

Trabalho Final de Graduação

Avaliação e teste de ataques cibernéticos via ferramenta de
EDR

Ana Clara Custódio Gosenheimer e Felipe Lopes Gurgel Nogueira

Brasília, Setembro de 2022

UNIVERSIDADE DE BRASÍLIA

FACULDADE DE TECNOLOGIA

UNIVERSIDADE DE BRASÍLIA
Faculdade de Tecnologia

Trabalho Final de Graduação

Avaliação e teste de ataques cibernéticos via ferramenta de
EDR

Ana Clara Custódio Gosenheimer e Felipe Lopes Gurgel Nogueira

Relatório submetido ao Departamento de Engenharia
Elétrica como requisito parcial para obtenção
do grau de Engenheiro de Redes de Comunicação

Banca Examinadora

Prof. Dr. Georges Daniel Amvame Nze, _____
ENE/UnB
Orientador

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

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

Agradecimentos

Ana Clara Custódio Gosenheimer

Em primeiro lugar, e de forma um tanto clichê, gostaria de agradecer a minha família. Muito obrigada aos meus pais Jaqueline e Remi, por sempre terem me apoiado no caminho da engenharia, mesmo nas épocas mais difíceis da longa caminhada. Agradeço, do fundo do meu coração, ao meu irmão mais velho, Arthur, por sempre ter me acompanhado, incentivado e cuidado de mim. Eu não estaria onde estou hoje sem a sua ajuda. Também agradeço a minha irmã, Amanda, pelas diversas conversas, risos e caminhos compartilhados. E agradeço aos meus irmãos mais novos, Gabriel, por todas as piadas e risadas, e Felipe, por todo o carinho.

Gostaria de agradecer a todos os colegas, amigos e professores que conheci durante esta caminhada, cada um de vocês marcou a minha história e sempre estará lá. De forma especial, gostaria de agradecer ao Paulo Thiago, a Larissa Pires, a Giovanna Castro e a Mirella Honório. Agradeço também aos amigos do ensino médio, em especial a Manuella Sousa, Luiza Fernandes e Júlia Siqueira, que sempre estiveram e sempre estarão me acompanhando. No quesito amizade, um muito obrigado aos meus amigos mais recentes conhecidos no trabalho, obrigada por todas as energias positivas, saídas e risadas.

De forma muito especial gostaria de agradecer por todo o suporte, carinho e atenção dada pelo meu namorado João, muito obrigada por tudo.

Agradeço, também, ao professor Georges Daniel Amvame Nze por todas as matérias ministradas e por nos acompanhar nesta jornada.

E por fim, mas não menos importante, muito obrigada Felipe Nogueira, por todos estes anos de engenharia, por todos os trabalhos feitos juntos e, principalmente, por ser minha dupla neste projeto final.

Agradecimentos

Felipe Lopes Gurgel Nogueira

Agradeço primeiramente aos meus pais, Lúcia e Renato, por todo apoio, pelo amor e pela educação. Agradeço também ao meu irmão, Fernando, pela inspiração, pelos ensinamentos e pela fraternidade.

Agradeço ao meu orientador, Georges Daniel Amvame Nze, pela paciência, apoio e orientação. Aos meus colegas de curso, pelo companheirismo, e aos meus professores, pelos desafios e ensinamentos.

Por fim, agradeço à minha dupla, Ana Clara, pela paciência, dedicação e amizade.

Resumo

Em uma realidade tão digital e globalizada, é de conhecimento geral que a área de Tecnologia da Informação tem recebido cada vez mais destaque, facilitando e melhorando diversos aspectos do dia-a-dia de cada um. Entretanto, no ano de 2020, percebeu-se que a tecnologia não representa apenas uma facilidade. Ela é, neste momento, a base do modelo de vida do ser humano do século XXI. Durante a pandemia, que se iniciou em 2020, o mundo pairava-se sobre incertezas e medos. Somente por meio da tecnologia foi possível ter contato com familiares, amigos e colegas. Esta tornou viável a execução de grande parte das tarefas desempenhadas em empresas de forma remota, mantendo-se a segurança e saúde destes trabalhadores. Com uma presença cada vez mais abrangente, o uso da tecnologia traz consigo não somente benefícios, mas também preocupações. Tendo-se a descoberta de novas ameaças, ataques e vulnerabilidades todos os dias, faz-se de extrema importância um cuidado crescente com a segurança de dados, informações, sistemas e equipamentos. Tornou-se essencial garantir a confiabilidade, integridade e disponibilidade das informações, zelando pelo controle e conformidade de dados e dispositivos. Para se proteger de tantas ameaças, várias estratégias, técnicas e ferramentas são utilizadas, buscando-se sempre manter a segurança em toda a sua integridade. Dentre tantas frentes, destaca-se a atenção necessária com dispositivos finais. Os dispositivos finais podem representar grandes riscos na segurança da informação, uma vez que estes são utilizados em elevada quantidade por usuários comuns, que, sem o devido cuidado e conhecimento, podem facilitar a entrada e proliferação de malwares, ameaças e ataques. Portanto, torna-se imprescindível garantir a segurança dos dispositivos finais, que pode ser obtida por meio de ferramentas especializadas para o caso. Uma das ferramentas com esta proposta é o EDR, Endpoint Detection and Response, que possui a finalidade de monitorar, detectar e realizar as medidas necessárias para responder a uma possível ameaça identificada no dispositivo final.

Palavras-chaves: Segurança, integridade, confiabilidade, disponibilidade, dispositivos finais, EDR, OSSEC, Wazuh, OpenEDR.

Abstract

In such a digital and globalized reality, it is common knowledge that the Information has received more and more prominence, facilitating and improving several aspects of the each one's day-to-day. However, in 2020, it was realized that technology does not represent just a facility, it is, at this moment, the basis of the model of life of the human being of the 21st century. During the pandemic, which began in 2020, the world was hovering over uncertainties and fears. Only through technology was it possible to contact family, friends and colleagues. The internet enabled the execution of most of the tasks performed in companies remotely, keeping workers safe and healthy. With an increasingly comprehensive presence, the use of technology brings with it not only benefits, but also concerns. With the discovery of new threats, attacks and vulnerabilities every day, it is extremely important to be careful with data security. It has become essential to ensure confidentiality, integrity and availability of information, ensuring the control and compliance of data and devices. To be safe from so many threats, various strategies, techniques and tools are used, always seeking to maintain security in all its integrity. Among so many fronts, it is necessary to pay attention to endpoints. Endpoints can represent major risks in information security, since they are widely used by common users, who, without due care and knowledge, can facilitate the entry and proliferation of malware, threats and attacks. Therefore, it is essential to guarantee the safety of endpoints, which can be obtained through specialized tools for the case. One of the tools with this proposal is EDR, Endpoint Detection and Response, which has the purpose of monitoring, detecting and taking the necessary measures to respond to a potential threat identified on the device.

Keywords: Security, integrity, confidentiality, availability, endpoint, EDR, OSSEC, Wazuh, OpenEDR.

SUMÁRIO

SUMÁRIO	7
LISTA DE FIGURAS	10
1 INTRODUÇÃO	1
1.1 MOTIVAÇÃO	2
1.2 OBJETIVOS	2
1.2.1 OBJETIVO GERAL	2
1.2.2 OBJETIVOS ESPECÍFICOS	2
2 REFERENCIAL TEÓRICO	3
2.1 SEGURANÇA DA INFORMAÇÃO	3
2.1.0.1 CONFIDENCIALIDADE	3
2.1.0.2 INTEGRIDADE	4
2.1.0.3 DISPONIBILIDADE	4
2.2 LOGS	4
2.3 SECURITY INFORMATION AND EVENT MANAGEMENT	5
2.4 AMEAÇAS, VULNERABILIDADES E RISCOS	6
2.5 ADVANCED PERSISTENT THREAT	7
2.6 ENDPOINT DETECTION AND RESPONSE	8
3 FERRAMENTAS UTILIZADAS	10
3.1 GNS3	10
3.2 EDRs	10
3.2.1 OSSEC+	11
3.2.2 WAZUH	11
3.2.3 OPENEDR	12
3.3 PILHA ELK	12
3.4 PFSense	13
3.5 VYOS	14
3.6 CYBER KILL CHAIN	14
4 ARQUITETURA PROPOSTA	17
4.1 METODOLOGIA	17
4.2 TOPOLOGIA	18
4.3 CONFIGURAÇÕES	20
4.3.1 PFSense	20
4.3.2 ROTEADOR VYOS	22

4.3.3	SWITCHES	24
4.3.4	PILHA ELK	24
4.3.5	OSSEC+	25
4.3.6	WAZUH	26
4.3.7	OPENEDR	27
5	TESTE E ANÁLISE	29
5.1	ATAQUES	31
5.1.1	SCAN NMAP	31
5.1.1.1	RESULTADO OSSEC+	32
5.1.1.2	RESULTADO WAZUH	33
5.1.1.3	RESULTADO OPENEDR	35
5.1.2	SSH BRUTE-FORCE	36
5.1.2.1	RESULTADO OSSEC+	36
5.1.2.2	RESULTADO WAZUH	37
5.1.3	RDP BRUTE-FORCE	38
5.1.3.1	RESULTADO OPENEDR	39
5.1.4	KERNEL-MODE ROOTKIT	41
5.1.4.1	RESULTADO OSSEC+	42
5.1.4.2	RESULTADO WAZUH	42
5.1.5	ATAQUE SHELLSHOCK	42
5.1.5.1	RESULTADO OSSEC+	43
5.1.5.2	RESULTADO WAZUH	44
6	ANÁLISE E COMPARAÇÃO DO DESEMPENHO DOS EDRS	47
6.1	SCAN NMAP	47
6.2	SSH BRUTE-FORCE	47
6.3	RDP BRUTE-FORCE	47
6.4	KERNEL-MODE ROOTKIT	48
6.5	ATAQUE SHELLSHOCK	48
6.6	COMPARATIVO FINAL	49
7	CONCLUSÃO	51
7.1	TRABALHOS FUTUROS	51
	Bibliografia	52
	ANEXO A – INSTALAÇÃO E CONFIGURAÇÃO DO SERVIDOR OSSEC+ NO UBUNTU 16.04	57
	ANEXO B – INSTALAÇÃO E CONFIGURAÇÃO DO AGENTE OSSEC NO UBUNTU 16.04	59
	ANEXO C – INSTALAÇÃO DO AGENTE OPENEDR NO WINDOWS 10	60

ANEXO D – INSTALAÇÃO DO SERVIDOR WAZUH NO CENTOS7 . . .	61
ANEXO E – INSTALAÇÃO DO AGENTE WAZUH NO CENTOS7	64
ANEXO F – INSTALAÇÃO DA PILHA ELK NO CENTOS7	66

LISTA DE FIGURAS

Figura 2.1 – Tríade CID [40]	3
Figura 2.2 – SIEM [57]	5
Figura 2.3 – Tipos de hackers [26]	7
Figura 3.1 – Pilha ELK [44]	13
Figura 3.2 – Fases da cyber kill chain	15
Figura 4.1 – Versão GNS3	18
Figura 4.2 – Proposta de topologia	19
Figura 4.3 – Topologia da rede vista no GNS3	20
Figura 4.4 – Endereçamento IP das interfaces do Firewall	21
Figura 4.5 – Regras da interface WAN do Firewall	21
Figura 4.6 – Regras da interface LAN do Firewall	22
Figura 4.7 – Regras da interface OPT1 do Firewall	22
Figura 4.8 – Configuração do Kibana	25
Figura 4.9 – Configuração para conectar ao Wazuh-API	25
Figura 4.10–Agente OSSEC disponível na interface de configuração do OSSEC+	25
Figura 4.11–Chaves de pareamento entre o servidor e os agentes	26
Figura 4.12–Configuração OSSEC - Arquivos de log	26
Figura 4.13–Configuração OSSEC - Conexão ELK	26
Figura 4.14–Configuração do Filebeat no Wazuh Server	27
Figura 4.15–Comunicação estabelecida com sucesso entre os nós Wazuh Agent e Server	27
Figura 4.16–Configuração do Filebeat - Comunicação ELK	28
Figura 4.17–Configuração do Filebeat - Arquivos de Log	28
Figura 5.1 – Regra da interface OPT1 para a permitir a realização dos testes	29
Figura 5.2 – Configuração de rede da máquina Kali Linux	31
Figura 5.3 – Comando nmap com alvo OSSEC Agent	32
Figura 5.4 – Geração de alerta pelo OSSEC Agent - Resultado Scan Nmap	32
Figura 5.5 – Mensagem de alerta OSSEC Agent - Resultado Scan Nmap	32
Figura 5.6 – Comando nmap com alvo Wazuh Agent	33
Figura 5.7 – Alerta resultado nmap com alvo Wazuh Agent	34
Figura 5.8 – Mensagem resultado nmap com alvo Wazuh Agent	34
Figura 5.9 – Firewall do Windows desligado	35
Figura 5.10–Firewall do Windows desligado - Rede Pública	35
Figura 5.11–Comando nmap com alvo Windows - OpenEdr Agent	36
Figura 5.12–SSH Brute-Force com alvo Linux Ubuntu 16.04 - OSSEC Agent	37
Figura 5.13–Mensagem de alerta Agente OSSEC - Usuário Inválido	37
Figura 5.14–Mensagem de alerta Agente OSSEC - Múltiplas falhas na tentativa de autenticação	37
Figura 5.15–SSH Brute-Force com alvo Linux CentOS 7 - Wazuh Agent	37
Figura 5.16–Mensagem de alerta Agente Wazuh - Usuário errou a senha mais de uma vez	38

Figura 5.17–Mensagem de alerta Agente Wazuh - Técnica Brute Force sob a tática <i>Credential Access</i>	38
Figura 5.18–Ativação do Remote Desktop	39
Figura 5.19–Topologia com a máquina atacante Windows conectada ao Firewall	39
Figura 5.20–Inserção do endereço de IP da máquina Windows a ser conectada	40
Figura 5.21–Tentativa de login com credencial errada no alvo Windows 10	40
Figura 5.22–Falha na tentativa de login com credencial errada no alvo Windows 10	40
Figura 5.23–Não foram gerados logs relativos ao ataque RDP Brute-Force pelo OpenEDR .	41
Figura 5.24–Ocultação e exibição do diamorphine a partir do envio do sinal 63 para um pid aleatório	42
Figura 5.25–Mensagem de alerta agente OSSEC - Problema desconhecido no sistema	42
Figura 5.26–Configuração adicional nos Agentes	43
Figura 5.27–Comando utilizado no ataque ShellShock	43
Figura 5.28–Retorno em resposta ao comando realizado - Ossec	44
Figura 5.29–Retorno em resposta ao comando realizado - Wazuh	45
Figura 5.30–Alarme gerado em resposta a tentativa de ataque	46
Figura 6.1 – Log relativo ao login de usuário via RDP no endpoint Windows	47
Figura 6.2 – Log obtido pelo Event Viewer	48
Figura 6.3 – Registros do arquivo access.log do agente OSSEC	49
Figura A.1–Gerenciador de agentes do OSSEC+	58
Figura A.2–Adição do agente ao servidor OSSEC+	58
Figura A.3–Obtenção da chave de pareamento entre o gerenciador e o agente	58
Figura B.1–Importação da chave de pareamento no agente OSSEC	59

LISTA DE ABREVIATURAS

Acrônimos

APT	Advanced Persistent Threat
BSD	Berkeley Software Distribution
CID	Confidencialidade, Integridade e Disponibilidade
ELK	Elasticsearch, Logstash e Kibana
EDR	Endpoint Detection and Response
GNS3	Graphical Network Simulator
HIDS	Host Intrusion and Detection System
IPS	Intrusion Prevention System
JSON	JavaScript Object Notation
MFA	Multi-factor authentication
NAT	Network Address Translation
NIST	National Institute of Standards and Technology
RDP	Remote Desktop Protocol
SIEM	Security Information and Event Management
SSH	Secure Shell
UnB	Universidade de Brasília
VPN	Virtual Private Network
XDR	Extended Detection and Response

1 Introdução

Com o avanço dos sistemas digitais e principalmente da Internet, as sociedades se mostram cada vez mais dependentes das redes de comunicação para operarem. Atualmente, existem mais de 5,1 bilhões de usuários ativos na Internet, é estimado que 90% de todos os dados presentes na Internet foram gerados entre 2019 e 2022 e estudos indicam que, até 2040, 95% das compras serão realizadas de forma online [1] [63]. A pandemia do COVID-19 impulsionou ainda mais essa dependência, já que o trabalho remoto foi amplamente adotado em uma escala global. Segundo pesquisas realizadas pela McKinsey em 2020, empresas aceleraram a digitalização tanto dos seus processos internos quanto os de interação com o cliente em até quatro anos num período menor do que seis meses devido à crise [28].

Em conjunto com esse crescimento do uso da tecnologia e de sua dependência, está o crescimento do número de ataques cibernéticos. Em 2020 houve um aumento de 358% em ataques por malware, que danificam sistemas e roubam dados sensíveis, enquanto que os por ransomware, que demandam pagamento em troca da devolução de uma funcionalidade roubada, aumentaram em 435% [23]. Somente em 2020, atacantes arrecadaram 406,34 milhões de dólares em criptomoedas a partir de ataques de ransomware [23]. A situação é tão crítica que o Fórum Econômico Mundial considera os ciberataques o segundo risco mais preocupante para o comércio global nos próximos dez anos [74].

