on 04/25/2022).

Rosa 2012 ROSA, A. C. Engenharia social: o elo mais frágil da segurança nas empresas. *Revista Eletrônica do Alto Vale do Itajaí*, v. 1, n. 2, p. 29–40, 2012.

Santos 2014 SANTOS, E. P. dos. Segurança da informação: Como garantir a integridade, a confidencialidade e a disponibilidade das informações em uma organização educacional privada de teresina/information security: How to ensure integrity, confidentiality and availability of the infor. *Revista FSA (Centro Universitário Santo Agostinho)*, v. 7, n. 1, 2014.

Shurman, Khrais e Yateem 2020 SHURMAN, M.; KHRAIS, R.; YATEEM, A. Dos and ddos attack detection using deep learning and ids. *Int. Arab J. Inf. Technol.*, v. 17, n. 4A, p. 655–661, 2020.

Silveira SILVEIRA, R. B. *História do Microsoft® Windows®*.

Swanagan 2021 SWANAGAN, M. *Intrusion Detection VS Prevention Systems: What's The Difference?* 2021. <https://purplesec.us/intrusion-detection-vs-intrusion-prevention-systems/>. (Accessed on 03/31/2022).

TechTudo 2014 TECHTUDO. *Windows 10, a evolução: relembre como era o software há 30 anos | Notícias | TechTudo*. 2014. <https://www.techtudo.com.br/noticias/2014/10/windows-10-evolucao-relembre-como-era-o-software-ha-30-anos.ghtml>. (Accessed on 04/16/2022).

Tecnologia 2020 TECNOLOGIA, A. *Como baixar e Instalar VirtualBox 6.1 (32Bits e 64Bits) - YouTube*. 2020. <https://www.youtube.com/watch?v=dACGyiBRxVA>. (Accessed on 04/20/2022).

VirtualBox 2022 VIRTUALBOX, O. V. *Chapter 1. First Steps*. 2022. <https://www.virtualbox.org/manual/ch01.html#virtintro>. (Accessed on 04/13/2022).

Vokorokos e Baláž 2010 VOKOROKOS, L.; BALÁŽ, A. Host-based intrusion detection system. In: IEEE. *2010 IEEE 14th International Conference on Intelligent Engineering Systems*. [S.l.], 2010. p. 43–47.

ANEXOS

I. INSTALAÇÃO DO ORACLE VIRTUAL BOX

A instalação do Oracle VirtualBox, para sistemas operacionais Windows é intuitiva e totalmente guiada via um executável, que pode ser baixado pelo site https://www.virtualbox.org/wiki/Downloads clicando na opção Windows hosts. O arquivo executável baixado e instalado nesta máquina foi o **VirtualBox-6.1.32-1492090-Win.exe** mas o site já disponibiliza atualmente a versão **VirtualBox-6.1.34-150636-Win.exe**. Para instalação do Oracle VirtualBox:

1. Execute o arquivo baixado através do link https://www.virtualbox.org/wiki/Downloads com o botão direito em Modo Administrador, clicando posteriormente em **Executar**;

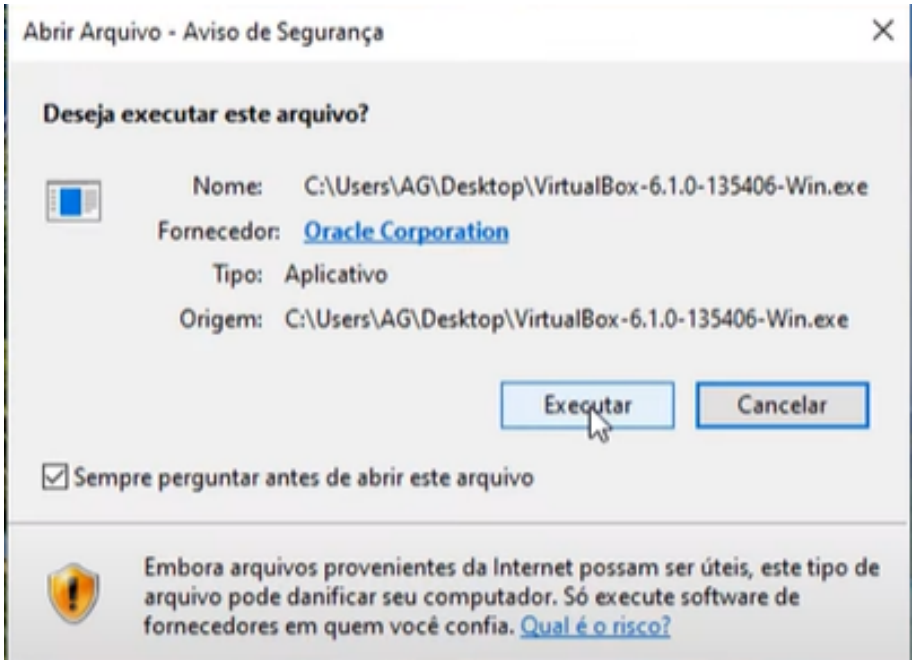

Figura I.1: Instalação Oracle VirtualBox - Passo 1. Fonte: [Tecnologia 2020]

2. Abrirá uma janela com informações relacionadas ao Oracle VirtualBox do que aquele executável irá fazer e solicitando ao usuário que clique na opção **Next**, sendo necessário clicar para prosseguimento da instalação.

Figura I.2: Instalação Oracle VirtualBox - Passo 2. Fonte: [Tecnologia 2020]

3. A próxima janela trará um informativo do quantitativo de armazenamento que o virtualizador necessitará para que seja instalado (218MB) bem como quais *VirtualBox Applications* fazem parte deste pacote de instalação. Por se tratar de item informativo, basta clicar em **Next** para prosseguir com a instalação.

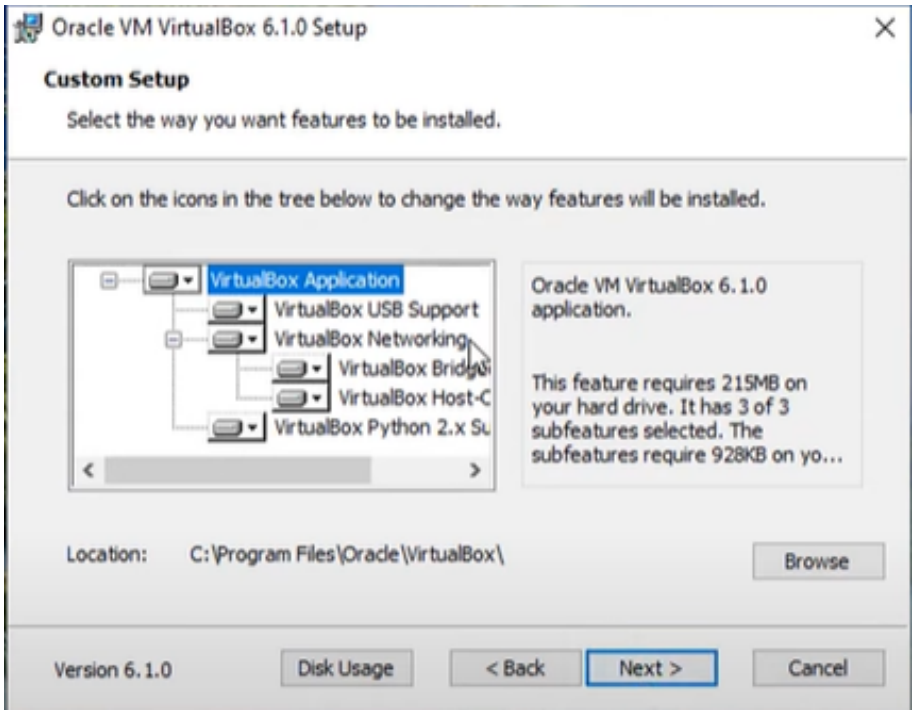

Figura I.3: Instalação Oracle VirtualBox - Passo 3. Fonte: [Tecnologia 2020]

4. A próxima janela informará onde o Oracle VirtualBox criará atálhos e que associará esses atalhos ao registro de arquivos. Os atalhos do Oracle VirtualBox são comumente criados no Menu Iniciar, na Área de Trabalho e na Barra de Execução Rápida, mais conhecida como Barra de Tarefas. Por se tratar de um item informativo, basta clicar em **Next**.

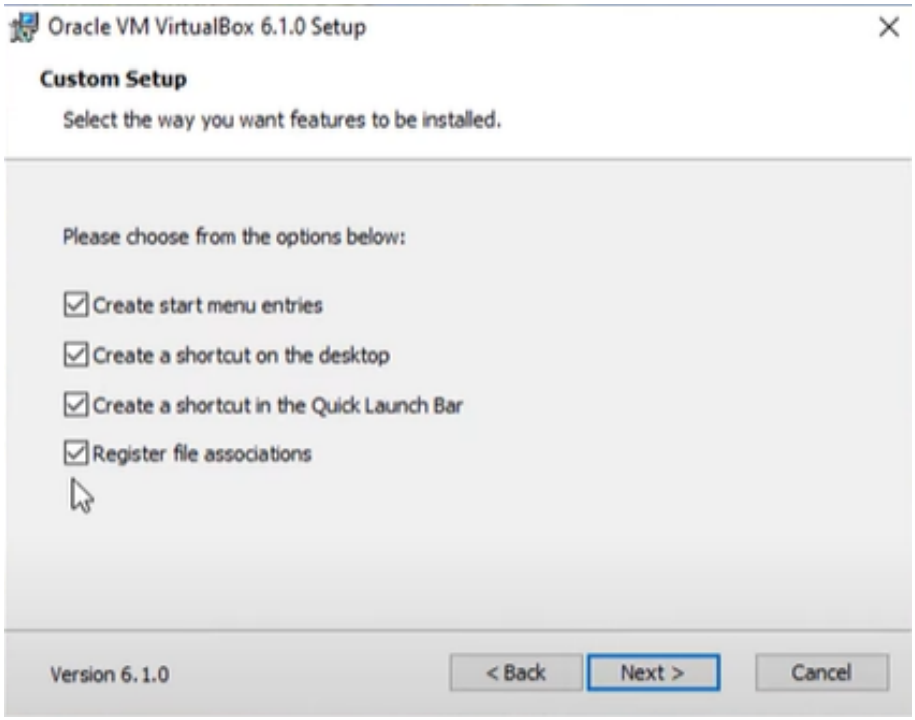

Figura I.4: Instalação Oracle VirtualBox - Passo 4. Fonte: [Tecnologia 2020]

5. A próxima janela é uma janela, apesar de informativa, importante. Ela informa ao usuário que a aplicação irá resetar as interfaces de rede e que por algumas frações de segundos o mesmo ficará desconectado. Isto ocorre por a aplicação precisa criar e habilitar suas interfaces virtuais de rede, que são necessárias para simular a placa de rede das máquinas virtuais que vierem a serem criadas. Como informado, por se tratar de um item informativo, basta clicar em **Next**.

Figura I.5: Instalação Oracle VirtualBox - Passo 5. Fonte: [Tecnologia 2020]

6. A próxima janela solicita ao usuário que seja iniciado a instalação, clicando em **Install**. Após este processo, haverá a inicialização da instalação e poderam, nesse processo, ser solicitado ao usuário a autorização para a criação da interface virtual de rede, como explicado no passo anterior. Desta forma, é necessário que o mesmo acompanhe o processo até que este finalize.

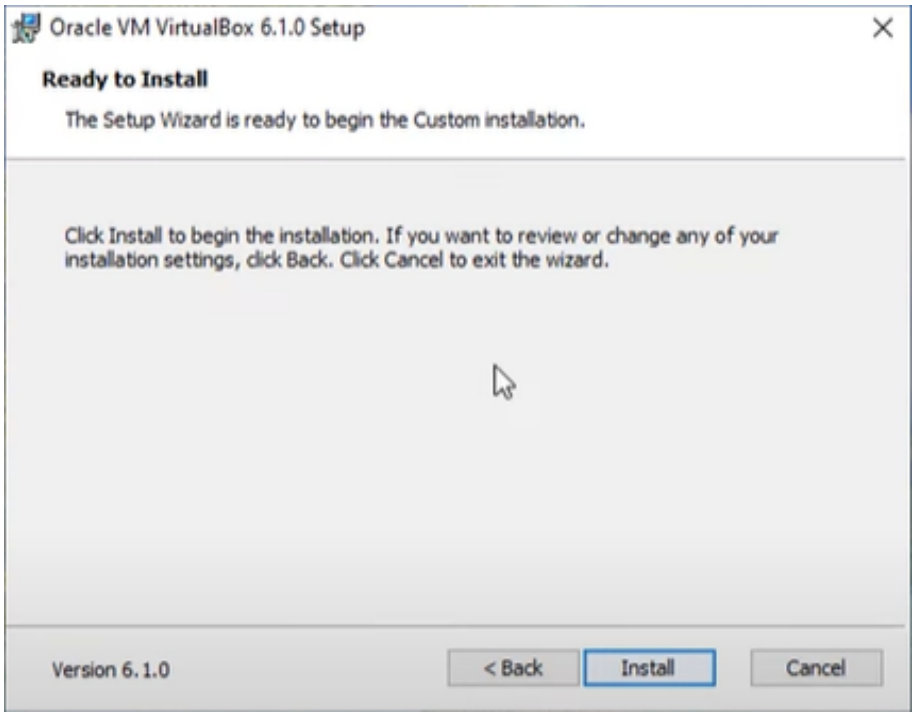

Figura I.6: Instalação Oracle VirtualBox - Passo 6. Fonte: [Tecnologia 2020]

7. Finalizado o processo de instalação, basta clicar em **Finish** que a aplicação abrirá de forma automática. A figura a seguir apresenta a interface inicial do Oracle VirtualBox caso todos os passos anteriores tenham sidos seguidos de forma correta.

Figura I.7: Instalação Oracle VirtualBox - Passo 7. Fonte: [Tecnologia 2020]

A criação de máquinas virtuais e importação à ferramenta de emulação e simulação **GNS-3** serão explorados nas seções 4.2.3 e 4.2.2 respectivamente, haja vista que a instalação do sistema emulador e simulador GNS-3 já pode ser feita com a instalação da máquina virtual GNS-3 VM, otimizando o processo de configuração ferramental.

II. INSTALAÇÃO DO GNS-3

A instalação do Emulador e Simulado GNS-3, assim como o virtualizador Oracle VirtualBox é intuitiva e guiada por um executável, que pode ser baixado pelo site https://www.gns3.com/software/download mediante criação de um cadastro no site da ferramenta, gerenciado pela SolarWinds Worldwide. O arquivo executável baixado nesta máquina foi o **GNS3-2.2.31-all-in-one-regular.exe**, que é a versão atual, até esta data, disponibilizada pelo site.

Para instalação do emulador e simulador GNS-3 (utilizaremos o tutorial disponível em **GNS3 Windows Install** a qual apresenta a versão 2.2.20. Contudo, o processo de instalação é idêntico ao para a versão corrente, a 2.2.31):

1. Execute, em modo administrador, o arquivo baixado, após criação de cadastro, através do link https://www.gns3.com/software/download. A primeira tela trará informações de versão e recomendações iniciais para instalação do sistema. Por se tratar de tela informativa, basta clicar em **Next >**.

Figura II.1: Instalação Graphicall Network Simulator 3 - Passo 1. Fonte: [GNS3]

2. O GNS3 é um software de código aberto gratuito distribuído sob a Licença Pública Geral GNU Versão 3. Leia o contrato de licença e, se concordar com o conteúdo, clique no botão **I Agree** para continuar a instalação:

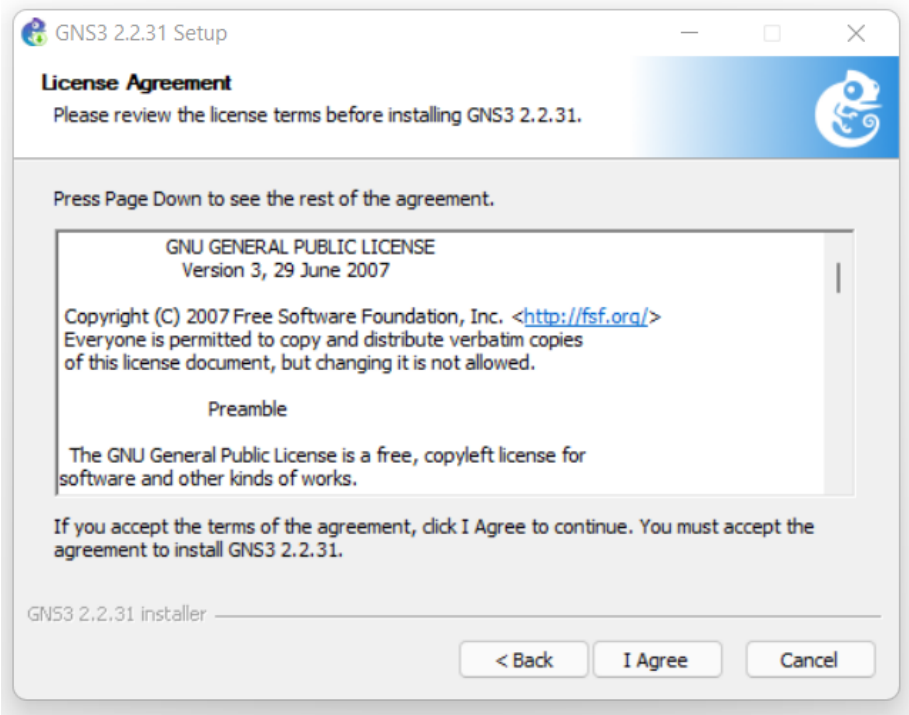

Figura II.2: Instalação Graphicall Network Simulator 3 - Passo 2. Fonte: [GNS3]

3. Selecione a pasta do Menu Iniciar para o atalho GNS3. O padrão é a pasta GNS3. Clique em **Next >** para continuar a instalação:

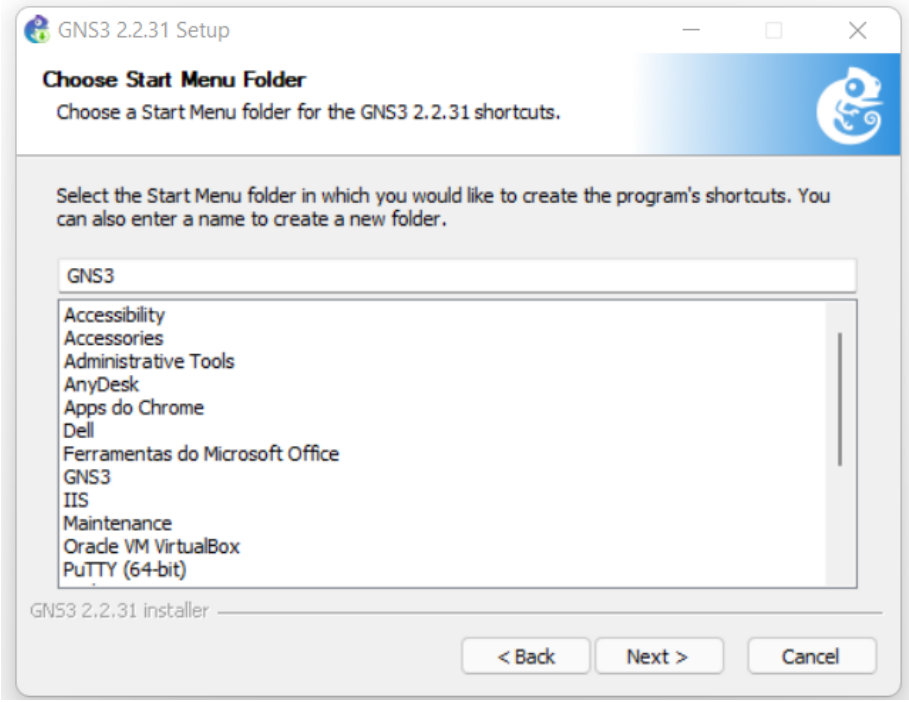

Figura II.3: Instalação Graphical Network Simulator 3 - Passo 3. Fonte: [GNS3]

4. O GNS3 vem com vários pré-requisitos e softwares opcionais. Por padrão, a maioria dos softwares é selecionada para instalação, mas você pode decidir instalar apenas softwares específicos. No caso

desta instalação, serão utilizados e instalados todos os recursos disponíveis pelo GNS-3. A título de esclarecimento, pelo site https://docs.gns3.com/docs/getting-started/installation/windows/ é possível obter um *brief* de cada uma das funcionalidades.