Os ataques cibernéticos, além de serem cada vez mais frequentes, estão cada vez mais sofisticados. Ameaças como Advanced Persistent Threats (APTs), que são infiltrações geralmente de longa duração realizadas por atacantes altamente organizados com objetivos bem definidos, vêm crescendo e são de difícil detecção. Muitos desses ataques utilizam ameaças sem arquivo ou Fileless, que é um tipo de malware que não deixa registro no sistema de arquivos da vítima, impossibilitando a detecção da invasão por soluções baseadas em assinatura, como antivírus [35].

A segurança da informação vem evoluindo com a necessidade de proteção contra ataques cibernéticos, porém, muito mais atenção ainda é dada às defesas que precedem o ataque em detrimento das defesas que o sucedem. Soluções como Firewalls, Network Access Control e MFA são comumente empregadas em redes corporativas e têm um papel importante para impedir que ataques ocorram. Pode-se dizer o mesmo de boas práticas de segurança, como o uso de senhas fortes e o seu posterior rotacionamento. No entanto, caso o ataque já tenha sido efetuado com sucesso, isto é, caso já tenha havido uma infiltração na rede, essas defesas não são as ideais para combatê-lo. Por outro lado, soluções como antivírus são as designadas justamente para esse tipo de situação (que sucedem o ataque) e são largamente empregadas. Infelizmente, com a proliferação de ameaças cada vez mais sorrateiras e ardilosas como as APTs, os antivírus por si só não são capazes de prover uma proteção adequada às redes de comunicação.

Para combater essas novas ameaças é necessário o uso de ferramentas que consigam analisar o comportamento dos dispositivos finais e fornecer um diagnóstico com relação às ações executadas

nos mesmos. Como os ataques estão cada vez mais sutis e deixando cada vez menos rastros, uma varredura no sistema de arquivos não será suficiente para constatar uma invasão ou não. É necessário correlacionar diversos logs gerados pela máquina com o intuito de averiguar se comportamentos potencialmente maliciosos estão sendo exibidos pelos hosts. Caso essas ferramentas de fato observem ações suspeitas, alertas são emitidos e, em certos casos, ações são tomadas para mitigar a ameaça. Soluções conhecidas como EDRs (Endpoint Detection and Response) se enquadram nessa categoria de ferramentas e elas focam em prover esse tipo de segurança a um endpoint, como um Desktop ou um Notebook.

1.1 Motivação

Sabendo do avanço da tecnologia, que pode ser utilizada tanto para a defesa, quanto para o ataque, mostra-se essencial o desenvolvimento de ferramentas e testes que possam monitorar, relacionar e alertar eventos adversos, sendo dados como possíveis ameaças.

Portanto, constata-se, claramente, a necessidade de garantir a segurança de sistemas por toda a sua integridade, dando atenção de forma especial e minuciosa às partes mais vulneráveis. Dentre estas, destaca-se o elemento humano, pois, sendo um usuário de sistemas de informação, pode facilitar a exploração de ameaças [20]. Com isso, torna-se fundamental o estudo e uso de ferramentas especializadas na segurança dos dispositivos finais.

1.2 Objetivos

1.2.1 Objetivo Geral

Desenvolver cenário virtualizado e controlado com a finalidade de testar ferramentas de EDRs, promovendo a análise de eficiência e comparação entre os resultados obtidos em cada situação proposta.

1.2.2 Objetivos Específicos

Para obter os resultados propostos, são necessários alguns objetivos específicos:

- Desenvolver arquitetura de ambiente virtualizado e controlado, de forma a possibilitar os passos seguintes.
- Coleta dos logs necessários para as análises nos próprios EDRs.
- Configuração e realização de testes para disponibilizar os logs obtidos pelos EDRs na pilha ELK.
- Realizar ataques às máquinas com agentes e analisar o desempenho das ferramentas utilizadas.

2 Referencial Teórico

2.1 Segurança da Informação

A Segurança da Informação tem sua origem em meados de 1950, quando obtém-se compreensão do que os dados representam e, portanto, da necessidade de mantê-los protegidos.

Com a aderência aos sistemas tecnológicos, pôde-se perceber o valor intrínseco que os dados contêm, e, também, a sensibilidade deles, dado que certas informações deveriam ser acessadas somente pelos seus donos, ou pessoas autorizadas. A situação tornou-se ainda mais evidente e crítica com o advento dos acessos online e da internet, quando dados começaram a ser compartilhados e vendidos. Assim, os riscos adjacentes a manter e manipular dados elevaram-se, chegando a atrair a atenção de criminosos, culminando com o aparecimento dos crimes cibernéticos [54].

Torna-se, então, imprescindível o cuidado e esforço contínuo para garantir o correto acesso e uso de dados, programas e até mesmo equipamentos. Mas, além destes cuidados com o manutenção da segurança através da tecnologia, mostra-se, também, essencial a conscientização de todos sobre práticas seguras, fazendo com que certos protocolos sejam empregados pela completude dos usuários, mesmo os não especializados na área.

O alicerce da Segurança da Informação traz em sua essência os conceitos da tríade CID [15]. Esta é formada pelo emprego, em conjunto, da Confidencialidade, garantindo que os dados sejam acessados somente por pessoas autorizadas, da Integridade, que assegura a precisão e consistência das informações, e, também, da Disponibilidade, sustentando que aqueles autorizados consigam acessar os dados.



Figura 2.1 – Tríade CID [40]

2.1.0.1 Confidencialidade

A confidencialidade, de maneira sucinta, tem por finalidade garantir a privacidade dos dados. Para isso, é preciso que somente as pessoas autorizadas tenham acesso às informações em questão, impedindo o acesso, ou até mesmo divulgação, de dados para pessoas, recursos ou processos não

autorizados. Este princípio pode ser alcançado por meio de controles de acesso, uso de autenticação e métodos de criptografia.

2.1.0.2 Integridade

Ao tratar-se da integridade, é preciso garantir a consistência, precisão e confiabilidade dos dados em todo o seu ciclo de vida. É importante assegurar que as informações presentes não tenham sido alteradas por alguém e, de fato, foram escritas por quem é determinado como autor. Para que isso seja possível, são utilizados alguns mecanismos, dentre estes destaca-se o uso do hashing.

O hash, quando utilizado de forma correta, permite a criação de um “resumo” da informação, sendo este único para cada texto e irreversível. Com isso, não é possível obter o texto a partir do hash gerado, pois trata-se de uma função não inversível [18]. Outras metodologias também são utilizadas para garantir a integridade, como mecanismos para a verificação dos dados, para a verificação da consistência e ainda para o controle de acesso.

2.1.0.3 Disponibilidade

O princípio da disponibilidade visa assegurar que os dados estejam acessíveis para todos aqueles que devem ou podem acessá-las. Para garantir que uma informação, que deve ser obtida, seja, de fato, obtida, várias formas de evitar falhas podem, e devem, ser empregadas. Com isso, surgem as considerações para a criação de redundâncias de sistemas e backups, além de buscar sempre a manutenção de equipamentos e atualizações de softwares para as versões mais recentes. Há ainda a criação de planos de recuperação rápida de desastres, em que é realizado o estudo sobre como proceder em caso de indisponibilidade, elencando como recuperar os sistemas no menor tempo possível.

2.2 Logs

Do inglês, a palavra *log* tem como um de seus significados o registro de um período de tempo ou de um evento [38]. No caso de sistemas de informação, o significado se mantém. Os logs são, de maneira sucinta, um conjunto de registros de eventos ocorridos dentro de um dado sistema [76]. Cada um desses registros detalha um determinado acontecimento num determinado período. Alguns exemplos de informações que são exibidas por eles são: consumo de recursos, desempenho de aplicações, processos que foram realizados, erros e usuários que acessaram o sistema. Os logs têm várias fontes, como sistemas operacionais, aplicações e softwares de segurança [41]. Na área da segurança da informação, muita ênfase é dada aos logs de segurança, pois eles são imprescindíveis na detecção de ataques e falhas de sistema [8]. Além disso, é através deles que se mensura o desempenho das redes de comunicação e dos sistemas que as compõem [76].

O gerenciamento e análise dos logs, no entanto, não é uma tarefa fácil. Como foi exposto previamente, existem diversas fontes de logs inseridas nas redes de comunicação. Isso contribui não só para uma grande quantidade de logs gerada continuamente, como também para uma grande quantidade de formatos de logs fornecida, já que nem todos os sistemas operam da mesma forma

quando se trata de registros de eventos [8]. Essa combinação de fatores dificulta consideravelmente o trabalho do analista de segurança. Para investigar os acontecimentos na rede, esse profissional tem de examinar uma quantidade desproporcional de arquivos de log em diversos formatos diferentes com uma grande atenção aos diversos detalhes que eles trazem. Além de os logs não serem *user friendly*, muitos deles não fornecem informações úteis para a detecção de anormalidades na rede [76]. Investigá-los manualmente um a um está cada vez mais inviável, principalmente quando não se sabe se de fato houve algum incidente de segurança.

2.3 Security Information and Event Management

Apesar de os logs serem fontes de informações de extrema importância, contendo informações cruciais para a análise de diversos eventos, inclusive aqueles relacionados a segurança, torna-se demasiado difícil e oneroso acompanhar e encontrar logs de eventos específicos em meio a grande quantidade de logs e de sistemas diferentes utilizados. Levando-se em consideração esta dificuldade, criou-se, então, o SIEM, Security Information and Event Management. O SIEM origina-se, portanto, com o intuito de facilitar a visibilidade e controle de todos os logs, permitindo a centralização dos eventos criados por diversos equipamentos, e, assim, tornando a análise dos logs mais acessível e objetiva [46].

A utilização do SIEM permite a apuração em tempo real, a melhor detecção e resposta a incidentes de segurança [47]. Neste, encontra-se a possibilidade de criação de regras a serem testadas e comparadas com os logs recebidos. Com isso, tem-se a possibilidade de programar a geração de alertas e relatórios, mostrando de forma mais eficaz casos de possíveis falhas de segurança. Assim, diminui-se o ônus de procurar eventos adversos dentre todos os ocorridos, permitindo ao profissional da segurança analisar os logs de forma mais eficiente e direta.

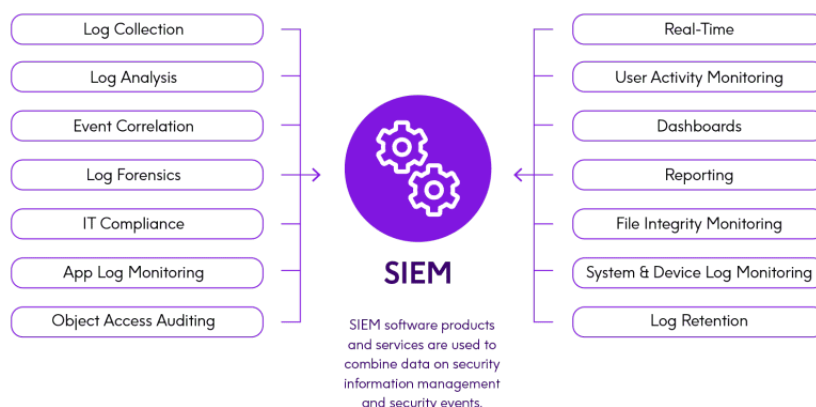


Figura 2.2 – SIEM [57]

Com a evolução da tecnologia, mostra-se, ainda, possível a melhoria das ferramentas até então utilizadas [16]. Os SIEMs passaram a incluir funções para a análise de dados analíticos, agregando o conhecimento de inteligência artificial, proporcionando melhor gerenciamento e classificação de eventos tidos como adversos [29].

2.4 Ameaças, Vulnerabilidades e Riscos

A internet tem sua origem datada em meados da década de 60, sendo criada dentro de laboratórios de pesquisa. Com o decorrer dos anos e com a ajuda dos avanços tecnológicos, nasce a ARPANET, que interligava instituições de pesquisa nos Estados Unidos [27]. No seu início, a internet era pouco difundida e utilizada apenas, praticamente, nos meios acadêmicos, sendo estes considerados confiáveis e seguros [14]. As progressões continuaram e nos anos 90 desenvolveu-se a World Wide Web, que, logo em seguida, foi aberta ao público externo [27].

Com uma quantidade cada vez maior de pessoas tendo acesso a internet, eleva-se o número de vulnerabilidades conhecidas [39] e, ainda, de pessoas más intencionadas que passam a utilizá-la como ferramenta. Assim, percebe-se a importância não apenas de evoluir os sistemas de comunicação, mas também de garantir a segurança dos mesmos.

Assim, no estudo da Segurança da Informação passam a ser reconhecidas diversas formas, meios e atores que podem comprometer a segurança da informação. Dentre estas destacam-se três definições de grau elevado para a gestão da segurança, as ameaças, os riscos e as vulnerabilidades:

- Ameaças: tidas como quaisquer circunstâncias ou eventos que podem levar ao potencial de interferir e causar danos para um ativo [17], causando a quebra da garantia da tríade CID [48].
- Vulnerabilidades: fraquezas presentes em sistemas de informação, procedimentos, controles internos, design ou implementações que podem ser exploradas por ameaças, a fim de comprometer um ativo [65].
- Riscos: definidos utilizando os conceitos de ameaça e vulnerabilidade. Um risco representa a probabilidade de que uma vulnerabilidade venha a de fato ser explorada por uma ameaça causando o comprometimento de um ativo, e resultando em consequências indesejáveis [12].

Sendo assim, as empresas adotam estratégias de gestão de risco, levando em consideração não somente o risco, mas também o custo operacional para garantir as medidas de proteção e os ganhos captados com estas. São utilizadas quatro formas de gestão de risco: a aceitação do risco, quando o custo supera o benefício da garantia de proteção, a prevenção de riscos, evitando a exposição ao risco, a redução do risco, que busca reduzir a possibilidade de exploração ou o impacto, e a transferência de risco, em que o risco é passado a um terceiro, por exemplo uma seguradora.

Outro quesito importante com relação a uma ameaça é o seu ator e, também, a sua motivação. A palavra “hacker” é usada comumente para caracterizar o responsável por uma ameaça, porém, dentro do estudo da segurança da informação, tem-se a classificação e diferenciação de variados hackers. São frequentemente utilizados três tipos de classificação:

- Hacker White Hat: conhecido como *ethical hacker*, é um trabalhador da área de segurança, que possui conhecimento apurado na área, sabendo atuar na exploração de ameaças, mas que utiliza suas habilidades de forma legal, ética e com boas intenções, permitindo o aprimoramento de medidas de segurança.

- **Hacker Gray Hat:** é um indivíduo que pode vir a cometer crimes e realizar ações consideradas como antiéticas, porém sem a finalidade de ganho pessoal ou causar danos.
- **Hacker Black Hat:** dado como criminoso, realiza ações antiéticas, violando sistemas de informação, por meio do uso da exploração de ameaças, com o intuito de ganho pessoal ou por motivos maliciosos.



Figura 2.3 – Tipos de hackers [26]

A motivação dos Hackers Gray Hat e Black Hat pode ser bastante variada, mas ressaltam-se três perfis conhecidos que estão presentes nas atividades cibercriminosas. Os hacktivistas, utilizam seus conhecimentos por motivacional político e social, manifestam-se e protestam contra organizações e governos de forma a, por exemplo, publicar artigos, vazam informações confidenciais e realizar ataques para gerar indisponibilidade de sistemas de informação. Os cibercriminosos são conhecidos por atuarem de forma autônoma ou em organizações voltadas ao crime cibernético, buscando o ganho pessoal. Tem-se, ainda, os hackers patrocinados por Estados, que roubam informações confidenciais de outros governos, coletam informações e sabotam sistemas de informação de governos estrangeiros.

2.5 Advanced Persistent Threat

Segundo o NIST (*National Institute of Standards and Technology*), as APTs são ataques caracterizados por serem executados por adversários extremamente qualificados e possuidores de recursos significativos capazes de possibilitá-los a alcançarem os seus objetivos através de diversos vetores de ataque [3]. Os atacantes são geralmente altamente organizados e possuem propósitos e alvos muito bem definidos [11]. Suas vítimas são tipicamente governos ou entidades relevantes que possam vir a possuir informações valiosas, como dados de segurança nacional e propriedade intelectual de extremo valor [11]. A intenção dos atacantes geralmente é se estabelecer na rede e ir expandindo o seu domínio dentro dela para, assim, roubar dados ou prejudicar o desempenho de sistemas.

As máquinas alvo dos atacantes vão consistir geralmente em dispositivos críticos da rede a ser penetrada. Obviamente, a proteção delas será mais robusta, visto que possuem um alto grau de importância dentro da estrutura da organização. Conseqüentemente, iniciar um ataque a partir

das mesmas é muito mais complicado. Com isso em mente, os atacantes focam em máquinas de menor relevância para penetrar a rede. Quando assumem o controle delas, os invasores vão tentando elevar privilégios e contaminar outros dispositivos até que consigam acesso à máquina alvo.

Durante este processo, os atacantes assumem uma postura sorrateira e buscam agir o mínimo possível dentro dos sistemas invadidos a fim de permanecerem não descobertos, o que contribui para a longa duração dessa categoria de ataque. Essa postura é complementada pelo uso de técnicas evasivas para dificultar ainda mais a detecção da invasão. Vulnerabilidades de dia zero, que são brechas descobertas em sistemas antes que haja uma correção por parte dos desenvolvedores, são comumente exploradas nas APTs [11]. O emprego desse tipo de vulnerabilidade em conjunto com as ameaças sem arquivo evita a detecção do ataque por soluções baseadas em assinatura, como antivírus [33].