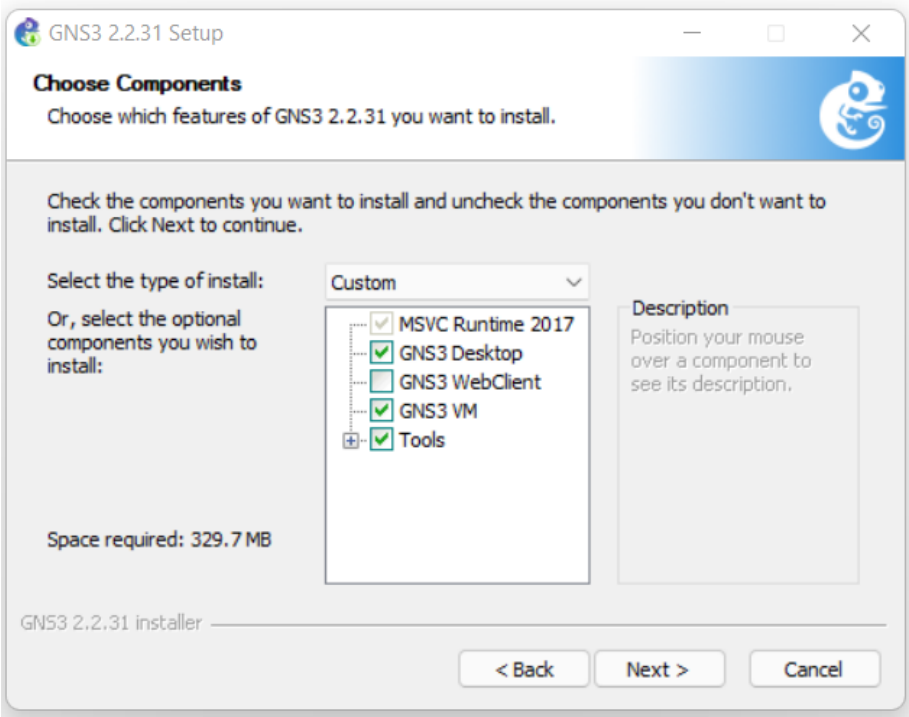

Figura II.4: Instalação Graphical Network Simulator 3 - Passo 4. Fonte: [GNS3]

5. Escolha um local de instalação. O local padrão é **C:\Program Files\GNS3**. Em seguida, clique em **Next >**:

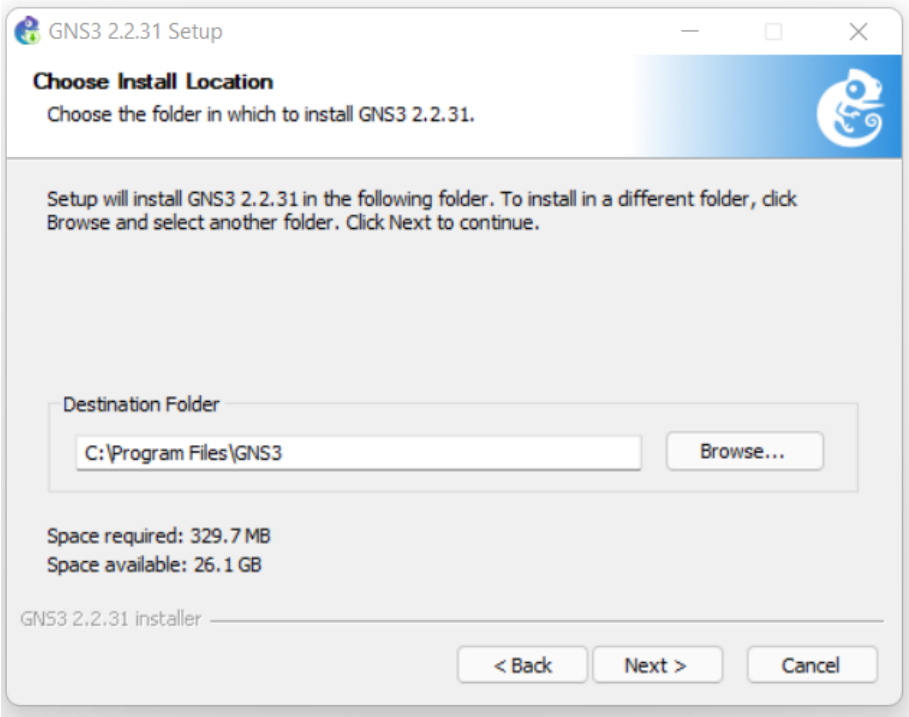

Figura II.5: Instalação Graphical Network Simulator 3 - Passo 5. Fonte: [GNS3]

6. A saída da instalação dependerá da seleção feita pelo usuário no item anterior. As isntruções do que deve ser feito em cada passo são informadas pelo site https://docs.gns3.com/docs/getting-started/installation/windows/. Em muitos casos de instalação limpa, todas as aplicações solicitaram permisão de execução. Neste caso, basta ir avanaçando os passos e aceitando as licenças (que são todas livres) para conclusão da instalação.

7. O sistema, quando selecionado a Opção GNS3-VM, faz automaticamente o downlod da versão compatível com o Virtualizador Selecionado bem como com a versão do Software que está sendo instalado. Para isto, deve-se se selecionar, conforme figura II.6 o virtualizador que será utilizado. No caso deste projeto, selecionou-se VirtualBox. Após a seleção

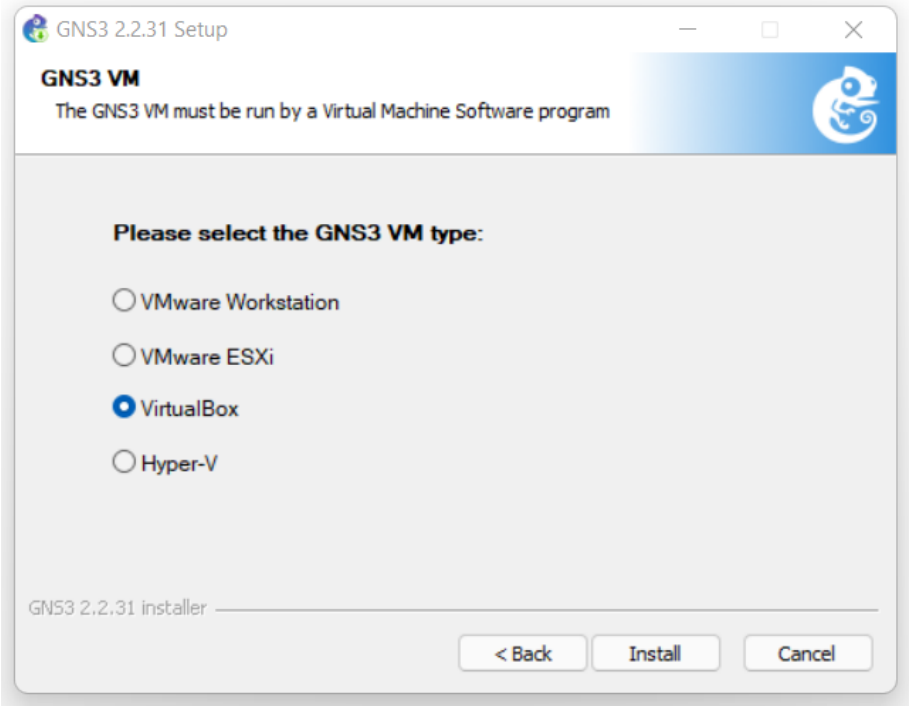

Figura II.6: Instalação Graphical Network Simulator 3 - Passo 6. Fonte: [GNS3]

8. Após o software GNS3 principal (e quaisquer itens opcionais selecionados) estiver instalado, clique em **Next >**

9. **Opcional:** Caso o usuário esteja interessado, ele pode instalar o Solarwinds Standard Toolset. A versão fornecida pelo GNS-3 é uma avaliação gratuita. Caso contrário, basta selecionar **No** e clique em *Next* > para continuar:

10. Finalizado, o executável reportará uma tela de "Instalação Completa"com um *checkbox* selecionado para iniciar o GNS-3. Uma janela do navegador será aberta mostrando ajuda e opções adicionais. Caso deseje iniciar a aplicação, mantenha a seleção **Start GNS3** habilitada e clique em Concluir para concluir a instalação do GNS3.

II.1 GNS-3 VM

A máquina virtual GNS-3 VM pode ser adquirida via download automático no ato da instalação do GNS-3, desde que seja selecionado a opção do virtualizador correto a qual se encontra instalado no host quanto pelo site https://www.gns3.com/software/download-vm. Para instalação do GNS-3 VM:

1. Descompacte o arquivo **GNS3.VM.VirtualBox.2.2.31.zip** e salve o arquivo **GNS3 VM.ova** em um local de fácil acesso.

2. Dê duplo clique no arquivo **GNS3 VM.ova** ou clique com o botão direito e selecione a opção *Open*. Ao realizar este procedimento, o usuário será direcionado ao Oracle VirtualBox após a abertura de

uma janela entitulada **Import Appliance Virtual**. Essa janela trará todas as informações necessárias, como mostrado a seguir:

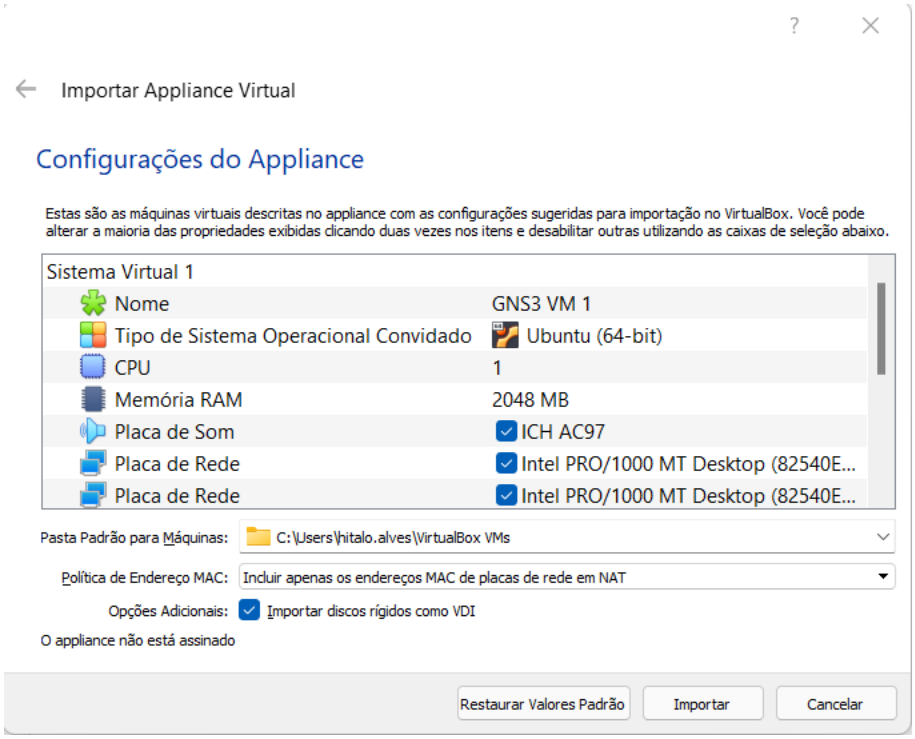

Figura II.7: Instalação Graphical Network Simulator 3 VM - Passo 1 Importação. Fonte:autor

3. Após conferência de todas as configurações disponíveis, basta clicar em **Importar**, para que o processo de importação se inicie. Não é necessário nenhuma ação no processo.