2.6 Endpoint Detection and Response

Os produtos de *Endpoint Detection and Response* são soluções de segurança integradas responsáveis pelo monitoramento constante dos endpoints [64], isto é, das máquinas que ficam nas extremidades das redes. Quando algum comportamento estranho é detectado pela ferramenta, alertas são disparados. Enquanto soluções de antivírus dependem de assinaturas de malware para detectar ataques, os EDRs são capazes de identificá-los a partir de indicadores de comprometimento (IOC), indicadores de ataque (IOA) e comportamentos anômalos [37]. Este é o grande diferencial dessas ferramentas: conseguir detectar ameaças a partir da observação de comportamentos anormais. Com isso, ataques inéditos, como explorações de dia zero, podem ser detectados antes que sejam finalizados, possibilitando uma redução considerável do prejuízo causado pela invasão [37]. As capacidades apresentadas pelos EDRs os fazem a solução canônica para combate às APTs [64].

O funcionamento dessas ferramentas varia, mas grande parte das soluções dessa categoria no mercado segue um padrão conhecido. Os EDRs coletam as informações dos endpoints e as enviam para um servidor centralizado que irá armazená-las e processá-las [33]. Nele, os diversos dados coletados são correlacionados e conferidos um certo nível de criticidade [33]. Essa atribuição é baseada numa coleção de regras definidas por analistas de segurança ou - em soluções mais avançadas - em *machine learning* e inteligência artificial [33] [37].

Apesar de possuírem diversas funcionalidades e estarem em constante evolução, é importante conhecer também os pontos fracos apresentados por essas soluções a fim de utilizá-las com sabedoria. Um dos problemas presentes nos EDRs é a geração de alarmes falsos, que, além de inconvenientes, podem resultar na ofuscação de alarmes válidos [64]. Alarmes falsos podem ser gerados por ações inofensivas de usuários que caracterizam técnicas conhecidas de ataque, como modificação de permissões em diretórios ou exclusão de arquivos [64]. Isso contribui para outro problema dos EDRs, que é a de análise de logs. Apesar de essas ferramentas otimizarem consideravelmente o processo de investigação, ainda é necessário que o analista de segurança verifique a autenticidade dos vários eventos que geraram os alertas. O último problema que deve ser levado

em consideração é o de retenção de logs a longo prazo [64]. Como muitos dados são gerados pelos endpoints continuamente, é necessário muito poder de armazenamento para retê-los. O analista de segurança deve então escolher entre ter um custo a mais e armazená-los por mais tempo ou economizar dinheiro e arriscar perder logs que possam vir a ser cruciais numa investigação de incidente de segurança.

3 Ferramentas Utilizadas

Esta seção detalha as ferramentas empregadas para a execução do projeto. A maior parte dos componentes aqui explicitados estão rodando em máquinas virtuais de sistemas operacionais Windows e Linux de diversas distribuições no ambiente utilizado de emulação de rede.

3.1 GNS3

O *Graphical Network Simulator 3*, mais conhecido como GNS3, é um software gratuito e de código aberto destinado à emulação, configuração e teste de redes reais e virtuais [25]. A partir dele, é possível criar topologias contendo máquinas virtuais diversas de forma rápida e intuitiva. A sua arquitetura consiste em dois componentes: o software GNS3-all-in-one e a máquina virtual GNS3 [25]. O primeiro é relativo à parte cliente da arquitetura e possui interface gráfica, enquanto que a segunda é relativa à parte servidor da arquitetura [25]. Existem três opções para o servidor quando se emprega o GNS3: servidor local GNS3, máquina virtual GNS3 local ou máquina virtual GNS3 remota [25]. O servidor local GNS3 roda na mesma máquina na qual está instalado o GNS3 [25]. A máquina virtual local GNS3 também roda na mesma máquina na qual está instalado, empregando um software de virtualização, como o VMware Workstation. Já a máquina virtual GNS3 remota roda num servidor externo utilizando VMware ESXi [25]. A vantagem de se utilizar uma máquina virtual GNS3 remota é que o computador no qual está instalado o GNS3-all-in-one não precisa ter muitos recursos para rodar as diversas máquinas virtuais inseridas nas topologias criadas. Neste caso, os recursos do servidor, que geralmente são mais abundantes, que serão amplamente utilizados.

3.2 EDRs

Muitas empresas são vítimas de ataques cibernéticos, pois não possuem uma visibilidade adequada do comportamento de seus dispositivos finais. Essa lacuna é explorada por atacantes veementemente, executando ataques prolongados e devastadores que ficam ocultos por tempo suficiente para que a única opção restante dos defensores seja ceder às suas demandas. Felizmente, soluções de EDR foram introduzidas no mercado a fim de resolver esse déficit que é constantemente explorado por cibercriminosos. No entanto, várias das ferramentas desta categoria possuem preços inacessíveis para empresas de menor porte, o que torna importante conhecer as alternativas gratuitas que também são oferecidas.

Assim sendo, serão empregadas três soluções conhecidas de EDR de código aberto gratuitas a fim de comparar os seus desempenhos frente a ameaças e ataques que assolam os ecossistemas das redes de computadores.

3.2.1 OSSEC+

Baseado no HIDS (Host Intrusion and Detection System) OSSEC, o OSSEC+ é uma solução gratuita e de código aberto que visa proteger os dispositivos das redes de comunicação [24]. Além das capacidades de detecção de rootkit e malware, monitoramento de integridade de arquivos, resposta ativa e detecção de invasão baseada em logs presentes no OSSEC, o OSSEC+ apresenta funcionalidades adicionais relevantes, como integração à pilha ELK e o uso de *machine learning* [24].

A sua arquitetura é baseada em um nó gerente que administra um ou mais nós agentes [52]. O nó gerente ou servidor é o nó que possui o programa servidor do OSSEC+ instalado. Ele armazena e centraliza todas as informações coletadas pelos agentes aos quais ele está associado, como logs, eventos e bases de dados verificadoras de integridade de arquivos [52]. As configurações principais, como regras e decodificadores, também ficam armazenadas no servidor [52]. Os nós agentes, por outro lado, são os hosts que possuem o agente OSSEC instalado. O agente coleta as informações obtidas a partir dos sistemas do host e as envia para o servidor armazená-las e processá-las [52]. É importante ressaltar que o OSSEC+ existe somente para o nó servidor. Os nós agentes continuam rodando o OSSEC.

O agente OSSEC é compatível com diversos sistemas operacionais como Windows, Fedora, Debian, Ubuntu, Rocky Linux 8 e CentOS, enquanto o servidor OSSEC+ é compatível com Rocky Linux, Ubuntu e CentOS [13]. Ainda é possível obter visibilidade de dispositivos que não se enquadram nesses sistemas operacionais a partir do modo *agentless*, destinado a máquinas as quais o agente OSSEC não pode ser instalado [58]. Roteadores que possuem o Cisco IOS e Firewalls da Checkpoint, por exemplo, podem ter seus logs analisados e a integridade de seus arquivos checada pelo nó servidor a partir deste modo [58].

3.2.2 Wazuh

O Wazuh é uma solução de segurança gratuita e de código aberto de XDR (*Extended Detection and Response*) e SIEM que fornece proteção a endpoints [68]. Ele possui diversos casos de uso e ainda se enquadra em normas e padrões de segurança de dados, como o PCI DSS e o HIPAA [69]. Essa solução de EDR pode ser utilizada para análise de logs, detecção de rootkits, monitoramento de integridade de arquivos, detecção de vulnerabilidades e resposta ativa, por exemplo [69].

O Wazuh consiste de quatro componentes: o Wazuh Agent, o Wazuh Server, o Wazuh Indexer e o Wazuh Dashboard [68]. O Wazuh Agent é justamente o agente instalado nos endpoints, enquanto que o Wazuh Server é o gerente dos diversos agentes nos endpoints que estão relacionados a ele. Já o Wazuh Indexer é um motor de busca e análise escalável. Por fim, o Wazuh Dashboard é uma interface web destinada à visualização e análise dos eventos de segurança coletados e alertas gerados. O único componente que não será usado no projeto é o Wazuh Dashboard, já que a pilha ELK será empregada. Essa solução é compatível com sistemas operacionais Linux de 64 bits, sendo recomendadas as distribuições CentOS e Ubuntu [68].

A arquitetura do Wazuh é bem semelhante à do OSSEC+. Os agentes monitoram as atividades dos endpoints e encaminham os dados coletados ao Wazuh Server para processamento [67]. Em

seguida, os resultados obtidos desse processo são direcionados ao Wazuh Indexer para classificação e armazenamento [67]. Assim como no OSSEC+, há a possibilidade de se empregar o modo agentless em certos dispositivos como Firewalls e pontos de acesso [67].

3.2.3 OpenEDR

Desenvolvido pela fabricante de soluções de segurança da informação Comodo, OpenEDR é uma ferramenta gratuita e de código aberto de *Endpoint Detection and Response* que coleta informações detalhadas das atividades de sistema de endpoints Windows [30] [49]. Além disso, a ferramenta emprega sistemas de monitoramento de arquivos, análise e decisão desenvolvidos pela própria fabricante a fim de prover mais visibilidade ao cliente [49]. A sua arquitetura difere das outras soluções abordadas neste documento por empregar apenas um agente em sua composição, ao invés de um agente e um servidor [49]. Os logs coletados ficam armazenados no próprio endpoint com o agente instalado, podendo ser enviados a plataformas de análise a partir de qualquer ferramenta de envio de logs [49].

3.3 Pilha ELK

A pilha ELK é dada como a utilização do Elasticsearch, Logstash e Kibana de forma conjunta, criando-se uma ferramenta de busca e análise, que pode ser caracterizada como uma solução de gerenciamento de logs [2]. Este conjunto tem o potencial de ser utilizado de forma a ser comparado com uma ferramenta de SIEM [44], em que concentram-se, de forma centralizada, todos os logs dos sistemas utilizados, sendo possível buscar, analisar, criar alertas, gerar relatórios e indicadores [43] a partir dos dados obtidos.

O Elasticsearch é desenvolvido sobre o Apache e teve sua primeira versão apresentada em 2010. É conhecido por ser um projeto open source, distribuído, do tipo Restful e baseado em JSON. Este componente é encarregado, dentro da pilha, de permitir e facilitar as ações de busca e análise [43], tendo alto índice de escalabilidade e flexibilidade. Isto é possível graças a tecnologia utilizada no funcionamento da ferramenta. Quando os dados chegam ao Elasticsearch, estes são submetidos ao processo de ingestão de dados, sendo analisados, normalizados e enriquecidos. Em seguida, os dados tratados são, então, indexados, sendo, por fim, utilizados para executar consultas complexas de forma simples e rápida.

Assim como o Elasticsearch, o Logstash também é uma ferramenta aberta e gratuita, podendo este ser definido como um *pipeline* para processamento de dados, usado para agregar, processar e enviar os dados para o Elasticsearch. É um componente que se encontra do lado do servidor e recebe logs de diversas fontes diferentes ao mesmo tempo. O Logstash utiliza, então, filtros, analisando cada evento, determinando os campos presentes para desenvolver uma estrutura destes dados. Em seguida, usa-se a estrutura criada para transformar os dados em um formato comum, permitindo uma análise mais eficiente [10].

O Kibana foi desenvolvido em 2013 pela comunidade Elasticsearch, tratando-se, assim como os outros componentes da pilha, de um projeto *open source*, sendo responsável pelo desenvolvimento

front-end do conjunto. Esta ferramenta viabiliza a existência da interface gráfica que é utilizada para fornecer busca, visualização, criação de gráficos e gerenciamento dos dados presentes no Elasticsearch, tarefa realizada por meio do uso de índices, que são criados através da ingestão de dados, convertendo-os para um formato comum e estruturado [45].

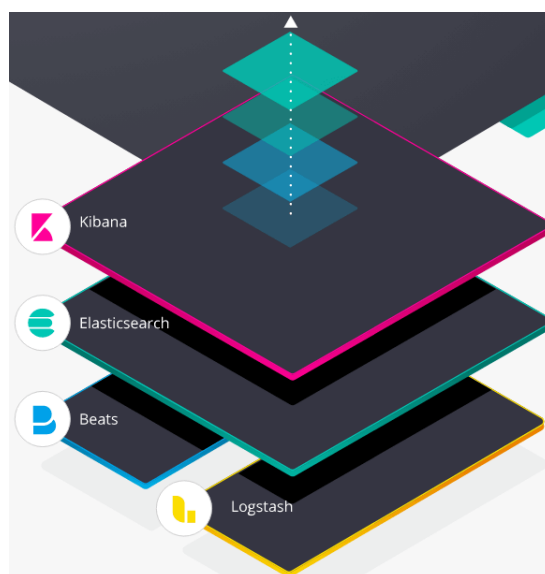


Figura 3.1 – Pilha ELK [44]

Com o decorrer dos anos, passou-se a perceber a vontade de uma forma um pouco mais simples de implementação, permitindo o uso de um componente diferente do Logstash. Assim, em 2015 foi apresentado o Beats, um projeto também caracterizado por ser *open source*, com a finalidade de enviar os dados necessários diretamente para o Logstash ou ao próprio Elasticsearch, tornando o envio direto e mais fácil [4]. Desse modo, a pilha ELK, passou a ser conhecida também como pilha Elastic.

3.4 pfSense

O pfSense é um software *open source* e gratuito [31], desenvolvido e hospedado pela Rubicon Communications LLC, conhecida como Netgate, uma empresa orientada a projetos de software aberto com o intuito de fornecer ferramentas para garantir a segurança de redes [60]. Ele foi disponibilizado pela primeira vez em 2004 com a finalidade de ser utilizado como um firewall e roteador.

O nome do projeto surge com a ideia de entender e dar sentido ao PF, “making sense of pf” [70], que é software de filtragem de pacotes hospedado em FreeBSD, um sistema operacional do tipo Unix-like que descende do chamado Research Unix via the Berkeley Software Distribution (BSD), tendo características avançadas de rede, segurança e armazenamento[61][72].

Por se tratar de um firewall, o pfSense permite a filtragem de pacotes [22], que decide se um tráfego de entrada ou saída será bloqueado. A sua decisão é baseada no estado da conexão, nas portas utilizadas e, ainda, nos protocolos empregados [71]. O administrador pode criar regras

bem definidas no firewall, bloqueando ou permitindo tráfegos específicos, mas também é possível configurar o firewall de forma que este consiga levar em consideração o contexto, utilizando-se o histórico de conexões prévias para a decisão, bloqueando, assim, conexões que apresentem alguma suspeita.

Além de ser *open source*, o pfSense apresenta ainda muita flexibilidade ao usuário, possibilitando customização, de forma a atender melhor suas necessidades específicas. Além disso, é regularmente atualizado, inclusive com *patches* para garantir a melhor segurança atual [59], possui interface web, que facilita o seu gerenciamento e permite, também, o uso de pacotes de terceiros [53], como o Snort, um IPS, Intrusion Prevention System, *open source*, que ajuda a prevenir contra atividades maliciosas na rede [73].

3.5 VyOS

O VyOS é um sistema operacional de rede gratuito e de código aberto baseado em Debian e no software de roteamento Quagga [66]. Ele pode ser usado tanto como um roteador, quanto como um Firewall e ainda possui funcionalidades voltadas para VPN [66]. A sua CLI e a sua sintaxe de configuração são vagarosamente derivadas do sistema operacional Junos da Juniper [66]. É possível rodá-lo tanto em plataformas físicas, como servidores de grande porte e máquinas baseadas em x86, quanto em plataformas virtuais, como VMware e Hyper-V [66].

3.6 Cyber Kill Chain

Desenvolvida pela Lockheed Martin, o cyber kill chain é um modelo elaborado para equipes de segurança da informação mapearem e analisarem as ações ofensivas tomadas pelos atacantes de maneira estruturada [75] [34]. Conhecer esse modelo ajuda o analista de segurança a entender o raciocínio do atacante durante qualquer que seja a etapa em que o ataque se encontre, possibilitando tanto uma melhor análise dos movimentos passados do invasor quanto uma melhor preparação da defesa para o que pode estar por vir. O cyber kill chain foi elaborado a partir do ponto de vista do atacante e consiste de sete fases: Reconhecimento, Armamento, Entrega, Exploração, Instalação, Comandar e Controlar e Agir no Objetivo.

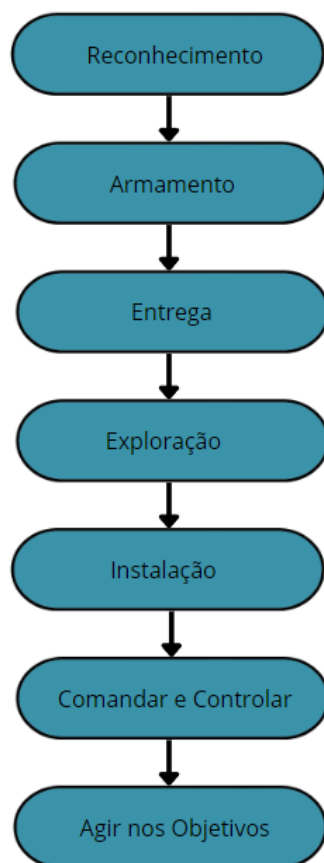


Figura 3.2 – Fases da cyber kill chain

A fase de Reconhecimento é caracterizada pela identificação, escolha e estudo do alvo [75]. O atacante realiza pesquisas tanto sobre funcionários da empresa a ser atacada quanto sobre suas máquinas e suas configurações. Plataformas como o LinkedIn podem ser usadas para coletar dados sobre os empregados, enquanto que ferramentas como o Shodan podem ser aplicadas para coletar dados sobre servidores conectados à Internet. Scans dentro da rede também podem ser executados a fim de se obter informações sobre as máquinas ali inseridas. No entanto, são mais arriscados por poderem ser detectados por ferramentas de monitoramento [75].

A fase de Armamento é a de concepção das armas cibernéticas a serem empregadas [75]. Elas são projetadas tendo em vista as vulnerabilidades percebidas a partir da fase de Reconhecimento. Duas ferramentas principais são configuradas até o final desta etapa: a ferramenta de controle remoto e a ferramenta de exploração [75]. A primeira é a destinada a prover acesso remoto à máquina alvo para o atacante de forma não detectada, possibilitando a ele, por exemplo, monitoramento do teclado, exploração do sistema, upload, download e execução de arquivos [75]. A segunda ferramenta é a que possibilita a execução da ferramenta de controle remoto. Ao se aproveitar das vulnerabilidades identificadas na fase anterior, a ferramenta de exploração despista ferramentas de monitoramento, estabelece uma *backdoor* no sistema da vítima, instala e executa a ferramenta de acesso remoto [75]. A ferramenta de exploração pode ser tanto um documento com extensão .doc ou .pdf quanto páginas web [75].

Em seguida, vem a fase de Entrega, que consiste na transmissão da arma cibernética ao

ambiente alvo [75]. Geralmente, a arma é camuflada de modo a fazer com que a vítima acredite que ela seja algo legítimo. Técnicas como *phishing* e DNS Spoofing, por exemplo, são empregadas nesta etapa a fim de realizar a entrega [75].

Com a entrega do arquivo malicioso (*payload*) sendo realizada com sucesso, a etapa seguinte é a de Exploração, que consiste na exploração de vulnerabilidades para executar códigos na máquina da vítima [75]. Algumas condições devem ser respeitadas para que o atacante obtenha sucesso nesta fase. A primeira é que o software ou sistema operacional utilizado pela vítima tem de ser o mesmo que o atacante se baseou para desenvolver a ferramenta de exploração [75]. A segunda é que o software ou o sistema operacional utilizado pela vítima não deve ser atualizado para versões que corrijam a vulnerabilidade explorada [75]. A terceira e última condição é que os mecanismos de segurança da vítima, como antivírus, não devem ser capazes de detectar os arquivos maliciosos [75].

Logo após, vem a fase de Instalação, onde o malware será instalado no alvo. Com os avanços na segurança dos hosts, os atacantes foram forçados a inovarem na forma como projetam os seus arquivos maliciosos a fim de evitar que eles fossem detectados após a sua instalação. Muitos malwares, por exemplo, desabilitam ferramentas de detecção como antivírus e mudam a configuração de DNS da máquina para evitar com que novas atualizações que potencialmente corrijam a vulnerabilidade sejam realizadas [75].

A sexta etapa é a de Comandar e Controlar, caracterizada pela comunicação remota e sorrateira entre o atacante e a máquina alvo através do malware instalado [75]. A partir desse canal de comunicação, o atacante consegue executar comandos e roubar dados. Com o encerramento desta etapa, o ambiente está pronto para a última fase do ataque.

Finalmente, chega-se à fase de Agir nos Objetivos, onde os comandos serão executados baseados no interesse do atacante. Os ataques a serem realizados podem consistir na distribuição de malware pela rede a fim de se obter informações sensíveis, na encriptação de arquivos da vítima a fim de extorqui-la ou na danificação de sistemas a fim de prejudicá-la [75].

Equipes de segurança da informação, ao entenderem os passos dos atacantes e as diversas etapas que eles percorrem para executar os ataques por completo, são capazes de melhorarem as suas defesas e analisarem melhor os movimentos dos ciber atacantes. A *cyber kill chain* não é simplesmente um roteiro que os invasores seguem em suas ações, mas também um *framework* para analistas de segurança reforçarem as defesas da rede e mitigarem ameaças que assolam os ambientes digitais.

4 Arquitetura Proposta

Com o objetivo de criar uma topologia de rede adequada para a realização de testes com as ferramentas do tipo EDRs, em que busca-se promover uma análise de eficiência e realizar, também, uma comparação entre os resultados obtidos em cada situação proposta pelos EDRs empregados, mostra-se, aqui, os passos necessários que foram utilizados a fim de alcançar a meta proposta. Explica-se, a seguir, a metodologia empregada, além da topologia e, por fim, as configurações vitais para o funcionamento do projeto.

4.1 Metodologia

Este trabalho busca analisar a eficiência de ferramentas do tipo EDRs ao serem expostas a cenários específicos que refletem a exploração de uma ameaça por um agente. A metodologia escolhida para ser aplicada ao projeto é a de emulação [32], criando-se um ambiente restrito e controlado para a coleta de informações. A emulação foi escolhida, uma vez que torna possível conduzir o experimento com a ajuda de softwares capazes de criar sinteticamente um ambiente virtual similar ao real, mas sem a necessidade dos equipamentos físicos em si, permitindo, ainda, a instalação de softwares reais no projeto criado. No caso em questão, foi utilizado o GNS3, o qual permitiu a virtualização de máquinas do tipo Linux, em específico das distribuições Ubuntu e CentOS, Windows, VyOS, e Firewall pfSense. Nas máquinas virtualizadas, foram instalados os agentes e servidores necessários para o experimento. Foram eles: Agente OSSEC e Servidor OSSEC+, Agente e Servidor Wazuh, Agente OpenEDR e, por fim, Servidor ELK. Os equipamentos físicos que acomodaram as virtualizações estão hospedados na própria UnB, tendo sido instalados e configurados pelos trabalhadores do Laboratório Latitude.

Para a organização e continuidade lógica do projeto, este pôde ser dividido em 5 etapas que foram necessárias para a implementação da emulação:

1. Escolha dos EDRs: a primeira parte realizada foi a busca de EDRs no mercado e escolha dos mesmos. Para as escolhas, levou-se em consideração os sistemas operacionais suportados e o fato de serem ferramentas gratuitas e de código aberto.
2. Configuração do ambiente na UnB: o projeto foi feito acessando remotamente um servidor GNS3 presente na UnB. Para ter acesso a esse ambiente, a equipe de TI do Laboratório Latitude instalou o servidor e, em seguida, criou-se uma VPN, permitindo o acesso externo.
3. Configuração da topologia: instalação e configuração do Agente OSSEC e Servidor OSSEC+, Agente e Servidor Wazuh e Agente OpenEDR.
4. Configuração da pilha ELK: instalação e configuração da pilha ELK, de forma a permitir o recebimento de logs dos EDRs utilizados.

5. Execução de testes e ataques: elaboração e execução de ameaças de segurança, buscando por alertas criados nos EDRs para alertar as ameaças.

4.2 Topologia

Para que fosse possível a implementação e configuração do projeto, foi de extrema importância não apenas decidir os sistemas operacionais e EDRs utilizados, mas também decidir a distribuição e equipamentos de redes utilizados, com o objetivo de garantir a comunicação principalmente entre os agentes, servidores EDR e a pilha ELK.

Como foi citado previamente, optou-se pela utilização do GNS3 como alicerce do projeto. Um detalhe importante a ser ressaltado é a escolha pela opção do GNS3 Server. Esta abordagem foi escolhida pois facilita o uso do software GNS3, não necessitando que as máquinas dos alunos possuam tanto recursos para manter a topologia inteira funcionando. Ela permite a conexão de mais de uma pessoa no mesmo projeto, ao mesmo tempo, de forma simples e gratuita, já que trata-se de um ambiente criado na UnB para uso dos alunos, diferentemente das alternativas de provedores de nuvem presentes no mercado.

A instalação e configuração da infraestrutura necessária para o download do GNS3 Server foi feita pela equipe de TI do Laboratório Latitude, dentro da própria UnB. A máquina física utilizada é um Servidor DELL Poweredge 750XS de 32 GB de RAM e SSD de 2 TB de armazenamento.

A conexão com a rede da UnB foi liberada via VPN pela equipe de TI do laboratório Latitude. Para os clientes, bastou-se ter as credenciais corretas criadas e realizar a instalação do cliente OpenVPN. Em seguida, foi necessário realizar a instalação do GNS3-all-in-one, que representa a parte cliente da conexão. Salienta-se que a versão escolhida para ser instalada é de extrema importância, uma vez que, para que haja a comunicação entre o GNS3 Server e o GNS3 Client, é necessário que ambos estejam operando na mesma versão. No caso em questão a versão utilizada foi a 2.2.32.

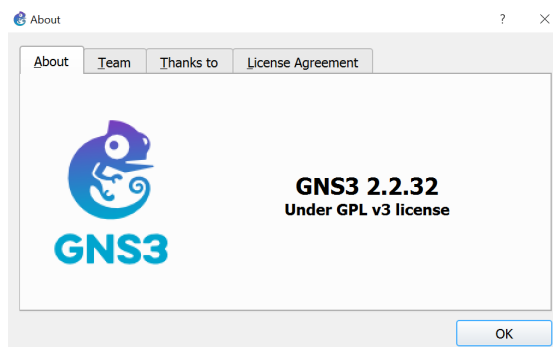


Figura 4.1 – Versão GNS3

Com o GNS3 funcionando, foram colocadas as máquinas virtuais a serem utilizadas. Em primeira instância foram colocadas 2 máquinas Linux com distribuição Ubuntu 16 LTS. A primeira dessas máquinas foi configurada para ser o Servidor OSSEC+ e a outra para ser o Agente OSSEC. Em seguida, foram acrescentadas 2 novas máquinas, também de sistema operacional Linux, mas

com distribuição CentOS 7. Nessas máquinas foram instalados o Servidor Wazuh e o Agente Wazuh. Colocou-se, então, uma nova máquina de sistema operacional Windows 10 *Education*. Esta versão e a chave foram extraídas do site da Azure por meio da conta educacional. Por fim, adicionou-se uma nova máquina com sistema operacional Linux, tendo a distribuição o CentOS 7. Esta nova máquina foi utilizada para a configuração da pilha ELK.

Para dar continuidade ao trabalho e transformá-lo em uma topologia mais legítima, foram colocados um roteador VyOS e também um Firewall pfSense. Com estes equipamentos, foi possível separar a rede dos agentes e a rede dos servidores, sendo usado para isso 2 sub-redes do tipo /24. E também garantir maior segurança entre a comunicação das redes internas para com a rede externa, que é representada no GNS3 pela imagem de uma nuvem branca com o nome de NAT. A topologia proposta pode ser vista a seguir, onde verifica-se a existência, como dito, da nuvem NAT, do roteador e das áreas dos clientes e dos servidores.

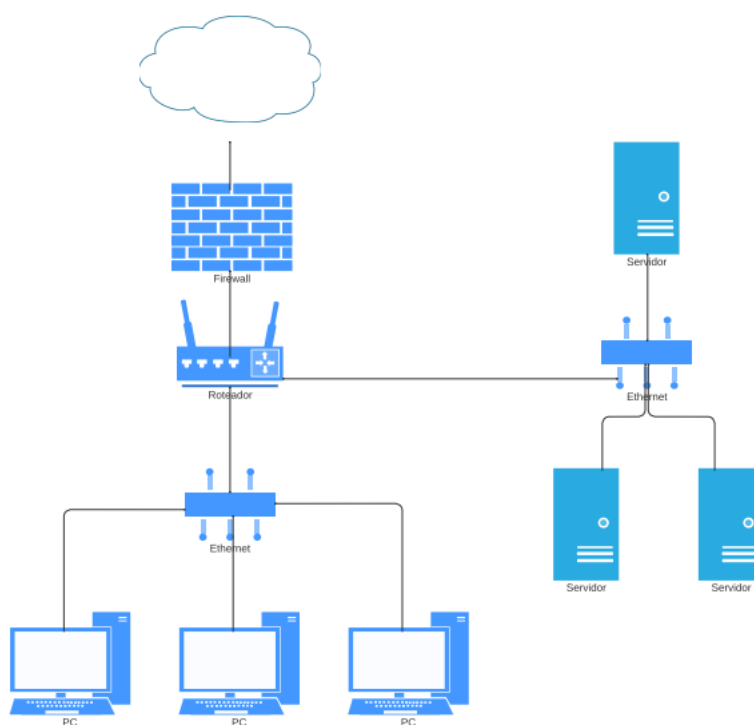


Figura 4.2 – Proposta de topologia

A rede utilizada pelos computadores agentes para comunicação com o roteador foi a 192.168.1.0/24. No roteador e nos switches foram feitas as configurações necessárias para o uso de VLANs. Já nos servidores, a rede empregada foi a 192.168.2.0/24. Assim como nas máquinas agentes, utilizou-se também a configuração de VLANs.

Para as soluções do OSSEC+ e do Wazuh, a configuração a ser utilizada é a Cliente-Servidor, isto é, um computador possui o agente do EDR, enquanto outro possui o servidor do EDR. Os agentes são responsáveis por monitorar as atividades realizadas nos hosts e enviar os logs para o servidor. Nos servidores é realizada o gerenciamento dos agentes monitorados. Por fim, para a utilização da pilha ELK no ambiente, foi feita a configuração, a partir do Filebeat, para que os

Tabela 4.1 – Configurações das máquinas utilizadas

Nome	IP	Sistema Operacional	vCPU	Memória
Elastic Server	192.168.40.10	CentOS 7	1	8 GB
OSSEC+ Server	192.168.10.10	Ubuntu 16.04	2	4 GB
Wazuh Server	192.168.20.10	CentOS 7	1	2 GB
OSSEC Agent	192.168.110.20	Ubuntu 16.04	2	4 GB
Wazuh Agent	192.168.120.20	CentOS 7	1	2 GB
OpenEDR Agent	192.168.130.20	Windows 10	1	2 GB

logs dos EDRs fossem passados para o servidor contendo a pilha ELK.

4.3 Configurações

A seguir serão explicitadas as configurações realizadas nos diversos dispositivos que compõem a rede. Uma máquina Kali Linux foi adicionada à uma interface do Firewall para a realização dos ataques posteriores. A topologia no GNS3 pode ser vista na Figura a seguir.

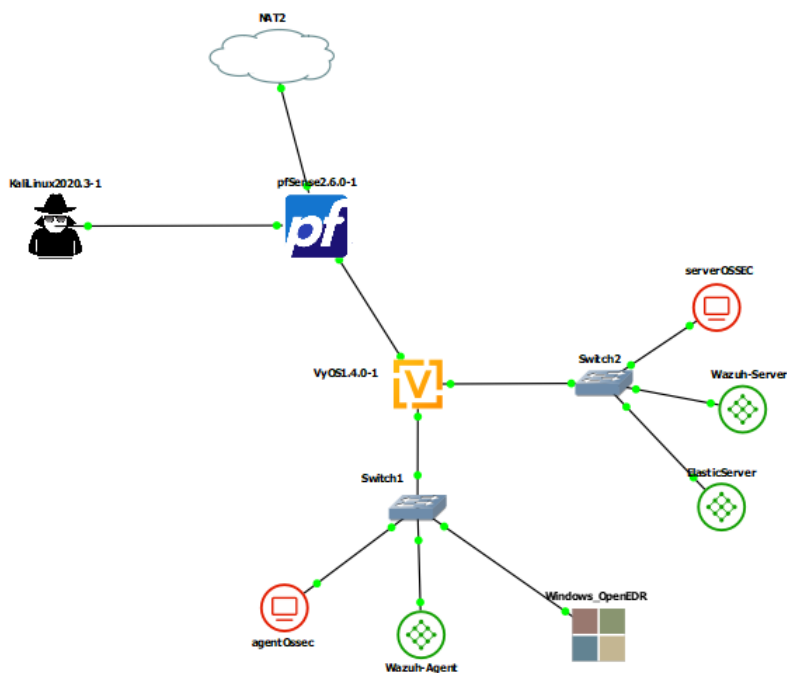


Figura 4.3 – Topologia da rede vista no GNS3

4.3.1 pfSense

Três interfaces do Firewall foram configuradas no projeto. A interface WAN é a conectada à NAT e recebe o seu endereço de IP via DHCP. A interface LAN conecta-se ao roteador VyOS e tem o IP 172.168.10.1/24. Nesta interface também foi configurado um servidor DHCP para fornecer o endereço de IP para a interface eth0 do roteador. A última interface a ser configurada é a OPT1,

a qual o atacante se conecta para realizar os ataques. O seu endereço de IP é 164.164.164.1/24 e a rede OPT1 é também uma rede externa.

```
*** Welcome to pfSense 2.6.0-RELEASE (amd64) on pfSense ***
WAN (wan)      -> em0      -> v4/DHCP4: 192.168.122.123/24
LAN (lan)      -> em1      -> v4: 172.168.10.1/24
OPT1 (opt1)    -> em5      -> v4: 164.164.164.1/24
```

Figura 4.4 – Endereçamento IP das interfaces do Firewall

Com relação às interfaces LAN e WAN foram mantidas as regras padrões do Firewall, já que inicialmente elas satisfazem os requisitos da topologia configurada. Todo o tráfego iniciado pela rede externa é barrado, enquanto que todo tráfego iniciado pela rede interna é liberado. Na interface OPT1 adicionou-se uma regra para bloquear todo tráfego vindo de fora.

Na seção seguinte, as mudanças realizadas nas regras do pfSense para possibilitar tanto a realização de ataques, quanto a posterior análise do desempenho das ferramentas de EDR frente às ameaças serão explicitadas.