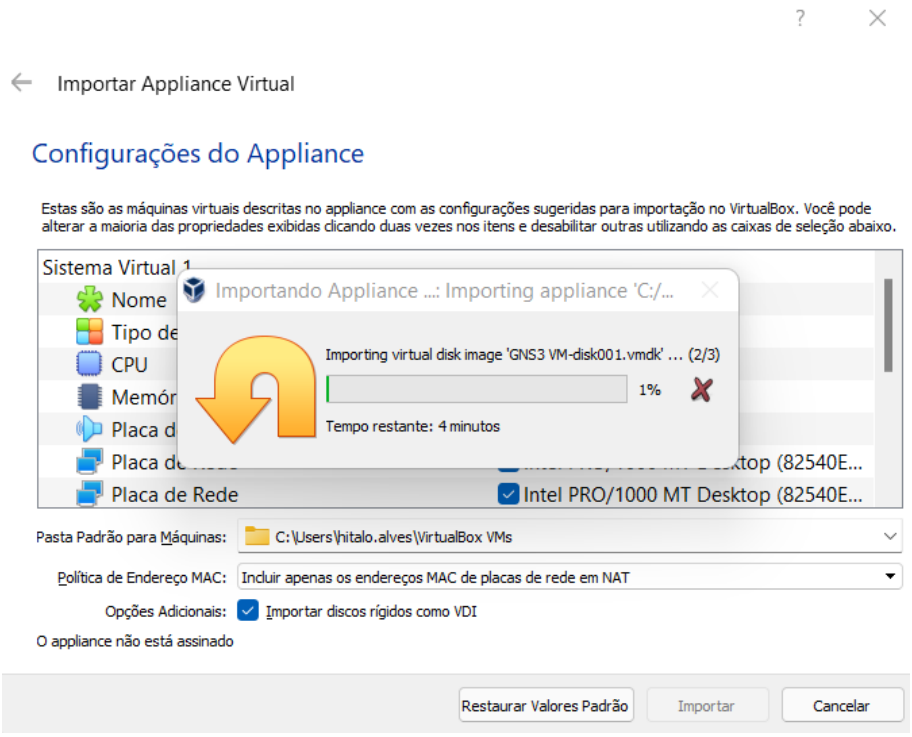

Figura II.8: Instalação Graphical Network Simulator 3 VM - Passo 2 Importação. Fonte:autor

4. Ao final do processo de importação, a máquina virtual deverá estar disponibilizada nas listagem de máquinas virtuais presentes na tela inicial do Oracle VirtualBox, como mostrado a seguir:

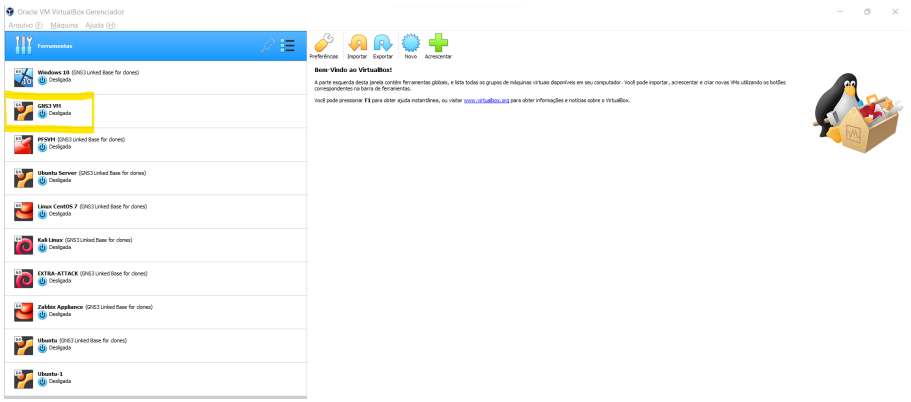

Figura II.9: Instalação Graphical Network Simulator 3 VM - Conclusão da Importação. Fonte:autor

5. **Opicional:** Após a conclusão da importação, é interessante iniciar a máquina virtual para verificar se as configurações foram importadas com sucesso. Não é um passo obrigatório, tendo em vista que a iniciação da mesma ocorrerá quando processo de importação ao GNS-3 GUI for concluída.

Após a importação concluída da máquina virtual GNS-3 VM ao virtualizador Oracle Virtual Box, é necessário que se realize a sincronização do *appliance virtual* com o GNS-3 GUI. Para isso, é necessário:

1. Iniciar ferramenta GNS-3 e posteriormente, após sua iniciliazação, clicar em **Edit** e selecionar a opção **Preferences**, como demonstrado na figura a seguir:

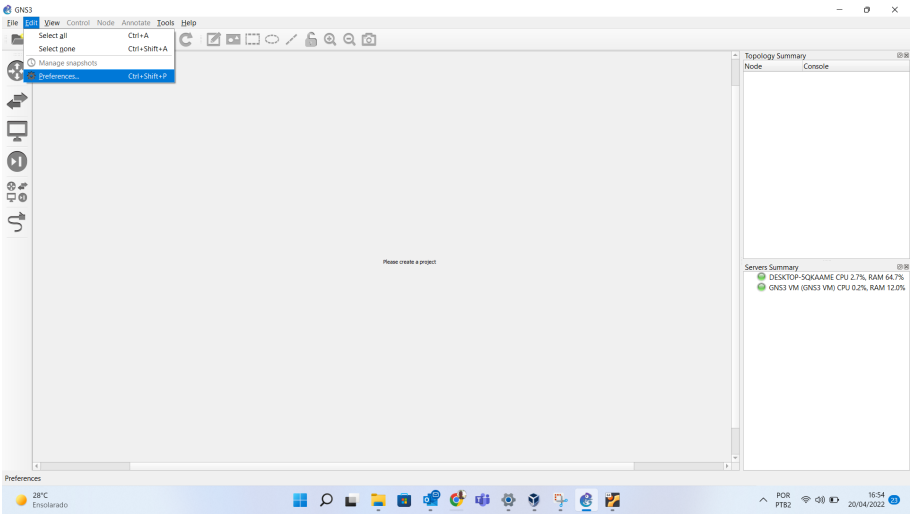

Figura II.10: Instalação Graphical Network Simulator 3 VM - Importação para o GNS-3 GUI: Passo 1. Fonte:autor

2. Selecionado a opção **Preferences**, será direcionado pelo GNS-3 GUI a tela **General Preferences**. Clique na opção **GNS3 VM** para que seja direcionado a tela **GNS3 VM Preferences** e realize as configurações conforme demonstrado a seguir (as configurações de vCPU e RAM devem obedecer os limites de hardware do seu computador, evitando assim uma sobrecarga de trabalho. Por padrão, o GNS-3 indica 1 vCPU e 1024GB de RAM para a GNS-3 VM):

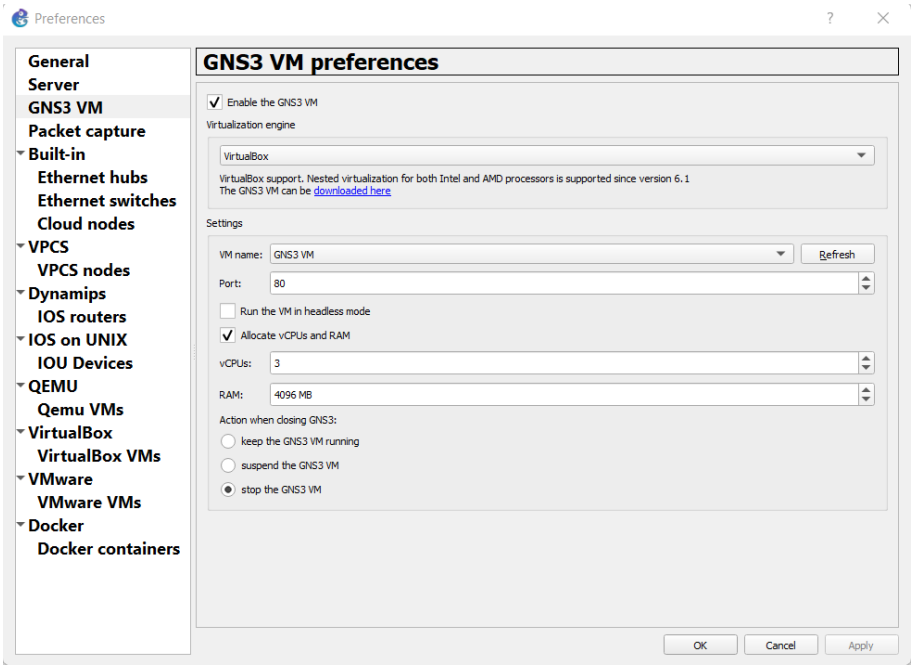

Figura II.11: Instalação Graphical Network Simulator 3 VM - Importação para o GNS-3 GUI: Passo 2. Fonte:autor

3. Finalizado as configurações, basta clicar em **Apply**. O GNS-3 GUI realizará a chamada da GNS-3 VM ao Oracle e o processo pode ser acompanhado pelo campo *Server Sumary*, conforme demonstrado via figura II.12, a qual demonstra a inicialização da GNS-3 VM e a conclusão desta inicialização, quando a mesma se torna operacional e apresenta não mais o indicador cinza, mas sim o

indicador verde.