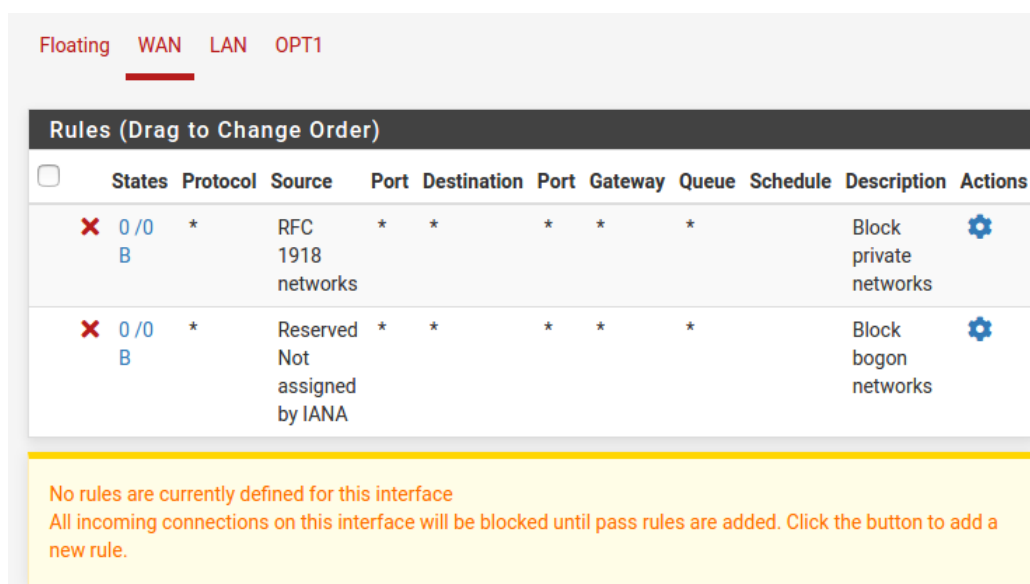


Figura 4.5 – Regras da interface WAN do Firewall

Floating WAN LAN OPT1

Rules (Drag to Change Order)

<input type="checkbox"/>	States	Protocol	Source	Port	Destination	Port	Gateway	Queue	Schedule	Description	Actions
<input checked="" type="checkbox"/>	3 / 47 KiB	*	*	*	LAN Address	80	*	*		Anti- Lockout Rule	
<input type="checkbox"/>	0 / 8 KiB	IPv4 *	LAN net	*	*	*	*	none		Default allow LAN to any rule	
<input type="checkbox"/>	0 / 0 B	IPv6 *	LAN net	*	*	*	*	none		Default allow LAN IPv6 to any rule	

Figura 4.6 – Regras da interface LAN do Firewall

Floating WAN LAN OPT1

Rules (Drag to Change Order)

<input type="checkbox"/>	States	Protocol	Source	Port	Destination	Port	Gateway	Queue	Schedule	Description	Actions
<input type="checkbox"/>	0 / 0 B	IPv4+6 *	OPT1 net	*	*	*	*	none			

Figura 4.7 – Regras da interface OPT1 do Firewall

4.3.2 Roteador VyOS

No roteador VyOS, configurou-se as VLANs nas interfaces eth1 e eth2. Uma VLAN foi criada para cada ferramenta utilizada no projeto, como demonstra a tabela a seguir.

Tabela 4.2 – VLANs da Topologia

Nome	IP	VLAN
OSSEC+ Server	192.168.10.10	10
OSSEC Agent	192.168.110.20	10
Wazuh Server	192.168.20.10	20
Wazuh Agent	192.168.120.20	20
OpenEDR Agent	192.168.130.20	30
Elastic Server	192.168.40.10	40

Sendo assim, foram configuradas três interfaces virtuais tanto na interface eth1, que conecta-se aos agentes, quanto na interface eth2, que conecta-se aos servidores. Essas configurações podem ser observadas a seguir.

Listing 4.1 – Configurações das interfaces do roteador VyOS

```
vyos@vyos# show
interfaces {
  ethernet eth0 {
    address dhcp
    hw-id 0c:50:7d:a6:00:00
  }
  ethernet eth1 {
    address 192.168.1.253/24
    hw-id 0c:50:7d:a6:00:01
    vif 10 {
      address 192.168.10.254/24
      description "VLAN 10 - OSSEC+"
    }
    vif 20 {
      address 192.168.20.254/24
      description "VLAN 20 - Wazuh"
    }
    vif 40 {
      address 192.168.40.254/24
      description "VLAN 40 - ELK"
    }
  }
  ethernet eth2 {
    address 192.168.2.254/24
    hw-id 0c:50:7d:a6:00:02
    vif 10 {
      address 192.168.110.254/24
      description "VLAN 10 - OSSEC+"
    }
    vif 20 {
      address 192.168.120.254/24
      description "VLAN 20 - Wazuh"
    }
    vif 30 {
      address 192.168.130.254/24
      description "VLAN 30 - OpenEDR"
    }
  }
}
```

A interface eth0 do roteador, que conecta-se ao pfSense, recebeu o endereço de IP 192.168.10.2/24 via DHCP, enquanto que as interfaces eth1 e eth2 receberam os endereços 192.168.1.253/24 e 192.168.2.254/24, respectivamente, de forma arbitrária.

Finalmente, foram configuradas as regras de NAT. Foi definido que a *outbound-interface* é a eth0, o endereço de origem é 192.168.0.0/16 e todas as requisições que saírem de dentro da rede interna sairão mascarados com o endereço de IP da interface eth0 do roteador.

Listing 4.2 – Configuração de NAT no roteador VyOS

```
nat {
  source {
    rule 100 {
      outbound-interface eth0
      source {
        address 192.168.0.0/16
      }
      translation {
        address masquerade
      }
    }
  }
}
```

4.3.3 Switches

Foram utilizadas as próprias appliances de switches do GNS3 para se comutar quadros para os dispositivos finais. As suas configurações foram referentes à questão das VLANs. As portas eth0 são aquelas que conectam os servidores e os agentes ao roteador VyOS. Portanto, foi definido o tipo *dot1q* nelas, a fim de permitir a passagem de pacotes com as *tags* das diversas VLANs. As outras interfaces tiveram o seu modo definido como *access* e a VLAN baseada na Tabela 4.2.

Tabela 4.3 – Interfaces do Switch que se conecta aos agentes

Interface	Modo	VLAN	Dispositivo Conectado
0	dot1q	1	Roteador VyOS
1	Access	10	Agente OSSEC
2	Access	20	Agente Wazuh
3	Access	30	Agente OpenEDR

Tabela 4.4 – Interfaces do Switch que se conecta aos servidores

Interface	Modo	VLAN	Dispositivo Conectado
0	dot1q	1	Roteador VyOS
1	Access	10	Servidor OSSEC+
2	Access	20	Servidor Wazuh
3	Access	40	Servidor Elastic

4.3.4 Pilha ELK

A instalação da pilha ELK foi realizada por partes. Na primeira parte, fez-se a instalação do serviço ElasticSearch, para isto, precisou-se apenas baixar o pacote, habilitá-lo e iniciá-lo. O próximo passo foi a instalação do Kibana, em que baixou-se o pacote e foram feitos ajustes nos arquivos de configurações para permitir a conexão de outras máquinas com o serviço do Kibana. Também foi feita a alteração da porta utilizada: optou-se pela porta 443 ao invés da 5601, sendo preciso realizar as configurações adicionais na própria máquina.

```
GNU nano 2.3.1 Arquivo: /etc/kibana/kibana.yml

# Set the interval in milliseconds to sample system and process performance
# metrics. Minimum is 100ms. Defaults to 5000.
#ops.interval: 5000

# Specifies locale to be used for all localizable strings, dates and number for$
# Supported languages are the following: English - en , by default , Chinese - $
#i18n.locale: "en"
server.host: "0.0.0.0"
server.port: 443
xpack.security.enabled: true
elasticsearch.username: "elastic"
elasticsearch.password: "██████████"
```

Figura 4.8 – Configuração do Kibana

Também, nesta máquina, foram realizadas as alterações necessárias para que o Kibana pudesse acessar a API disponibilizada pelo Wazuh-manager.

```
GNU nano | 2.3.1 Arquivo: ../etc/kibana/data/wazuh/config/wazuh.yml

# Values:
# - true: use his/her authentication context. Require Wazuh
# - false or not defined: get same permissions of Wazuh API
# run_as: <true|false>
hosts:
- wazuhapi:
  url: https://192.168.20.10
  port: 55000
  username: wazuh-wui
  password: wazuh-wui
```

Figura 4.9 – Configuração para conectar ao Wazuh-API

4.3.5 OSSEC+

Após a instalação tanto do agente OSSEC quanto do servidor OSSEC+, foi necessário o pareamento entre os dois. Para isso, definiu-se no servidor o nome e o endereço de IP do agente a ser conectado (192.168.110.20).

```
root@osboxes:/var/ossec/bin# ./manage_agents

*****
* OSSEC HIDS v3.6.0 Agent manager. *
* The following options are available: *
*****
(A)dd an agent (A).
(E)xtract key for an agent (E).
(L)ist already added agents (L).
(R)emove an agent (R).
(Q)uit.
Choose your action: A,E,L,R or Q: l

Available agents:
  ID: 005, Name: agentOssec, IP: 192.168.110.20

** Press ENTER to return to the main menu.
```

Figura 4.10 – Agente OSSEC disponível na interface de configuração do OSSEC+

Em seguida, extraiu-se uma chave no servidor e a inseriu no agente para realizar a associação

entre ambos.

```
root@osboxes:/var/ossec/etc# cat client.keys
001 #####122.227 8a59188963cd10d97a60c5987e3fe376c012a51f1b8e27e
0f4308a6553a6a636
002 #####122.227 df9b7694dd167f5c0ef22db20f3fa79626d72371cbbc83a
af14de3b4e9f7ee8f
003 #####22.227 088a9c0cb0924e3c01918f718b941b956cbd1a713e45e689
9e5900f5b3b4e285
004 #####2.13 cf399658dd9564e05f1bc89232253cacd3797d231de5b5cf87
0d9f05097158f3
005 agent0ssec 192.168.110.20 26678e3be395391d1abc59abe68945b3b248535a903aa04d44
a1fa299fae08ef
root@osboxes:/var/ossec/etc#
```

Figura 4.11 – Chaves de pareamento entre o servidor e os agentes

A configuração no Filebeat consistiu em adicionar-se o arquivo de log a ser transmitido e na informação do endereço de IP e porta, que serão utilizados para conectar ao servidor Elastic.

```
GNU nano 2.5.3 File: filebeat.yml
- type: log
  id: logs_ossec+
  # Change to true to enable this input configuration.
  enabled: true

# Paths that should be crawled and fetched. Glob based paths.
paths:
- /var/ossec/logs/alerts/2022/Aug/*
- /var/ossec/logs/alerts/2022/Sep/*
```

Figura 4.12 – Configuração OSSEC - Arquivos de log

```
GNU nano 2.5.3 File: filebeat.yml

# Configure what output to use when sending the data collected by the beat.
# ----- Elasticsearch Output -----
output.elasticsearch:
  # Array of hosts to connect to.
  hosts: ["192.168.40.10:9200"]

  # Protocol - either `http` (default) or `https`.
  #protocol: "https"

# Authentication credentials - either API key or username/password.
#api_key: "id:api_key"
username: "elastic"
password: "██████████"
```

Figura 4.13 – Configuração OSSEC - Conexão ELK

4.3.6 Wazuh

Foi realizada a instalação do Wazuh-manager de acordo com a documentação. O Wazuh-manager utiliza-se das portas 1514 e 1515 para a comunicação com outras máquinas, por isso foram feitas as configurações adicionais de forma a permitir conexões para com estas portas. Em seguida, foi realizada a instalação do Filebeat, que é responsável por repassar os logs adquiridos para a máquina contendo a pilha ELK. Um arquivo de base para a configuração do Filebeat

pode ser encontrado no próprio repositório do Wazuh. Utilizando-se deste arquivo, fizeram-se as alterações necessárias para o bom funcionamento da ferramenta.

```
GNU nano 2.3.1 Arquivo: filebeat.yml

# Wazuh - Filebeat configuration file
filebeat.modules:
- module: wazuh
  alerts:
    enabled: true
  archives:
    enabled: false

setup.template.json.enabled: true
setup.template.json.path: '/etc/filebeat/wazuh-template.json'
setup.template.json.name: 'wazuh'
setup.template.overwrite: true
setup.ilm.enabled: false

output.elasticsearch:
  hosts: ['http://192.168.40.10:9200']
  username: "elastic"
  password: "██████████"

setup.kibana.host: "http://192.168.40.10:443"
```

Figura 4.14 – Configuração do Filebeat no Wazuh Server

Com estes passos já realizados, fez-se, então, a instalação, em outra máquina, do Wazuh-agent, em que configurou-se o IP do Wazuh-manager para possibilitar a comunicação entre eles. Com isso, os serviços do Wazuh foram iniciados em cada máquina e pela máquina do Wazuh-agent foi possível verificar o estado de conexão entre eles.

```
[root@wazuh-agent-30 osboxes]# grep ^status /var/ossec/var/run/wazuh-agentd.state
status='connected'
```

Figura 4.15 – Comunicação estabelecida com sucesso entre os nós Wazuh Agent e Server

4.3.7 OpenEDR

A instalação do OpenEDR foi realizada na máquina com sistema operacional do tipo Windows, uma vez que este EDR somente possui compatibilidade com este sistema operacional [50]. O download da ferramenta foi feito através do próprio site da ferramenta, em seguida o arquivo foi descompactado e o EDR instalado.

Com esses passos realizados, precisou-se realizar as configurações para permitir a integração deste com a pilha ELK. Diferentemente dos outros EDRs escolhidos, este não possui a aplicação de servidor e deve ser configurado diretamente para a comunicação com a pilha ELK. Para isso, fez-se a instalação do Filebeat e a configuração do mesmo, alterando-se os IPs no arquivo de configuração para permitir a comunicação com o ElasticSearch e o Kibana.

```
# ----- Elasticsearch Output -----
output.elasticsearch:
  # Array of hosts to connect to.
  hosts: ["192.168.40.10:9200"]

  # Protocol - either `http` (default) or `https`.
  #protocol: "https"

  # Authentication credentials - either API key or username/password.
  #api_key: "id:api_key"
  username: "elastic"
  password: "██████████"
```

Figura 4.16 – Configuração do Filebeat - Comunicação ELK

Também foi necessário acrescentar uma configuração manual para que os logs gerados pelo OpenEDR pudessem ser enviados a pilha ELK.

```
# ===== Filebeat inputs =====

filebeat.inputs:

# Each - is an input. Most options can be set at the input level, so
# you can use different inputs for various configurations.
# Below are the input specific configurations.

- type: filestream

  # Change to true to enable this input configuration.
  enabled: true

  # Paths that should be crawled and fetched. Glob based paths.
  paths:
    # - /var/log/*.log
    - C:\ProgramData\edrsvc\log\output_events\*
```

Figura 4.17 – Configuração do Filebeat - Arquivos de Log

5 Teste e Análise

Antes de dar início aos testes foi necessário criar brechas na rede, a fim de permitir que o atacante execute os ataques selecionados. Primeiramente, liberou-se o tráfego em ambos os sentidos na interface OPT1, de modo com que o atacante consiga acessar a rede LAN do Firewall, como ilustra a Figura a seguir.

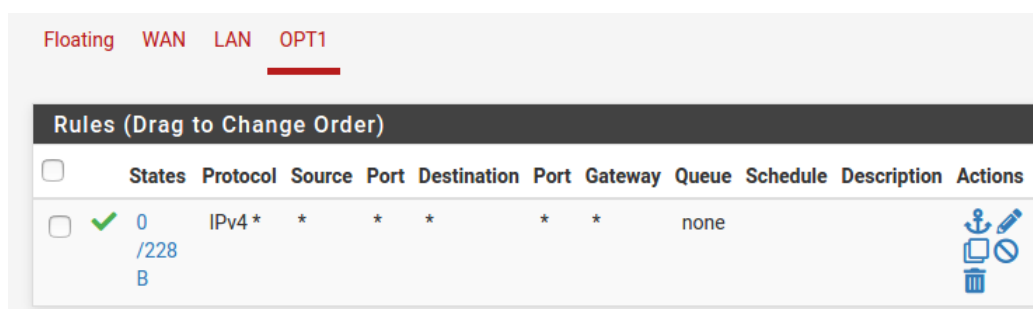


Figura 5.1 – Regra da interface OPT1 para a permitir a realização dos testes

Em seguida, criou-se uma regra de destino NAT no roteador VyOS para permitir que um dispositivo da rede externa acesse os agentes EDR configurados no projeto. Foi definido que qualquer pacote que tiver endereço de IP de destino 172.168.10.2 que chegue na interface eth0 do roteador VyOS deve ter o endereço traduzido para o IP do agente EDR escolhido e enviado ao mesmo. Quando o alvo dos ataques foi o agente OSSEC, o endereço de IP de tradução foi 192.168.110.20.

Listing 5.1 – Configuração NAT para ataque ao OSSEC Agent

```

nat {
  destination {
    rule 12 {
      destination {
        address 172.168.10.2
      }
      inbound-interface eth0
      translation {
        address 192.168.110.20
      }
    }
  }
}

```

Já quando o alvo dos ataques foi o agente Wazuh, o endereço de IP de tradução foi 192.168.120.20.

Listing 5.2 – Configuração NAT para ataque ao Wazuh Agent

```
nat {
  destination {
    rule 12 {
      destination {
        address 172.168.10.2
      }
      inbound-interface eth0
      translation {
        address 192.168.120.20
      }
    }
  }
}
```

Finalmente, quando o alvo dos ataques foi o agente OpenEDR, o endereço de IP de tradução foi 192.168.130.20.