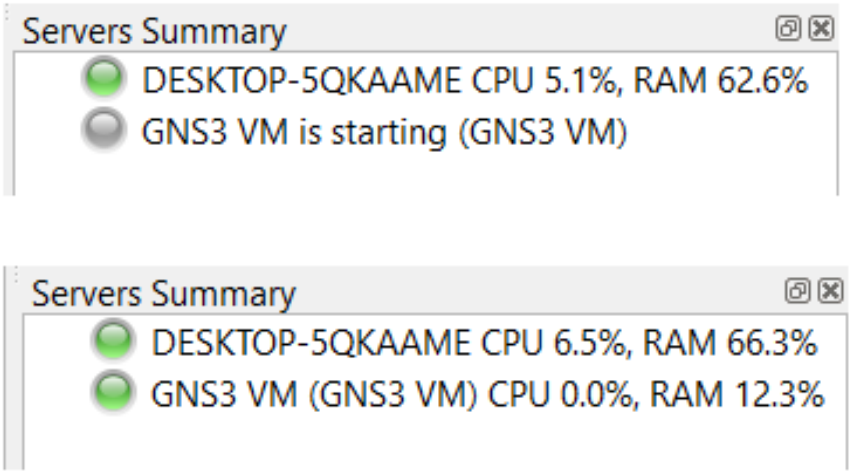

Figura II.12: Instalação Graphical Network Simulator 3 VM - Importação para o GNS-3 GUI: Passo 3. Fonte:autor

4. Outra maneira de conferir se a importação resultou em sucesso é verificar a própria máquina virtua GNS-3 VM. Para que a mesma opere sem erros, o retorno de valores deverá estar em conformidade com a imagem a seguir (vale ressaltar que os endereços IPs podem variar de rede pra rede):

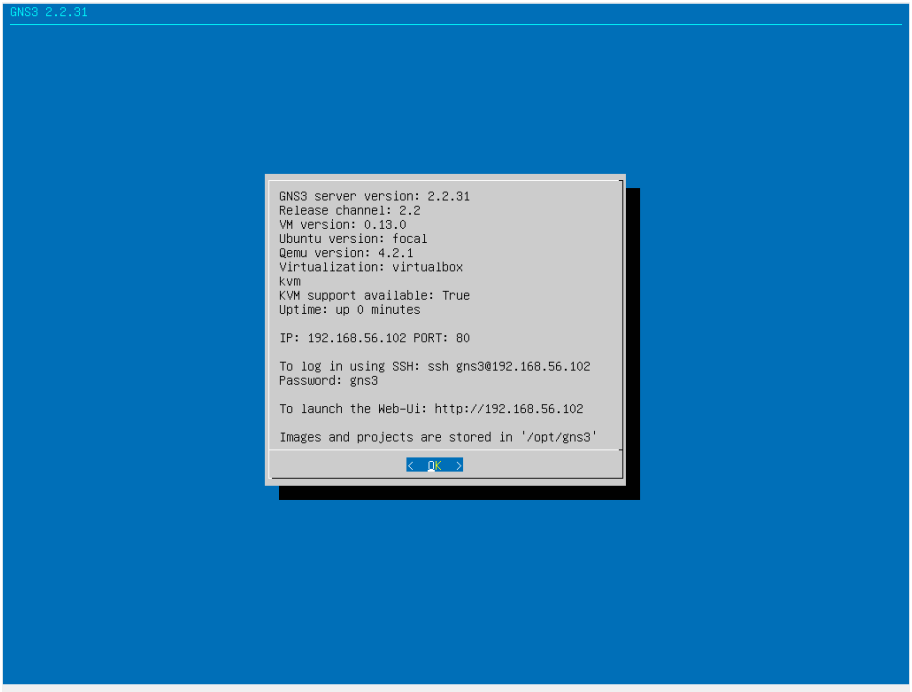

Figura II.13: Instalação Graphical Network Simulator 3 VM - Importação para o GNS-3 GUI: Conclusão. Fonte:autor

Feito isso, todo o processo de importação do GNS-3 se encontra concluído, sendo agora necessário a criação das máquinas virtuais e importação ao GUI do GNS-3, descrito pela seção 4.2.3.

III. IMPORTAÇÃO DE VMS AO GNS-3

Após aplicação dos tutoriais, será possível ter disponível todas as instâncias virtuais necessárias para execução do projeto (salvaguarda o HIDS OSSEC, que deverá ser instalado após todas as configurações de segmentação de rede e definição de topologia, abordadas na seção 4.2.4). Contudo, estas instâncias deverão estar também disponíveis para utilizaçã no GNS-3 GUI, isto é, serem *GNS3 Linked Base for Clones*. Para isto, é necessário:

1. Com o GNS-3 GUI aberto, clicar em **Edit** e depois em **Preferences**, como demonstrado pela figura II.7.

2. Com o **GNS3 General Preferences** aberto, clicar em **VirtualBox** para ser direcionado a tela **VirtualBox Templates**.

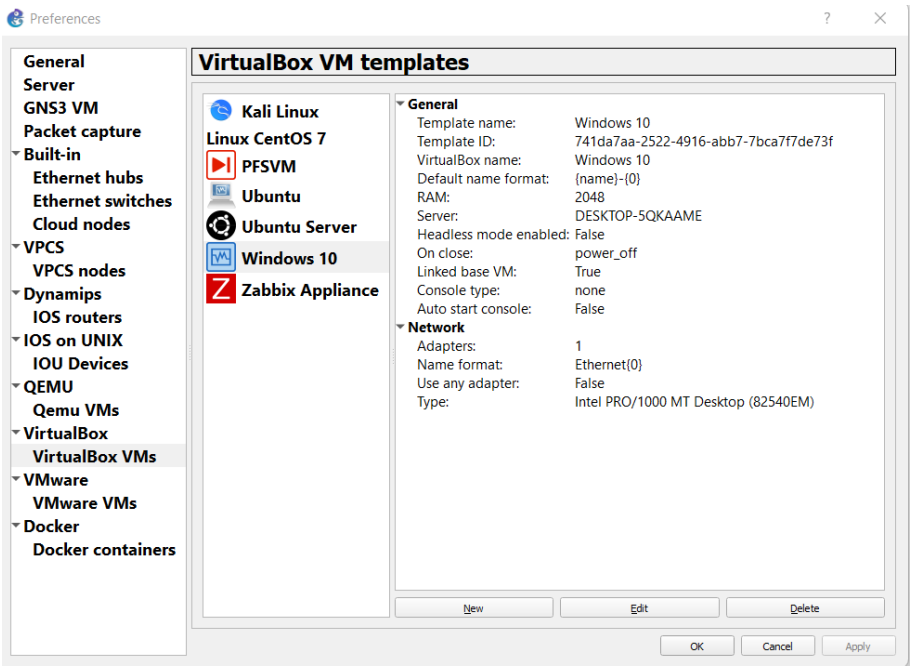

Figura III.1: Importação de VMs do Oracle VirtualBox ao GNS-3 GUI - Passo 1. Fonte: autor

3. Esta área apresenta ao usuário todos os *appliance* que foram importados a partir do VirtualBox. O processo de importação inicia ao usuário clicar na opção **New**.

4. Após iniciar a importação, o GNS-3 questiona em qual tipo de servidor será executada a VM. Como o Oracle Virtual Box se encontra instalado em seu host, deverá ser executado de forma local. Para isto, seleciona a opção **Run this VirtualBox VM on my local Computer**.

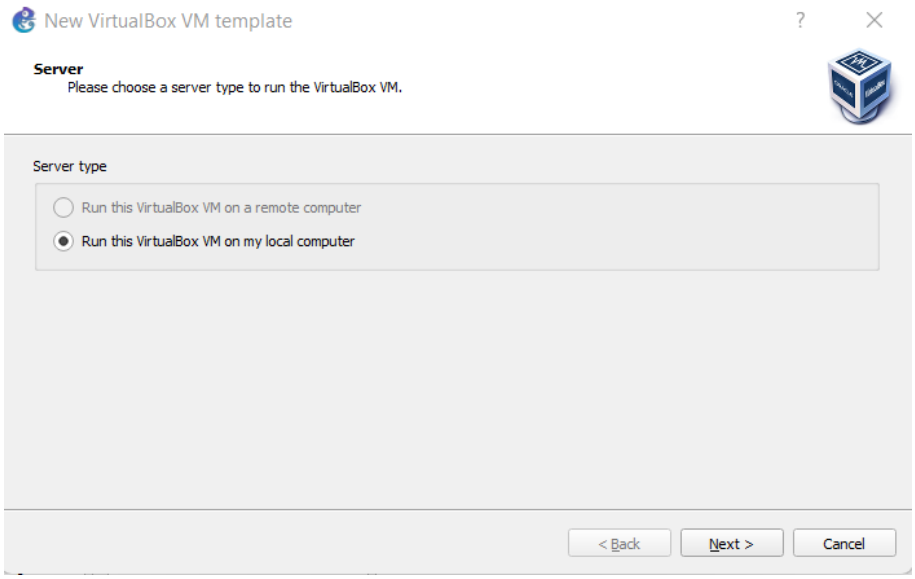

Figura III.2: Importação de VMs do Oracle VirtualBox ao GNS-3 GUI - Passo 2. Fonte: autor

5. O GNS-3 GUI lhe fornecerá a lista de Máquinas Virtuais que estão instaladas no Oracle VirtualBox. Caso os tutoriais registrados na tabela 4.5 tenham sido seguidos, todas as máquinas virtuais estarão disponíveis. Seleciona a que deseja importar e clique em **Finish**.

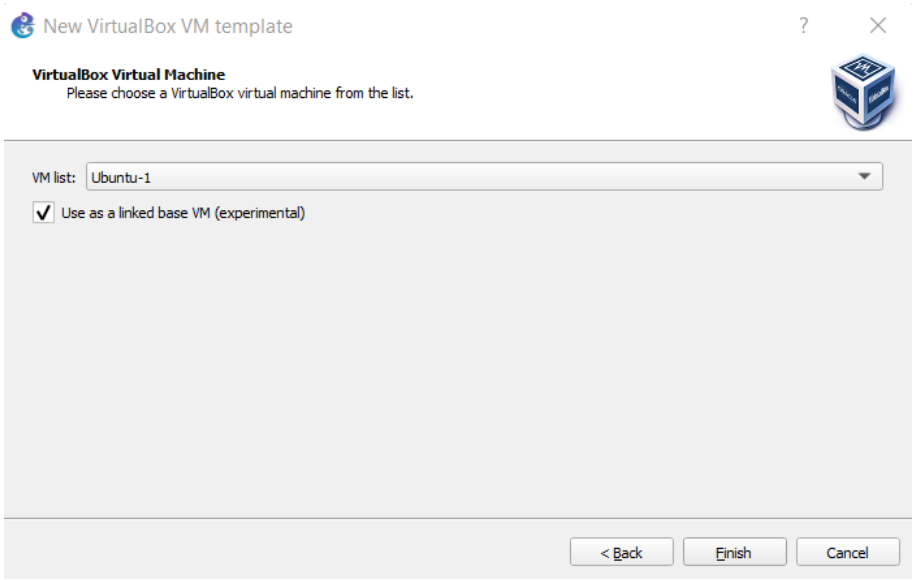

Figura III.3: Importação de VMs do Oracle VirtualBox ao GNS-3 GUI - Passo 3. Fonte: autor

6. Uma vez importado, a máquina virtual ficará já disponível pra uso. Contudo, é importante salientar que em muito casos, é de interesse replicar mais de uma imagem destas máquinas que foram importadas. Desta forma, para que isso seja possível, temos que fazer com que o Appliance importado se torne uma base de Clones. Isto é feito ao selecionar a máquina virtual importada, clicar em **Edit** e posteriormente selecionar a opção **Use as linked base VM (experimental)**. Por fim, clique em **OK** e posteriormente em **Apply** e **OK**.

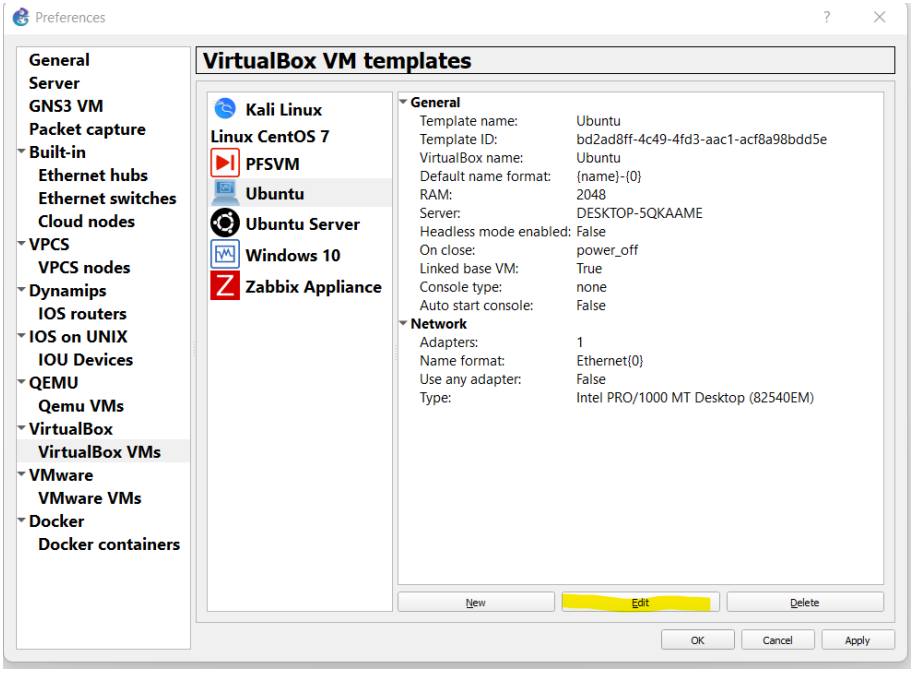

Figura III.4: Importação de VMs do Oracle VirtualBox ao GNS-3 GUI - Passo 4. Fonte: autor

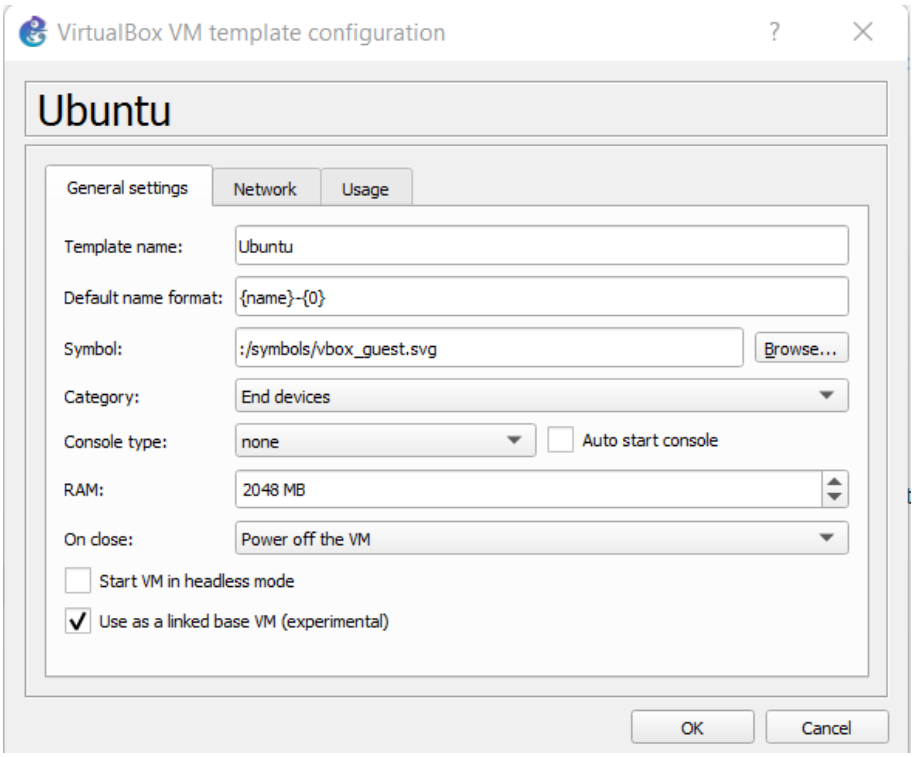

Figura III.5: Importação de VMs do Oracle VirtualBox ao GNS-3 GUI - Passo 5. Fonte: autor

Feito todas as importações necessárias, é necessário se criar um novo cenário para desenho da topologia. Na tela inicial do GNS-3 GUI, clique em **File** e posteriormente em **New Blank Projet**. Atribua um nome e defina o local onde o arquivo será salve e posteriormente, clique em **OK**.

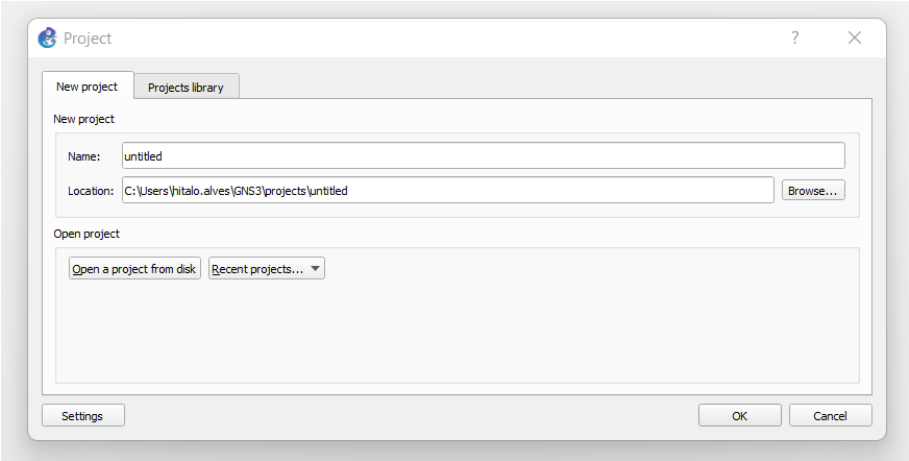

Figura III.6: Criação de uma nova área de trabalho para desenho de cenário no GNS-3 GUI. Fonte: autor

Com uma nova área de trabalho e com todas as instâncias virtuais importadas, é possível agora desenhar toda a topologia apresentada pelo cenário da figura 4.2.

IV. INSTALAÇÃO HIDS OSSEC

IV.1 OSSEC SERVER - CANONICAL UBUNTU 20.04

O processo de instalação do OSSEC SERVER foi guiado pelo tutorial disponível pelo site Step by Step Guide to Install OSSEC HIDS on Ubuntu 20.04 LTS [Club 2021] e será reproduzido e explicado a seguir:

- Para início do processo, é sempre importante verificar se o sistema Ubuntu Server 20.04 se encontra atualizado. Desta forma, executaremos, assim como todo o processo de instalação, em modo root, uma atualização do sistema, por meio do comando a segur:

```
pfg@pfg:~$ sudo su -
[sudo] senha para pfg:*********
root@pfg:~# apt-get update
```

- Uma vez com o sistema instalado, é necessário instalar os pacotes requeridos pelo HIDS OSSEC, como listado a seguir:
  - **unzip:** Necessário para manuseio de arquivos com extensão .zip;
  - **make:** Utilizado para manter grupos de progamas a partir do código-fonte;
  - **gcc:** Ferramenta necessária para compilar arquivos escritos em linguagem C;
  - **build-essential:** Lista de pacotes que são necessários para se criar um pacote DEB (Debian);
  - **php / php-cli / php-common:** Pacotes necessários para operação de um frame em linguagem php, como será o caso da Interface Web OSSEC WUI.
  - **libapache2-mod-php:** Módulo *lib* do Apache v2 para a linguagem PHP.
  - **apache2-utils:** Servidor HTTP Apache, útil para servidores Web (execução da Interface Web OSSEC WUI);
  - **inotify-tools:** Ferramenta de monitoramento de pastas e arquivos, que auxiliará na notificação ao OSSEC sobre modificações de arquivos através da geração de logs.

```
root@pfg:~# apt install wget unzip make gcc build-essential -y
root@pfg:~# apt install php php-cli php-common libapache2-mod-php
root@pfg:~# apt install apache2-utils inotify-tools -y
```

- Após a instalação de toda biblioteca necessária para operação do OSSEC, é necessário realizar o download do arquivo instalador para o Canonical Ubuntu 20.04. Utilizou-se o comando **wget** para fazer o download da versão 3.6.0. A versão mais atual disponibilizada pela OSSEC é a 3.7.0, mas

a mesma ainda apresenta instabilidades na coleta de logs e, desta forma, optou-se pela versão 3.6.0. Por ser um sistema livre, o mesmo possui repositório no github e o download direto pode ser feito via link HIDS OSSEC 3.6.0. Após realizar o download do arquivo, utilizou-se do comando **tar** para fazer a descompactação do arquivo. A extensão *-xvzf* indica a solicitação de extração de arquivos (x), detalhado (v), informando que o arquivo é um arquivo compactado (z) e para utilizar o arquivo tar (*ossec-hids-3.6.0.tar.gz*) para a operação. Assim sendo:

```
root@pfg:~# wget https://github.com/ossec/ossec-hids/archive/
3.6.0.tar.gz
root@pfg:~# tar -xvzf 3.6.0.tar.gz
```

- Uma vez descompactado, é necessário acessar a pasta e executar o arquivo **install.sh**. Esse arquivo retornará um script shell a qual será necessário repassar alguns valores.

```
root@pfg:~# cd ossec-hids-3.6.0/
root@pfg:~# sh install.sh
```

IV.1.1 Valores de instalação

Após a execução do Script **install.sh**, é necessário preencher algumas informações cruciais para a operação do HIDS OSSEC. Estas informações variam de caso a caso. As informações aqui contidas fazem jus a este projeto e como abordado, podem ser ou replicadas para reprodução deste ou embasadas para produção livre. Assim sendo:

- **Definição da Linguagem:** O HIDS fornece em média 15 variações de linguagem de instalação, sendo o padrão a língua inglesa. Por maior comodidade, foi selecionado a língua portuguesa (br).

```
(en/br/cn/de/el/es/fr/hu/it/jp/nl/pl/ru/sr/tr) [en]: br
Script de instalação OSSEC HIDS v3.6.0 - http://www.ossec.net
Você está iniciando o processo de instalação do OSSEC HIDS
Você precisará de um compilador C pré-instalado em seu sistema
Sistema: Linux pfg 5.13.0-37-generic
Usuário: root
Host: pfg

-- Aperte ENTER para continuar ou Ctrl+C para abortar. --
```