Listing 5.3 – Configuração NAT para ataque ao OpenEDR Agent

```
nat {
  destination {
    rule 12 {
      destination {
        address 172.168.10.2
      }
      inbound-interface eth0
      translation {
        address 192.168.130.20
      }
    }
  }
}
```

Com relação à máquina Kali Linux que irá realizar os ataques, foram definidos os seguintes endereços de IP e gateway, respectivamente: 164.164.164.164/24 e 164.164.164.1.

```
kali@kali:~$ ifconfig
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    inet 164.164.164.164 netmask 255.255.255.0 broadcast 164.164.164.255
    inet6 fe80::be20:97e4:e936:504d prefixlen 64 scopeid 0<link>
    ether 0c:6f:72:2d:00:00 txqueuelen 1000 (Ethernet)
    RX packets 0 bytes 0 (0.0 B)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 16 bytes 1178 (1.1 KiB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING> mtu 65536
    inet 127.0.0.1 netmask 255.0.0.0
    inet6 ::1 prefixlen 128 scopeid 0<host>
    loop txqueuelen 1000 (Local Loopback)
    RX packets 12 bytes 640 (640.0 B)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 12 bytes 640 (640.0 B)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

kali@kali:~$ route -n
Kernel IP routing table
Destination      Gateway         Genmask         Flags Metric Ref    Use Iface
0.0.0.0          164.164.164.1 0.0.0.0         UG    100    0      0 eth0
164.164.164.0   0.0.0.0        255.255.255.0  U    100    0      0 eth0
kali@kali:~$
```

Figura 5.2 – Configuração de rede da máquina Kali Linux

5.1 Ataques

Cinco ataques foram selecionados para serem realizados: Scan NMAP, SSH Brute-Force, RDP Brute-Force, Kernel-Mode Rootkit e Ataque Shellshock. As máquinas que rodam em Linux não serão alvos do RDP Brute-Force, pois este ataque foge do escopo do sistema operacional. Da mesma forma, a máquina Windows não será alvo do ataque SSH Brute-Force. Ela também não será alvo da Exposição de Kernel-Mode Rootkit nem do ataque Shellshock, já que ambos ataques configuram ameaças a sistemas UNIX. Devido a problemas na instalação do módulo diamorphine no CentOS, este ataque também ficará de fora do escopo do Wazuh.

5.1.1 Scan NMAP

O nmap é uma ferramenta gratuita e de código livre que tem por finalidade auxiliar com a descoberta de redes e com auditorias de segurança. O Network Mapper, como também é conhecido, utiliza-se de pacotes do tipo IP para poder determinar os hosts disponíveis em certa rede, os serviços que estes podem estar ofertando, os sistemas operacionais, as portas que permitem conexão e até mesmo tipos de filtragem de pacotes que podem ocorrer, como é o caso dos Firewalls [42].

A função do nmap pode variar de acordo com quem o utiliza. Ele pode ser uma ferramenta aliada ao time de segurança, uma vez que permite verificar a aplicação de regras específicas de segurança, mas também pode servir de apoio a atacantes, possibilitando a descoberta de informações triviais para o reconhecimento da rede, etapa inicial da cyber kill chain.

O comando nmap foi realizado de uma máquina Kali Linux, que já possui o software instalado, de fora da rede. Para que o resultado fosse mais efetivo, optou-se por utilizar o parâmetro -A, que permite detectar o tipo do sistema operacional encontrado, além de escaneamento de scripts, e o uso do comando traceroute, que permite verificar os saltos feitos até o host.

5.1.1.1 Resultado OSSEC+

Como resultado para o comando nmap, sendo o OSSEC Agent o alvo, percebe-se a descoberta da ferramenta de um host ligado. Deste host, descobriu-se o tipo de sistema operacional utilizado, no caso um Linux, além da distribuição instalada, que é o Ubuntu Linux, a porta 22 configurada para conexões e o serviço utilizado por esta, que permite conexões do tipo SSH, Secure Shell, no host em questão.

```
kali@kali:~$ nmap -A 172.168.10.2
Starting Nmap 7.91 ( https://nmap.org ) at 2022-09-13 14:00 EDT
Nmap scan report for 172.168.10.2
Host is up (0.0033s latency).
Not shown: 999 closed ports
PORT      STATE SERVICE VERSION
22/tcp    open  ssh      OpenSSH 7.2p2 Ubuntu 4ubuntu2.10 (Ubuntu Linux; protocol 2.0)
| ssh-hostkey:
|   2048 9f:66:87:0d:d3:9b:bc:b1:f7:fb:14:97:b6:77:d5:6b (RSA)
|   256  75:1d:ec:9d:8a:5c:e3:77:6d:21:f9:f3:53:7d:16:ef (ECDSA)
|_  256  20:35:0a:b7:4f:93:64:f1:67:0d:d7:b3:14:1a:f2:72 (ED25519)
Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 13.84 seconds
kali@kali:~$
```

Figura 5.3 – Comando nmap com alvo OSSEC Agent

Em seguida, buscou-se na interface web, disponibilizada pela pilha ELK, alertas que tivessem sido gerados pelo agente OSSEC, presente na máquina alvo, em resposta ao scan realizado. Como pode ser visto a seguir, um alerta foi criado às 14:00:39.

```
> Sep 13, 2022 @ 14:00:39 @timestamp: Sep 13, 2022 @ 14:00:39.060 input.type: log ecs.version: 1.6.0 ho
st.os.version: 16.04.6 LTS (Xenial Xerus) host.os.family: debian host.os.name:
Ubuntu host.os.kernel: 4.15.0-142-generic host.os.codename: xenial host.os.pla
tform: ubuntu host.id: 71c1ba81e6a44556ab2aa78a0db23431 host.containerized: fal
se host.ip: 192.168.10.10, fe80::b0db:6118:d3fe:cb4a host.mac: 0c:fc:71:28:00:0
```

Figura 5.4 – Geração de alerta pelo OSSEC Agent - Resultado Scan Nmap

Ao expandir o alerta encontrado, pode-se visualizar a mensagem completa do alerta gerado. Nela, percebe-se a descoberta do agente de uma tentativa de comunicação SSH de forma insegura, podendo ser caracterizada como uma tentativa de escanear a porta. Nota-se, também, que o alerta possui o IP utilizado pela máquina atacante para realizar o comando.

```
message      ** Alert 1663092030.16387: - syslog,sshd,recon,
2022 Sep 13 14:00:30 (agentOssec) 192.168.110.20->/var/log/auth.l
og
Rule: 5706 (level 6) -> 'SSH insecure connection attempt (scan).'
```

Figura 5.5 – Mensagem de alerta OSSEC Agent - Resultado Scan Nmap

5.1.1.2 Resultado Wazuh

Quando o comando do nmap foi utilizado com alvo a máquina hospedando o Wazuh Agent, recebeu-se como resultado que a máquina utilizada se trata, provavelmente, de um sistema operacional Linux versão 4,3,5 ou 2.6, porém não foi descoberta a distribuição, no caso um Centos 7. Assim como no Ossec Agent, percebe-se que o scan encontrou a porta 22 aberta para comunicação SSH, como pode ser visto pela Figura 5.6.

```
root@kali:/home/kali# nmap -A 172.168.10.2
Starting Nmap 7.91 ( https://nmap.org ) at 2022-09-13 15:29 EDT
Nmap scan report for 172.168.10.2
Host is up (0.0024s latency).
Not shown: 999 filtered ports
PORT      STATE SERVICE VERSION
22/tcp    open  ssh      OpenSSH 7.4 (protocol 2.0)
ssh-hostkey:
|_ 2048 1a:cd:58:64:68:a6:e4:15:eb:bc:65:70:68:e6:2e:41 (RSA)
|_ 256  ed:f0:cc:37:7a:e9:5d:1f:bf:ea:c6:02:6d:32:32:7a (ECDSA)
|_ 256  64:65:e4:f3:9d:d2:d2:d2:7e:35:c0:cc:4a:29:3d:09 (ED25519)
Warning: OSScan results may be unreliable because we could not find at least 1 open and 1 closed port
Device type: general purpose
Running (JUST GUESSING): Linux 4.X|3.X|5.X|2.6.X (98%)
OS CPE: cpe:/o:linux:linux_kernel:4.4 cpe:/o:linux:linux_kernel:3 cpe:/o:linux:linux_kernel:5.1 cpe:/o:linux:linux_kernel:2.6.32
Aggressive OS guesses: Linux 4.4 (98%), Linux 4.0 (96%), Linux 3.10 - 4.11 (93%), Linux 3.11 - 4.1 (93%), Linux 3.2 - 4.9 (93%), Linux 5.1 (93%), Linux 2.6.32 (93%), Linux 2.6.32 or 3.10 (91%), Linux 2.6.32 - 2.6.35 (90%), Linux 2.6.32 - 2.6.39 (89%)
No exact OS matches for host (test conditions non-ideal).
Network Distance: 2 hops

TRACEROUTE (using port 22/tcp)
HOP RTT ADDRESS
1 1.12 ms 164.164.164.1
2 2.05 ms 172.168.10.2

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 9.95 seconds
```

Figura 5.6 – Comando nmap com alvo Wazuh Agent

Ao olhar os alertas gerados pelo Wazuh, que foram repassados para a pilha ELK, encontra-se a tentativa de uma conexão SSH em que não foram passadas as informações para a identificação. Mostra-se, ainda, o IP utilizado pelo atacante e a porta. Na parte de regras, nota-se que é levantada a possibilidade de um escaneamento de porta.

```

t _index          wazuh-alerts-4.x-2022.09.13
t agent.id       001
t agent.ip       192.168.120.20
t agent.name     wazuh-agent
t data.srcip     164.164.164.164
t data.srcport   42936
t decoder.name   sshd
t decoder.parent sshd
t full_log       Sep 13 15:29:37 wazuh-agent-30 sshd[7270]: Did not receive iden
tification string from 164.164.164.164 port 42936
t id            1663097378.21721
t input.type     log
t location       /var/log/secure
t manager.name   osboxes
t predecoder.hostname wazuh-agent-30
t predecoder.program_name sshd
t predecoder.timestamp Sep 13 15:29:37
t rule.description sshd: insecure connection attempt (scan).
# rule.firedtimes 1
t rule.gdpr      IV_35.7.d

```

Figura 5.7 – Alerta resultado nmap com alvo Wazuh Agent

Na continuação do alerta, pode-se ver que o Wazuh classificou o ataque pela regra presente no MITRE ATT&CK como movimento Lateral.

```

t rule.gpg13      4.12
t rule.groups     syslog, sshd, recon
t rule.id         5706
# rule.level      6
rule.mail         false
t rule.mitre.id   T1021.004
t rule.mitre.tactic Lateral Movement
t rule.mitre.technique SSH
t rule.nist_800_53 SI.4
t rule.pci_dss    11.4
t rule.tsc        CC6.1, CC6.8, CC7.2, CC7.3
timestamp        Sep 13, 2022 @ 15:29:38.086

```

Figura 5.8 – Mensagem resultado nmap com alvo Wazuh Agent

5.1.1.3 Resultado OpenEDR

Para que fosse possível a realização do ataque no host Windows, foi preciso criar uma brecha de segurança, permitindo que o scan fosse feito. Nisso, desligou-se o Firewall de proteção de Rede, disponibilizado pelo próprio sistema operacional.

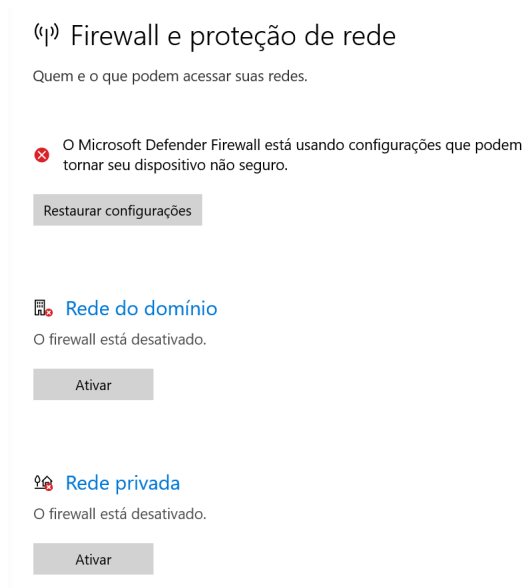


Figura 5.9 – Firewall do Windows desligado



Figura 5.10 – Firewall do Windows desligado - Rede Pública

Conseguindo-se realizar o ataque, percebe-se a descoberta de diversas portas abertas e dos serviços disponibilizados nestas portas. Chama-se a atenção para a porta 3389, que é utilizada no Windows para conectar-se remotamente a máquina, pelo uso do RDP, Remote Desktop Protocol. Apesar das informações obtidas, o nmap não conseguiu identificar o sistema operacional utilizado.

```

root@kali:/home/kali# nmap -A 172.168.10.2
Starting Nmap 7.91 ( https://nmap.org ) at 2022-09-13 17:50 EDT
Nmap scan report for 172.168.10.2
Host is up (0.0024s latency).
Not shown: 995 closed ports
PORT      STATE SERVICE          VERSION
135/tcp   open  msrpc            Microsoft Windows RPC
139/tcp   open  netbios-ssn     Microsoft Windows netbios-ssn
445/tcp   open  microsoft-ds?
3389/tcp  open  ms-wbt-server   Microsoft Terminal Services
|_ rdp-ntlm-info:
|   Target_Name: DESKTOP-JRHFTVJ
|   NetBIOS_Domain_Name: DESKTOP-JRHFTVJ
|   NetBIOS_Computer_Name: DESKTOP-JRHFTVJ
|   DNS_Domain_Name: DESKTOP-JRHFTVJ
|   DNS_Computer_Name: DESKTOP-JRHFTVJ
|   Product_Version: 10.0.19041
|_ System_Time: 2022-09-13T21:51:09+00:00
|_ ssl-cert: Subject: commonName=DESKTOP-JRHFTVJ
|   Not valid before: 2022-09-12T20:37:14
|_ Not valid after: 2023-03-14T20:37:14
|_ ssl-date: 2022-09-13T21:51:16+00:00; -8s from scanner time.
5357/tcp  open  http             Microsoft HTTPAPI httpd 2.0 (SSDP/UPnP)
|_ http-server-header: Microsoft-HTTPAPI/2.0
|_ http-title: Service Unavailable
No exact OS matches for host (If you know what OS is running on it, see https://nmap.org/submit/ ).
TCP/IP fingerprint:
OS:SCAN(V=7.91%E=4%D=9/13%OT=135%CT=1%CU=44081%PV=N%DS=2%DC=T%G=Y%TM=6320FB
OS:5C%P=x86_64-pc-linux-gnu)SEQ(SP=103%GCD=1%ISR=104%TI=I%TS=U)OPS(O1=M5B4N
OS:W8NNS%O2=M5B4NW8NNS%O3=M5B4NW8%O4=M5B4NW8NNS%O5=M5B4NW8NNS%O6=M5B4NNS)WI
OS:N(W1=FFFF%W2=FFFF%W3=FFFF%W4=FFFF%W5=FFFF%W6=FF70)ECN(R=Y%DF=Y%T=80%W=FF
OS:FF%O=M5B4NW8NNS%CC=N%Q=)T1(R=Y%DF=Y%T=80%S=0%A=S+%F=AS%RD=0%Q=)T2(R=N)T3
OS:(R=N)T4(R=N)T5(R=Y%DF=Y%T=80%W=0%S=Z%A=S+%F=AR%O=%RD=0%Q=)T6(R=N)T7(R=N)
OS:U1(R=Y%DF=N%T=80%IPL=164%UN=0%RIPL=G%RID=G%RIPCK=G%RUCK=G%RUD=G)IE(R=N)

```

Figura 5.11 – Comando nmap com alvo Windows - OpenEdr Agent

Procurando pelos logs gerados pela ferramenta OpenEdr, infelizmente nenhum alerta relacionado ao ataque foi encontrado.

5.1.2 SSH Brute-Force

O Brute-Force, ou Força-Bruta em português, retrata um método de ataque que busca realizar uma conexão com o host atacado, por meio da tentativa da combinação de várias senhas [9], até que se encontre uma efetiva. No caso do SSH, o ataque de força bruta tem como objetivo a tentativa de várias formas de login e senha para o uso do controle remoto do host pelo protocolo Secure Shell, um protocolo da camada de aplicação utilizado para conexão remota segura [62].

Para que fosse possível a realização do ataque, ou da tentativa dele, instalou-se nos hosts Linux, que estão com os agentes EDRs instalados, o OpenSSH. Trata-se de uma ferramenta livre e de código aberto que permite a utilização do protocolo SSH para realizar comunicações entre máquinas [51].