- **Definição do tipo de instalação:** O Script em execução fornece suporte para execução de 4 tipos de serviços oferecidos pelo HIDS OSSEC que são: Servidor, Agente, Local e Hibrido. Para este caso, instalaremos o Servidor HIDS OSSEC. Por padrão, o local de instalação do HIDS OSSEC, seja qualquer das opções escolhidas é o **/var/ossec**. Neste caso, basta confirmar pressionando Enter

```
Que tipo de instalação você deseja (servidor, agente, local,
híbrido ou ajuda)? Servidor

Escolha onde instalar o OSSEC HIDS [/var/ossec]: Pressione Enter
```

- **Notificação de Email:** O HIDS OSSEC possibilita o envio de e-mail a cada situação ocorrido em sistema. Caso seja do interesse do usuário, ele pode habilitar essa função tanto nessa etapa de instalação quanto posteriormente. Por padrão, o HIDS OSSEC pede que esta funcionalidade seja habilitada. No caso deste projeto, não será necessário tendo em vista a instalação da OSSEC WUI.

```
Você quer notificação por e-mail? (s/n) [s]: n
```

- **Checagem de integridade:** O checagem de integridade é um parâmetro responsável por conferir a integridade de cada arquivo. Esta funcionalidade cria comparações através de um banco de dados com dígitos verificadores destes arquivos. Uma vez apresentado uma alteração, ele compara estes arquivos e reporta ao sistema que houve uma alteração e atualiza sua tabela com novos verificadores. Como faz sentido, em um sistema de detecção, verificar possíveis alterações de arquivos, esta função será habilitada para este projeto.

```
Você quer executar o cheque de integridade daemon? (s/n) [s]: s
Executando syscheck (daemon de verificação de integridade)...
```

- **Mecanismo de Detecção de Rootkit:** O mecanismo **rootkit** é um mecanismo malware que se trata da permissividade de acesso e control de um dispositivo por parte de um hacker. Este processo permite roubo, dentro de uma vasta lista de dados, de informações valiosas para o banco atacado. Tendo em vista que estes malwares garantem acesso por meio de uma intrusão, esta função será habilitada.

```
Deseja executar o mecanismo de detecção de rootkits? (s/n) [s]: s
Executando rootcheck (detecção de rootkit)...
```

- **Resposta Ativa:** A resposta ativa é um grande atrativo para transformar o HIDS OSSEC em um detector de intrusão ativo, possibilitando o bloqueio de um dispositivo atacante de forma ativa. Como é de interesse deste projeto verificar esta funcionalidade dentro do HIDS OSSEC, a mesma será habilitada.

```
A resposta ativa permite que você execute um comando específico com
base nos eventos recebidos. Por exemplo, você pode bloquear um
endereço IP ou desabilitar o acesso de um usuário específico.

Deseja habilitar a resposta ativa? (s/n) [s]: s
Resposta ativa habilitada...
```

- **Habilitação de SysLog Remoto:** O HIDS OSSEC vem embarcado com o SysLog, protocolo desenvolvido em 1980 documentado pela RFC 3164 em 2001, que usa a porta UDP 514 para envio de alertas e/ou notificações de eventos atrelados a redes IP ou até mesmo a hosts via rede. Como é de interesse que os agentes se comuniquem com o servidor e que ele colete estes logs e notifique via OSSEC WUI, esta funcionalidade será habilitada. Após esta confirmação, o preenchimento de valores foi concluído com sucesso. Basta pressionar **ENTER** para concluir a instalação.

```
Deseja habilitar o syslog remoto (porta 514 udp)? (s/n) [s]: s

-- Pressione ENTER para continuar --
```

Após a instalação do HIDS OSSEC Servidor, é necessário iniciar o serviço. Desta forma, o comando a ser executado para tal é descrito a seguir, com sua respectiva saída:

```
root@pfg:~# /var/ossec/bin/ossec-control start

Started ossec-maild...
Started ossec-execd...
Started ossec-analysisd
Started ossec-logcollector
Started ossec-syscheckd
Started ossec-monitord
```

IV.2 OSSEC AGENT - RHEL CENTOS 8

A instalação do HIDS OSSEC para sistemas RHEL CentOS é semelhante a instalação do HIDS OSSEC Server para os sistemas Canonical Ubuntu, diferenciando em chamadas de comando e necessidade de pacotes. Tal processo foi embasado no tutorial disponível em CentOS 7.x Install OSSEC agent [Notes_Wiki 2019] que, apesar de apresentar um tutorial para uma versão anterior ao utilizado neste projeto, se fez bastante representativo na versão utilizada. Assim sendo:

- Para início do processo, é sempre importante verificar se o sistema CentOS 7 se encontra atualizado. Desta forma, executaremos, assim como todo o processo de instalação, em modo root, uma

atualização do sistema, por meio do comando a segur:

```
[pgraduacao@localhost~]$ sudo su -
[sudo] senha para pgraduacao:*********
[root@localhost~]# yum update
```

- Uma vez com o sistema instalado, é necessário instalar os pacotes requeridos pelo HIDS OSSEC, como listado a seguir:
  - **make:** Utilizado para manter grupos de progamas a partir do código-fonte;
  - **gcc:** Ferramenta necessária para compilar arquivos escritos em linguagem C;
  - **openssl-devel:** Instala o OpenSSL, implementação d SSL e TLS para código aberto

```
[root@localhost~]# yum install make gcc openssl-devel -y
```

- Após a instalação de toda biblioteca necessária para operação do OSSEC AGENT, é necessário realizar o download do arquivo instalador para o RHEL CentOS 7. Utilizou-se o comando **wget** para fazer o download da versão 3.7.0, versão mais atual disponibilizada pela OSSEC. Por ser um sistema livre, o mesmo possui repositório no github e o download direto pode ser feito via link HIDS OSSEC 3.7.0. Após realizar o download do arquivo, utilizou-se do comando **tar** para fazer a descompactação do arquivo. A extensão *-xvzf* indica a solicitação de extração de arquivos (x), detalhado (v), informando que o arquivo é um arquivo compactado (z) e para utilizar o arquivo tar (*ossec-hids-3.7.0.tar.gz*) para a operação. Assim sendo:

```
[root@localhost~]# wget https://github.com/ossec/ossec-hids/
archive/3.7.0.tar.gz
[root@localhost~]# tar -xvzf 3.7.0.tar.gz
```

- Uma vez descompactado, é necessário acessar a pasta e executar o arquivo **install.sh**. Esse arquivo retornará um script shell a qual será necessário repassar alguns valores.

```
[root@localhost~]# cd ossec-hids-3.7.0/
[root@localhost~]# sh install.sh
```

IV.2.1 Valores de instalação

Após a execução do Script **install.sh**, é necessário preencher algumas informações cruciais para a operação do HIDS OSSEC. Estas informações variam de caso a caso. As informações aqui contidas fazem jus a este projeto e como abordado, podem ser ou replicadas para reprodução deste ou embasadas para produção livre. Assim sendo:

- **Definição da Linguagem:** O HIDS fornece em média 15 variações de linguagem de instalação, sendo o padrão a língua inglesa. Por maior comodidade, foi selecionado a língua portuguesa (br).

```
(en/br/cn/de/el/es/fr/hu/it/jp/nl/pl/ru/sr/tr) [en]: br
Script de instalação OSSEC HIDS v3.7.0 - http://www.ossec.net
Você está iniciando o processo de instalação do OSSEC HIDS
Você precisará de um compilador C pré-instalado em seu sistema
Sistema: Linux pfg 5.13.0-37-generic
Usuário: root
Host: pfg

-- Aperte ENTER para continuar ou Ctrl+C para abortar. --
```

- **Definição do tipo de instalação:** O Script em execução fornece suporte para execução de 4 tipos de serviços oferecidos pelo HIDS OSSEC que são: Servidor, Agente, Local e Hibrido. Para este caso, instalaremos o Agente HIDS OSSEC. Por padrão, o local de instalação do HIDS OSSEC, seja qualquer das opções escolhidas é o **/var/ossec**. Neste caso, basta confirmar pressionando Enter

```
Que tipo de instalação você deseja (servidor, agente, local,
híbrido ou ajuda)? Agente

Escolha onde instalar o OSSEC HIDS [/var/ossec]: Pressione Enter
```

- Após selecionar a opção de Agente, é necessário fornecer o IP do servidor a qual o agente se comunicará. No caso deste projeto, o IP do servidor é dado por: **172.16.75.253**. As configurações subsequentes faram jus as questões de **Checagem de integridade**, **Mecanismo de Detecção de Rootkit** e **Resposta Ativa**. Como já apresentado pela seção IV.1.1, essas opçoes devem ser configuradas, para as inteções deste projeto, como SIM.

```
Qual é o endereço IP ou nome de host do servidor OSSEC HIDS?
172.16.75.253

Você quer executar o cheque de integridade daemon? (s/n) [s]: s
Executando syscheck (daemon de verificação de integridade)...

Deseja executar o mecanismo de detecção de rootkits? (s/n) [s]: s
Executando rootcheck (detecção de rootkit)...

A resposta ativa permite que você execute um comando específico com
base nos eventos recebidos. Por exemplo, você pode bloquear um
endereço IP ou desabilitar o acesso de um usuário específico.
Deseja habilitar a resposta ativa? (s/n) [s]: s
Resposta ativa habilitada...

-- Pressione ENTER para continuar --
```

IV.3 OSSEC AGENT - MICROSOFT WINDOWS 10 PRO

A instalação do HIDS OSSEC Agente no Microsoft Windows passa por executável a qual pode ser encontrado no site https://updates.atomicorp.com/channels/atomic/windows/ossec-agent-win32-3.7.0-24343.exe. A versão mais atual disponibilizada pelo site faz jus a versão 3.7.0. O HIDS OSSEC é multiversão, isto é, faz integração de seu agente com seu servidor independente da versão habilitada em cada um. Contudo, recomenda-se utilizar a mesma versão em seu sistema. Neste caso, foi utilizado a versão 3.7.0 para todos os dispositivos. Para instalação do HIDS OSSEC Agent no Microsoft Windows:

- Execute o arquivo **ossec-agent-win32-3.7.0-24343.exe** em modo administrador. Irá abrir uma janela de boas vindas e informando o que será instalado. Por se tratar de uma janela informativa, basta clicar em **Next >**;