5.1.2.1 Resultado OSSEC+

A seguir é vista a tentativa do ataque explicitado a partir da máquina Kali Linux.


```
kali@kali:~$ ssh billy@172.168.10.2
billy@172.168.10.2's password:
Permission denied, please try again.
billy@172.168.10.2's password:
Permission denied, please try again.
billy@172.168.10.2's password:
billy@172.168.10.2: Permission denied (publickey,password).
kali@kali:~$
```

Figura 5.12 – SSH Brute-Force com alvo Linux Ubuntu 16.04 - OSSEC Agent

Foram emitidos dois alertas pelo agente OSSEC: um relativo à tentativa de login a partir de um usuário inválido, no caso, “billy” e outro relativo às múltiplas falhas na tentativa de se autenticar. Ambos foram visualizados na plataforma ELK. Nota-se também, o nível de criticidade dos alertas, sendo 5 o nível do primeiro e 10 o nível do segundo.

```
t message
** Alert 1663091269.13946: - syslog,sshd,invalid_login,authentication_failed,
2022 Sep 13 13:47:49 (agentOssec) 192.168.110.20->/var/log/auth.log
Rule: 5710 (level 5) -> 'Attempt to login using a non-existent user'
Src IP: 164.164.164.164
Sep 13 13:47:51 agentOssec sshd[3972]: Failed password for invalid user billy from 164.164.164.164 port 40724 ssh2
```

Figura 5.13 – Mensagem de alerta Agente OSSEC - Usuário Inválido

```
t message
** Alert 1663091287.15011: mail - syslog,access_control,authentication_failed,
2022 Sep 13 13:48:07 (agentOssec) 192.168.110.20->/var/log/auth.log
Rule: 2502 (level 10) -> 'User missed the password more than one time'
Sep 13 13:48:09 agentOssec sshd[3972]: PAM 2 more authentication failures; logname= uid=0 euid=0 tty=ssh ruser= rhost=164.164.164.164
```

Figura 5.14 – Mensagem de alerta Agente OSSEC - Múltiplas falhas na tentativa de autenticação

5.1.2.2 Resultado Wazuh

A execução do ataque à máquina do agente Wazuh é ilustrada na Figura a seguir.

```
root@kali:/home/kali# ssh billy@172.168.10.2
billy@172.168.10.2's password:
Permission denied, please try again.
billy@172.168.10.2's password:
Permission denied, please try again.
billy@172.168.10.2's password:
billy@172.168.10.2: Permission denied (publickey,gssapi-keyex,gssapi-with-mic,password).
root@kali:/home/kali#
```

Figura 5.15 – SSH Brute-Force com alvo Linux CentOS 7 - Wazuh Agent

O EDR emitiu um alerta indicando que o usuário errou a senha mais de uma vez e especificou o ataque baseado no *framework* Mitre. A técnica do ataque foi classificada como Brute Force sob a tática de *Credential Access*.

```

t _index          wazuh-alerts-4.x-2022.09.13
t agent.id        001
t agent.ip        192.168.120.20
t agent.name      wazuh-agent
t decoder.name    sshd
t full_log        Sep 13 15:22:30 wazuh-agent-30 sshd[7188]: PAM 2 more authentication failures; logname= uid=0 euid=0 tty=ssh ruser= rhost=164.164.4.164.164
t id              1663096951.21214
t input.type      log
t location        /var/log/secure
t manager.name    osboxes
t predecoder.hostname wazuh-agent-30
t predecoder.program_name sshd
t predecoder.timestamp Sep 13 15:22:30
t rule.description syslog: User missed the password more than one time
# rule.firedtimes 1
t rule.gdpr       IV_35.7.d, IV_32.2
t rule.gpg13      7.8
t rule.groups     syslog, access_control, authentication_failed

```

Figura 5.16 – Mensagem de alerta Agente Wazuh - Usuário errou a senha mais de uma vez

```

t rule.hipaa      164.312.b
t rule.id         2502
# rule.level      10
rule.mail         false
t rule.mitre.id   T1110
t rule.mitre.tactic Credential Access
t rule.mitre.technique Brute Force
t rule.nist_800_53 AU.14, AC.7
t rule.pci_dss    10.2.4, 10.2.5
t rule.tsc        CC6.1, CC6.8, CC7.2, CC7.3
timestamp        Sep 13, 2022 @ 15:22:31.692

```

Figura 5.17 – Mensagem de alerta Agente Wazuh - Técnica Brute Force sob a tática *Credential Access*

5.1.3 RDP Brute-Force

Muito similar ao SSH Brute-Force, o RDP Brute-Force consiste na tentativa de acesso a um host remoto a partir da combinação de várias senhas. A diferença é que o protocolo utilizado para

este ataque é o Remote Desktop Protocol, protocolo desenvolvido pela Microsoft, que fornece ao usuário uma interface gráfica para que ele tenha acesso a outro computador na rede.

Foi necessário habilitar o Remote Desktop na máquina Windows para permitir que outros computadores se conectem a ele e o controlem. No Windows, em Settings e Remote Desktop, habilitou-se essa funcionalidade.

Remote Desktop

Remote Desktop lets you connect to and control this PC from a remote device by using a Remote Desktop client (available for Windows, Android, iOS and macOS). You'll be able to work from another device as if you were working directly on this PC.

Enable Remote Desktop



Figura 5.18 – Ativação do Remote Desktop

5.1.3.1 Resultado OpenEDR

Para este ataque, removeu-se a máquina Kali Linux e inseriu-se um Windows 10 com as mesmas configurações de rede na interface OPT1 do Firewall para se realizar o ataque, como ilustra a Figura a seguir.

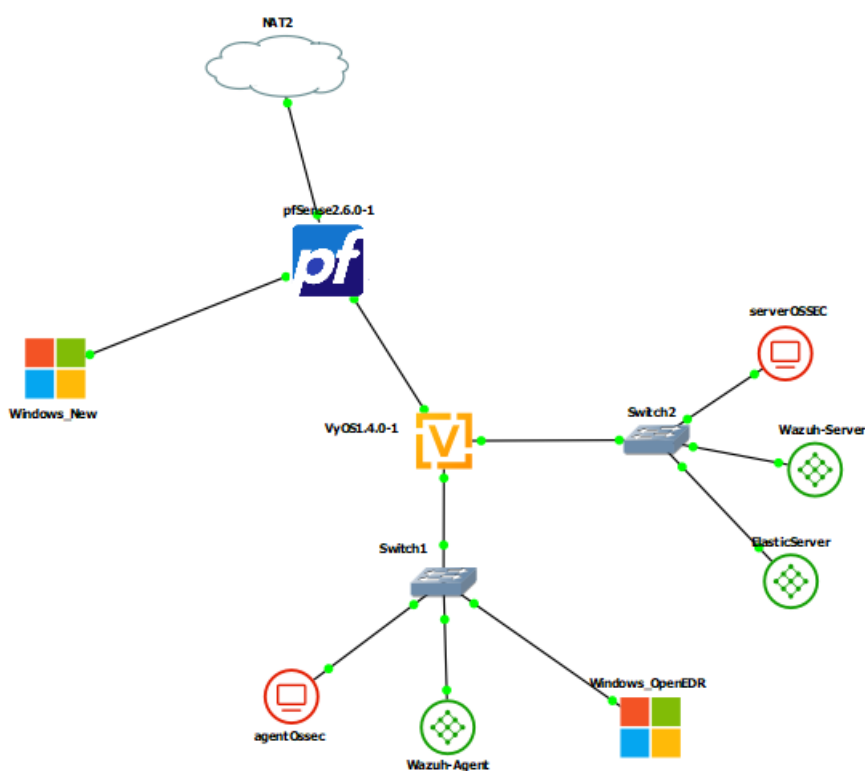


Figura 5.19 – Topologia com a máquina atacante Windows conectada ao Firewall

Na máquina do atacante, acessou-se o aplicativo Remote Desktop Connection e inseriu-se o endereço de IP da máquina destino 172.168.10.2.

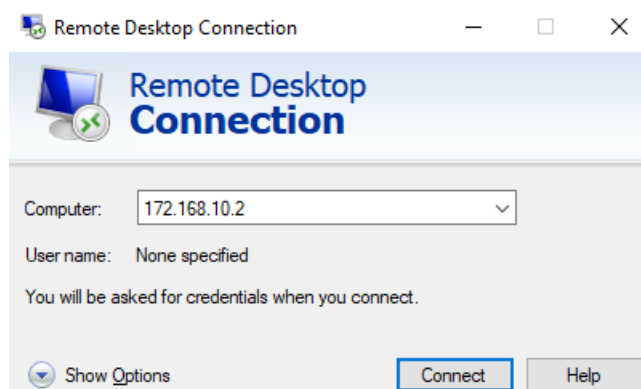


Figura 5.20 – Inserção do endereço de IP da máquina Windows a ser conectada

Em seguida, inseriu-se uma credencial errada para realizar a tentativa de acesso à máquina.

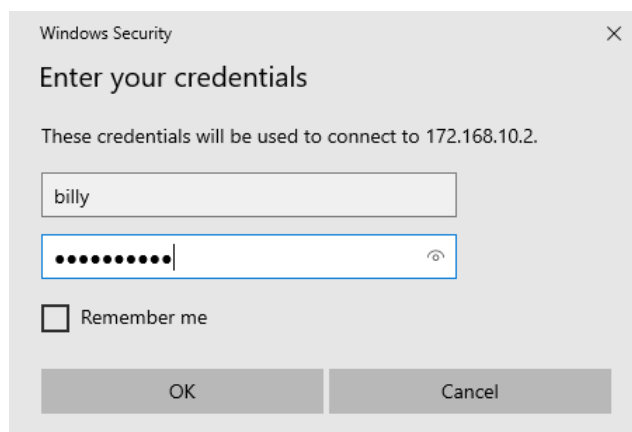


Figura 5.21 – Tentativa de login com credencial errada no alvo Windows 10

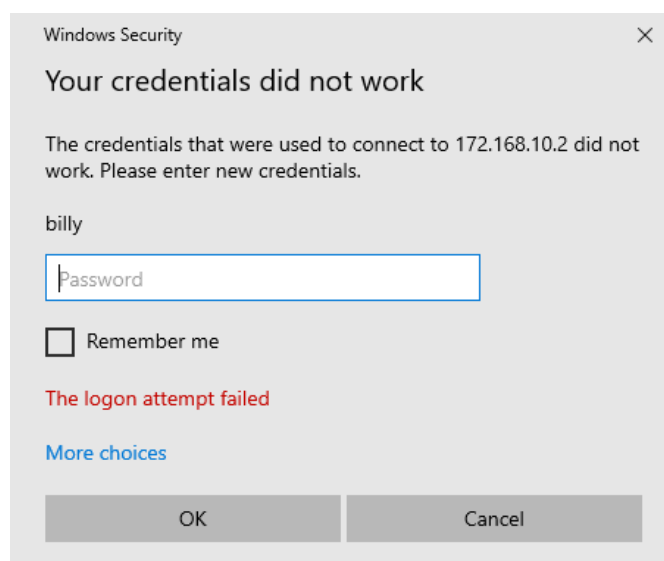


Figura 5.22 – Falha na tentativa de login com credencial errada no alvo Windows 10

O EDR não foi capaz de detectar a tentativa de ataque. No arquivo de log do OpenEDR buscou-se a palavra-chave “billy” (usuário da credencial usada pelo atacante) a fim de se verificar se a ferramenta gerou algum alerta relativo a esse incidente. No entanto, como ilustra a Figura a seguir, não ficou registrado nenhum evento relativo a essa tentativa de acesso.

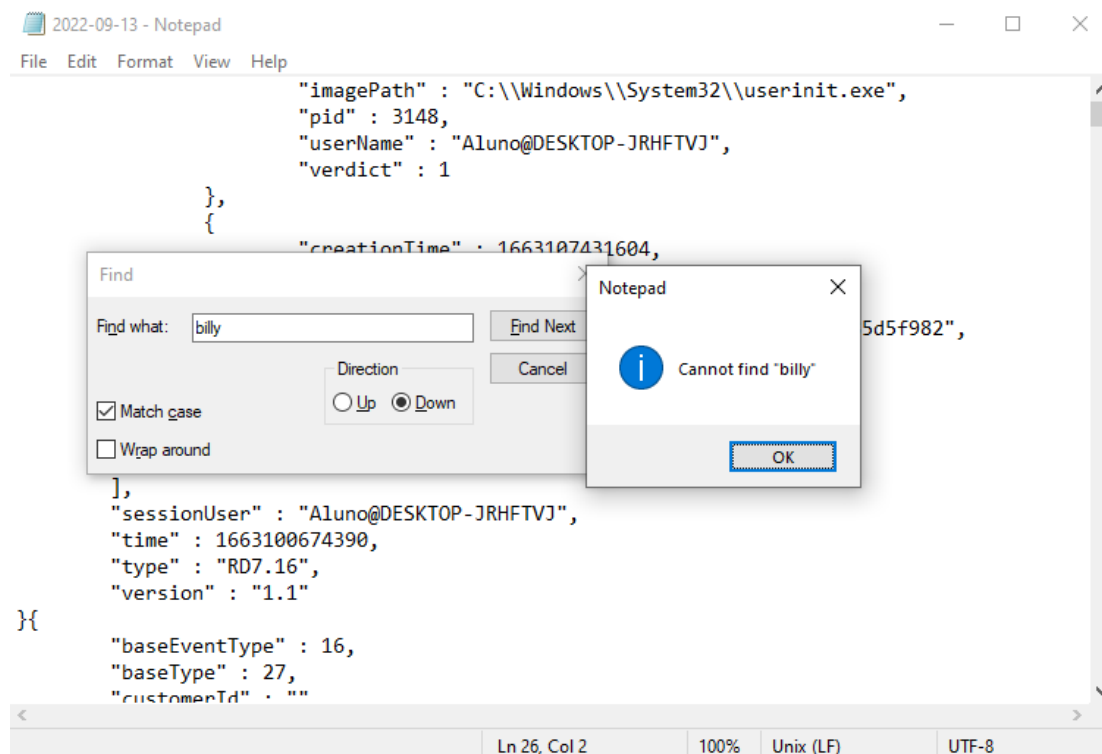


Figura 5.23 – Não foram gerados logs relativos ao ataque RDP Brute-Force pelo OpenEDR

5.1.4 Kernel-Mode Rootkit

Um *rootkit* é um programa, na maioria das vezes malicioso, empregado para mascarar a existência de processos que estão rodando na máquina do usuário [36]. Quando o *rootkit* é instalado, ele altera o funcionamento do sistema operacional e dos *softwares* de detecção de intrusão da máquina invadida, a fim de manter o ataque indetectado [36]. O *rootkit*, portanto, é um elemento da fase de Armamento da Cyber Kill Chain.

Quando o *rootkit* está rodando, não é possível visualizá-lo na lista de processos do módulo kernel, nem visualizar processos de interesse a este malware que seriam listados a partir do comando “ps” (*process status*) no terminal do Linux.

Para a realização deste ataque, considera-se que o atacante já obteve acesso à máquina alvo (via SSH, por exemplo) e deseja mascarar a execução de algum malware empregando um *rootkit*. Foi escolhido o *rootkit* Diamorphine para a realização dos testes. Ele foi desenvolvido para kernels Linux 2.6.x/3.x/4.x/5.x e ARM64 [19]. Quando carregado, o módulo é iniciado de forma invisível. Para tornar o módulo visível, envia-se o sinal 63 para qualquer pid (*process identifier*) [19]. Ao enviar o sinal 31, é possível esconder ou exibir dado processo que o recebe [19]. Ao se enviar o sinal 64 para qualquer pid, o usuário ganha privilégios administrativos (usuário root) [19]. Por

fim, ao se definir o parâmetro `MAGIC_PREFIX`, é possível esconder arquivos e diretórios que iniciem com o valor definido [19].

Seguiu-se os passos expostos em [19] para a instalação e carregamento do *rootkit*. Com o módulo carregado, enviou-se o sinal 63 para um pid qualquer para torná-lo visível. Logo após, enviou-se novamente o sinal 63 para um pid qualquer para torná-lo invisível. Esta sequência de ações pode ser vista na Figura a seguir, que mostra tanto a exibição quanto a ocultação do módulo.

```
root@agent0ssec:~/Diamorphine# kill -63 0
root@agent0ssec:~/Diamorphine# lsmod | grep diamorphine
diamorphine          16384  0
root@agent0ssec:~/Diamorphine# kill -63 0
root@agent0ssec:~/Diamorphine# lsmod | grep diamorphine
root@agent0ssec:~/Diamorphine#
```

Figura 5.24 – Ocultação e exibição do *diamorphine* a partir do envio do sinal 63 para um pid aleatório

5.1.4.1 Resultado OSSEC+

O OSSEC+ emitiu um alerta informando que um problema desconhecido no sistema foi detectado, atribuindo a ele o nível 2. A Figura a seguir ilustra a visualização da mensagem obtida a partir da interface web da pilha ELK.

```
t message          ** Alert 1663199300.23430: mail - syslog,errors,
                    2022 Sep 14 19:48:20 (agent0ssec) 192.168.110.20->/var/log/syslog
                    Rule: 1002 (level 2) -> 'Unknown problem somewhere in the system.
                    '
                    Sep 14 19:48:19 agent0ssec kernel: [ 2545.027884] diamorphine: mo
                    dule verification failed: signature and/or required key missing -
                    tainting kernel
```

Figura 5.25 – Mensagem de alerta agente OSSEC - Problema desconhecido no sistema

Ainda é possível visualizar na mesma Figura que houve falha na verificação do módulo *diamorphine*.

5.1.4.2 Resultado Wazuh

Infelizmente não foi possível realizar a tentativa deste ataque utilizando-se o CentOS 7 escolhido. Ao tentar rodar o comando “make”, o diretório “build” da pasta contendo o kernel não é reconhecido. Tentou-se mudar a versão do kernel, o que foi possível, porém não se obteve sucesso na realização da tentativa do ataque.

5.1.5 Ataque Shellshock

O Shellshock é um ataque realizado em sistemas do tipo Unix, uma vez que estes possuem uma vulnerabilidade presente no Bash, interpretador de comandos. Com o uso desta vulnerabilidade é possível realizar a execução de comandos de forma remota [21].

A vulnerabilidade foi descoberta e divulgada em 2014, estando associada as CVE-2014-6271, CVE-2014-6277, CVE-2014-6278, CVE-2014-7169, CVE-2014-7186 e CVE-2014-7187 [21]. A exploração da falha permite conseguir inserir código malicioso dentro de variáveis de ambiente,

podendo ser uma vulnerabilidade considerada do tipo Execução Arbitrária de Código, em que o atacante tem a capacidade de escolher e executar qualquer comando.

Esta vulnerabilidade abre portas para que um atacante, utilizando-se de linhas de comando, consiga enviar requisições HTTP que contenham código malicioso de seu interesse. Com isso, para este caso, foi realizada a instalação do software nginx, que permite a criação de um servidor web de forma simples, nas máquinas contendo os agentes EDRs.