Figura IV.1: Instalação do Agente HIDS OSSEC no Microsoft Windows 10 Pro - Passo 1. Fonte: autor

- A próxima janela apresentará os Termos de Acordo para instalação do sistema. Para prosseguimento é necessário aceitar os termos, logo basta clicar em **I Agree**;

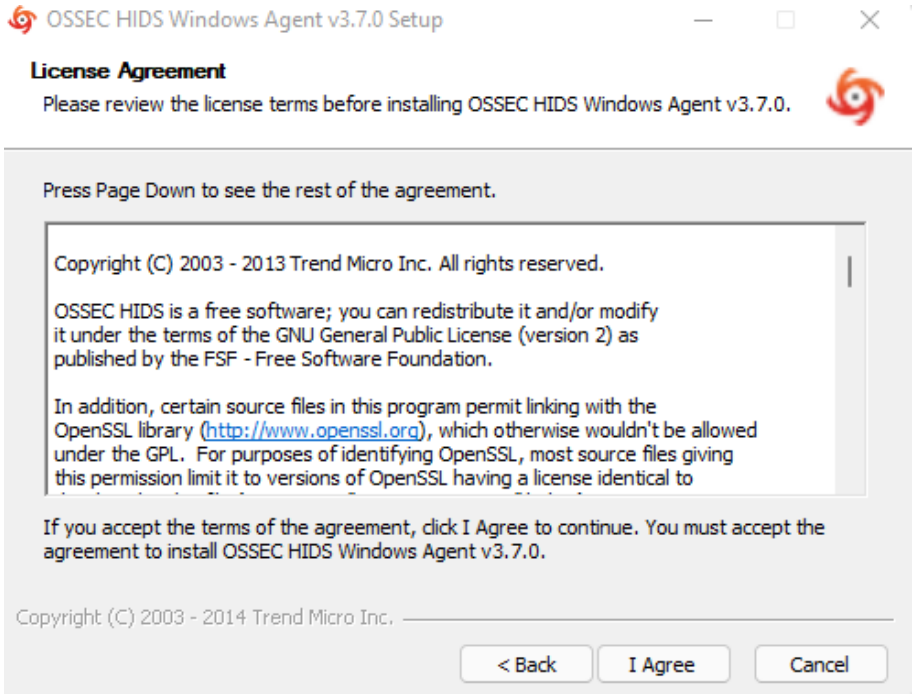

Figura IV.2: Instalação do Agente HIDS OSSEC no Microsoft Windows 10 Pro - Passo 2. Fonte: autor

- A próxima janela informará quais componentes é desejável que seja instalado. Como é de interesse deste projeto que o serviço de scan e monitoramento de logs e a checagem de integridade estejam

ativas, mantem-se o *checkbox* selecionado.

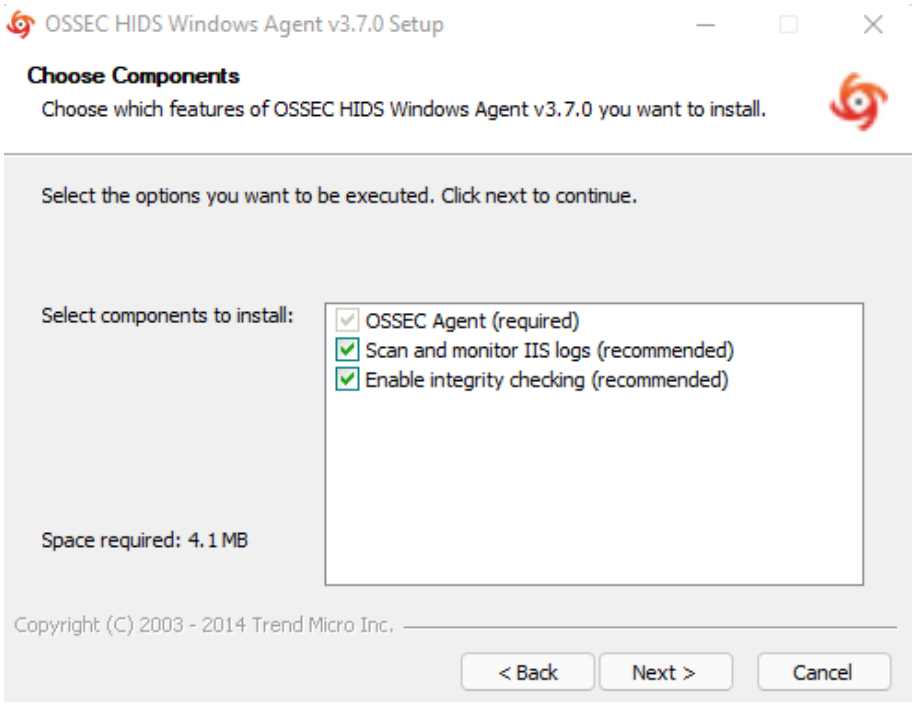

Figura IV.3: Instalação do Agente HIDS OSSEC no Microsoft Windows 10 Pro - Passo 3. Fonte: autor

- A próxima janela apresentará o local onde o HIDS OSSEC Agente será instalado. Por padrão, o diretório a ser instalado é o *C:\Program Files\(x86)ossec-agent* e o tamanho padrão do arquivo é de 4.1MB. Finalizado as verificações, basta clicar em **Install**.

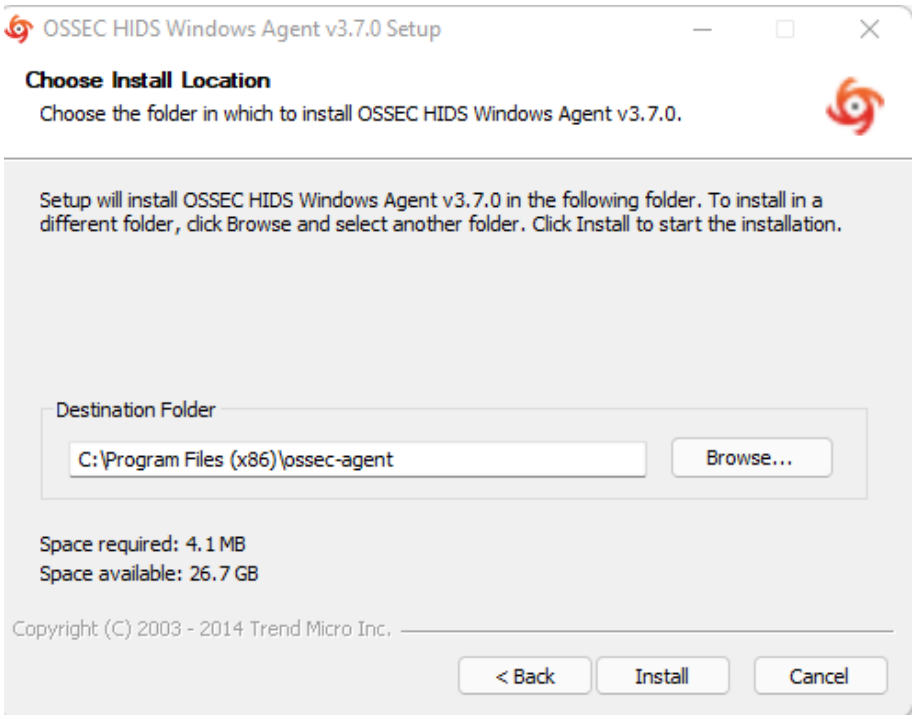

Figura IV.4: Instalação do Agente HIDS OSSEC no Microsoft Windows 10 Pro - Passo 4. Fonte: autor

- Após a conclusão basta clicar em **Next >** e por fim em **Finish**. Caso não queira executar imediatamente o OSSEC Agent Manager, basta desmarcar o *checkbox***Run OSSEC Agent Manager**.

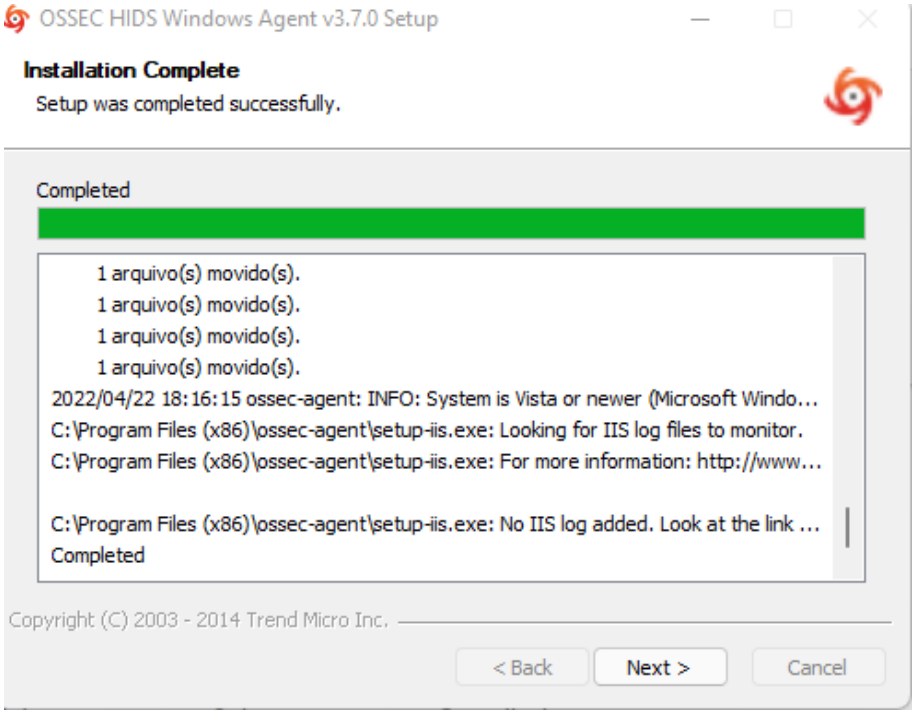

Figura IV.5: Instalação do Agente HIDS OSSEC no Microsoft Windows 10 Pro - Passo 5. Fonte: autor

- É necessário realizar a configuração básica do OSSEC Agent no Windows via OSSEC Agent Manager. Para isso, é necessário inserir o endereço IP do Servidor OSSEC e também a chave de autenticação obtida atravé do registro do agente no servidor. Para realizar o registro de agentes no servidor OSSEC, basta verificar a documentação *Windows Agent Installation*. **Apesar da documentação guiar o passo-a-passo para o Agente Windows, o processo de registro até o *Step 3* é idêntico para qualquer que seja o Sistema Operacional e deve ser executado sempre no Servidor HIDS OSSEC.**

Figura IV.6: Instalação do Agente HIDS OSSEC no Microsoft Windows 10 Pro - Passo 6. Fonte: autor