Para tentar aprimorar a resposta dos EDRs com relação ao ataque a ser executado, alterou-se as configurações presentes no arquivo com caminho `/var/ossec/etc/ossec.conf`, adicionando-se o código necessário para permitir o acesso aos logs gerados pelo próprio nginx [7].

```
<localfile>
  <log_format>apache</log_format>
  <location>/var/log/nginx/access.log</location>
</localfile>

<localfile>
  <log_format>apache</log_format>
  <location>/var/log/nginx/error.log</location>
</localfile>
</ossec_config>
```

Figura 5.26 – Configuração adicional nos Agentes

Por fim, para realizar a tentativa do ataque, utilizou-se da máquina Kali Linux. Nesta, criou-se uma variável `ShellshockTarget`, de forma a representar o host a ser atacado. No cabeçalho do comando, a ser enviado, acrescentou-se “User Agent”, seguido pelos caracteres “() ;; ;”, que são colocados para interpretar o resultado do cabeçalho como um comando. No caso em questão, fez-se a tentativa de injetar no cabeçalho os comandos `/bin/cat /etc/passwd`, de forma a receber como resultado a visualização do conteúdo presente no arquivo de senhas do sistema operacional.

```
root@kali:/home/kali# ShellshockTarget="172.168.10.2"
root@kali:/home/kali# curl --insecure $ShellshockTarget -H "User-Agent: () { ;; }; /bin/cat /etc/passwd"
```

Figura 5.27 – Comando utilizado no ataque ShellShock

5.1.5.1 Resultado OSSEC+

Quando o comando foi realizado tendo como alvo o host com o OSSEC Agent instalado, obteve-se como resposta apenas o código HTML disponibilizado pelo nginx para a página inicial no Ubuntu Linux.

```

root@kali:/home/kali# ShellshockTarget="172.168.10.2"
root@kali:/home/kali# curl --insecure $ShellshockTarget -H "User-Agent: () { ;; }; /bin/cat /etc/passwd"
<!DOCTYPE html>
<html>
<head>
<title>Welcome to nginx!</title>
<style>
  body {
    width: 35em;
    margin: 0 auto;
    font-family: Tahoma, Verdana, Arial, sans-serif;
  }
</style>
</head>
<body>
<h1>Welcome to nginx!</h1>
<p>If you see this page, the nginx web server is successfully installed and
working. Further configuration is required.</p>

<p>For online documentation and support please refer to
<a href="http://nginx.org/">nginx.org</a>.<br/>
Commercial support is available at
<a href="http://nginx.com/">nginx.com</a>.</p>

<p><em>Thank you for using nginx.</em></p>
</body>
</html>

```

Figura 5.28 – Retorno em resposta ao comando realizado - Ossec

Com isso, percebe-se que o ataque não foi bem sucedido, uma vez que não foi possível visualizar as senhas salvas no arquivo `/etc/passwd`. Entretanto, a tentativa de ataque foi realizada e assim, coloca-se o sistema do host em risco. Apesar disso, não encontrou-se nenhum alarme criado pela ferramenta Ossec Agent que buscasse avisar a tentativa do ataque em questão.

5.1.5.2 Resultado Wazuh

Assim como no caso anterior, o retorno dado ao comando de ataque foi apenas o código HTML e CSS utilizado pelo nginx como página inicial para os sistemas CentOS, não sendo encontrada nenhuma senha.


```
root@kali:/home/kali# curl --insecure $ShellshockTarget -H "User-Agent: () { :; }; /bin/cat /etc/passwd"
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN">
<html>
<head>
  <title>Welcome to CentOS</title>
  <style rel="stylesheet" type="text/css">

    html {
      background-image:url(img/html-background.png);
      background-color: white;
      font-family: "DejaVu Sans", "Liberation Sans", sans-serif;
      font-size: 0.85em;
      line-height: 1.25em;
      margin: 0 4% 0 4%;
    }

    body {
      border: 10px solid #fff;
      margin:0;
      padding:0;
      background: #fff;
    }

    /* Links */

    a:link { border-bottom: 1px dotted #ccc; text-decoration: none; color: #204d92; }
    a:hover { border-bottom:1px dotted #ccc; text-decoration: underline; color: green; }
    a:active { border-bottom:1px dotted #ccc; text-decoration: underline; color: #204d92; }
    a:visited { border-bottom:1px dotted #ccc; text-decoration: none; color: #204d92; }
    a:visited:hover { border-bottom:1px dotted #ccc; text-decoration: underline; color: green; }

    .logo a:link,
    .logo a:hover,
    .logo a:visited { border-bottom: none; }
```

Figura 5.29 – Retorno em resposta ao comando realizado - Wazuh

Apesar disso, o Wazuh Agent foi capaz de reconhecer o comando utilizado, criando um alerta para os administradores. Nos detalhes do alarme gerado, percebe-se que a requisição HTTP realizada é do tipo GET, sendo respondida com sucesso, código 200. Mostra-se, ainda, no `full_log` o comando utilizado no ataque, e, a classificação do Wazuh com relação ao comando percebido, detectado como um ataque do tipo Shellshock.

† _index	wazuh-alerts-4.x-2022.09.13
† agent.id	001
† agent.ip	192.168.120.20
† agent.name	wazuh-agent
† data.id	200
† data.protocol	GET
† data.srcip	164.164.164.164
† data.url	/
† decoder.name	web-accesslog
† full_log	164.164.164.164 - - [13/Sep/2022:16:13:33 -0400] "GET / HTTP/1.1" 200 4833 "-" "() { ;; } /bin/cat /etc/passwd" "-"
† id	1663100014.45960
† input.type	log
† location	/var/log/nginx/access.log
† manager.name	osboxes
† rule.description	Shellshock attack detected
# rule.firedtimes	2
† rule.gdpr	IV_35.7.d
† rule.groups	web, accesslog, attack
† rule.id	31168

Figura 5.30 – Alarme gerado em resposta a tentativa de ataque

6 Análise e Comparação do Desempenho dos EDRs

6.1 Scan Nmap

Como foi possível acompanhar pela seção 5, os EDRs OSSEC+ e Wazuh demonstraram capacidade para alertar a tentativa de escaneamento dos hosts em busca de portas abertas e informações dos mesmos. Porém, é preciso levar em consideração a classificação que cada um forneceu ao ataque. Neste quesito, ambos os EDRs o classificaram como sendo uma possível tentativa de scan e atribuíram o nível 6 ao alerta (ataque de pequena relevância) [55] [56]. O Wazuh o classificou ainda como uma possível tentativa de movimento lateral. O OpenEDR, por outro lado, não foi capaz de alertar a tentativa de ataque de reconhecimento realizada no host.

6.2 SSH Brute-Force

Tanto o OSSEC+ quanto o Wazuh foram capazes de identificar o ataque. O OSSEC+ emitiu dois alertas: um de nível 5, relativo à inserção de um nome de usuário não existente, e um de nível 10, relativo às múltiplas falhas na tentativa de autenticação. Alertas de nível 5 não são muito relevantes e configuram erro gerado por usuário, o que é condizente com o incidente relatado [55]. Alertas de nível 10 já são bem mais significantes, já que são emitidos devidos a múltiplos erros gerados por usuário, podendo configurar um ataque [55]. O Wazuh emitiu apenas um alerta de nível 10, relativo às múltiplas falhas na tentativa de login. Ele ainda foi capaz de classificar o ataque segundo o *framework* Mitre, especificando corretamente a técnica do ataque como Brute-Force e a tática como *Credential Access*.

6.3 RDP Brute-Force

O OpenEDR foi o único alvo deste ataque e não se mostrou capaz de detectá-lo. Com relação ao protocolo RDP, os únicos registros gerados pelo EDR são relativos ao sucesso na autenticação, o que não é suficiente para a identificação do ataque. A Figura a seguir ilustra o log coletado na interface web da pilha ELK relativo à conexão realizada com sucesso a partir da credencial de nome de usuário “osboxes” cadastrada na máquina.

```

  Sep 13, 2022 @ 17:23:23.063 message: "userName" : "osboxes@DESKTOP-JRHFTVJ", @timestamp: Sep 13, 2022 @ 17
:23:23.063 log.offset: 43 log.file.path: C:\ProgramData\edrsvc\log\output_event
s\2022-09-13.log input.type: filestream ecs.version: 1.6.0 host.architecture:
x86_64 host.name: DESKTOP-JRHFTVJ host.os.name: Windows 10 Education N host.os
.kernel: 10.0.19041.1288 (WinBuild.160101.0800) host.os.build: 19044.1288 host

```

Figura 6.1 – Log relativo ao login de usuário via RDP no endpoint Windows

Como o Brute-Force consiste na inserção de múltiplas combinações de usuário e senha, muitos erros de credencial são gerados por ele consequentemente. Por não conseguir alertar esses eventos, o OpenEDR apresenta uma brecha a este tipo de ataque.

Procurou-se no aplicativo Event Viewer do Windows registros relativos ao ataque, a fim de se determinar se o próprio sistema operacional conseguiu identificar uma possível anormalidade. A única informação encontrada relativa ao ataque feito foi a de tentativa de conexão RDP recebida pelo endpoint. O Event Viewer não registrou logs relativos à falha na autenticação.

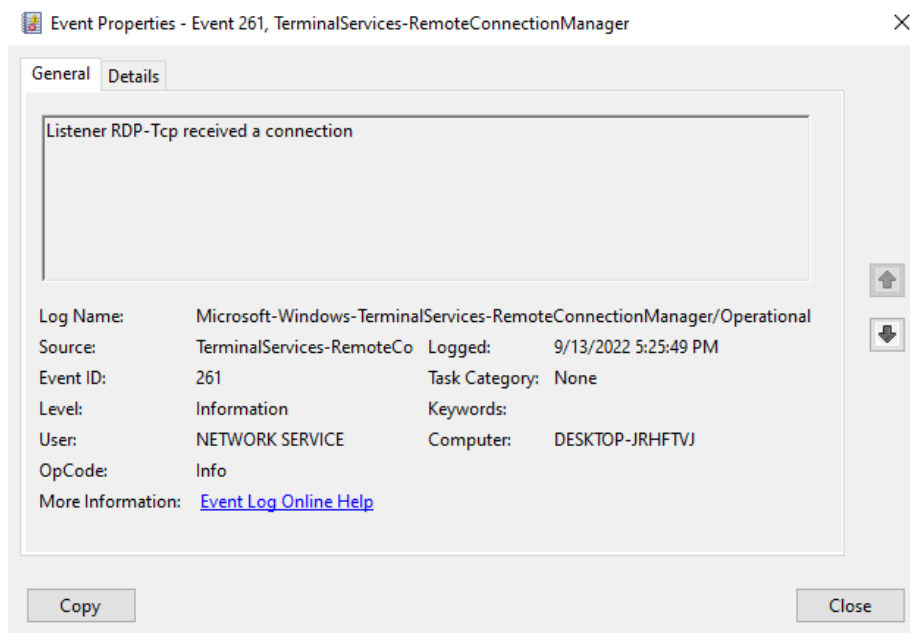


Figura 6.2 – Log obtido pelo Event Viewer

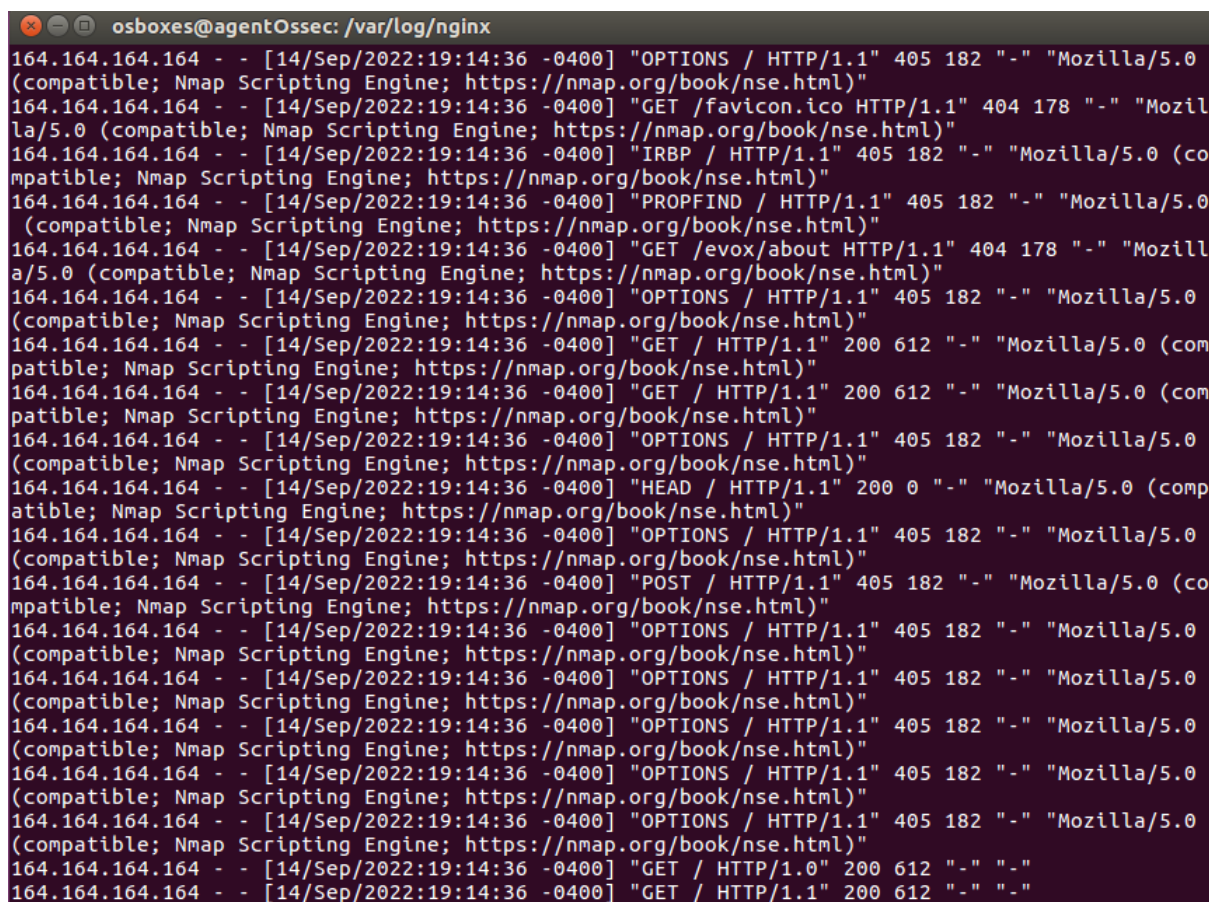
6.4 Kernel-Mode Rootkit

Neste ataque, o EDR OSSEC+ mostrou-se capaz de alarmar que uma atividade estranha estava acontecendo no host. Mesmo com o rootkit estando escondido, o EDR alertou que existia um problema e foi capaz de encontrar o seu nome, “diamorphine”, tratando-se de um problema relacionado ao kernel. Apesar disso, o EDR não foi capaz de informar mais detalhes, a fim de facilitar a análise e a resposta do time de segurança, nem de classificar o incidente corretamente com relação à sua criticidade. Foi atribuído o nível 2 ao alarme, que caracteriza notificações de baixa prioridade do sistema, de forma a não apresentar relevância com relação à segurança do dispositivo, o que não é condizente com a importância do incidente.

6.5 Ataque Shellshock

Em relação ao ataque Shellshock, o EDR OSSEC+ não foi capaz de alertar a tentativa de ataque realizada, apesar das configurações adicionais colocadas no arquivo do EDR. Analisou-se os arquivos de logs gerados pelo Nginx para verificar se a partir deles era possível extrair alguma informação útil para a identificação do incidente de segurança. O arquivo error.log estava vazio,

diferentemente do access.log, que registrou a conexão a partir do IP 164.164.164.164 e as respostas do endpoint.



```
osboxes@agentOssec: /var/log/nginx
164.164.164.164 - - [14/Sep/2022:19:14:36 -0400] "OPTIONS / HTTP/1.1" 405 182 "-" "Mozilla/5.0
(compatible; Nmap Scripting Engine; https://nmap.org/book/nse.html)"
164.164.164.164 - - [14/Sep/2022:19:14:36 -0400] "GET /favicon.ico HTTP/1.1" 404 178 "-" "Mozil
la/5.0 (compatible; Nmap Scripting Engine; https://nmap.org/book/nse.html)"
164.164.164.164 - - [14/Sep/2022:19:14:36 -0400] "IRBP / HTTP/1.1" 405 182 "-" "Mozilla/5.0 (co
mpatible; Nmap Scripting Engine; https://nmap.org/book/nse.html)"
164.164.164.164 - - [14/Sep/2022:19:14:36 -0400] "PROPFIND / HTTP/1.1" 405 182 "-" "Mozilla/5.0
(compatible; Nmap Scripting Engine; https://nmap.org/book/nse.html)"
164.164.164.164 - - [14/Sep/2022:19:14:36 -0400] "GET /evox/about HTTP/1.1" 404 178 "-" "Mozill
a/5.0 (compatible; Nmap Scripting Engine; https://nmap.org/book/nse.html)"
164.164.164.164 - - [14/Sep/2022:19:14:36 -0400] "OPTIONS / HTTP/1.1" 405 182 "-" "Mozilla/5.0
(compatible; Nmap Scripting Engine; https://nmap.org/book/nse.html)"
164.164.164.164 - - [14/Sep/2022:19:14:36 -0400] "GET / HTTP/1.1" 200 612 "-" "Mozilla/5.0 (com
patible; Nmap Scripting Engine; https://nmap.org/book/nse.html)"
164.164.164.164 - - [14/Sep/2022:19:14:36 -0400] "GET / HTTP/1.1" 200 612 "-" "Mozilla/5.0 (com
patible; Nmap Scripting Engine; https://nmap.org/book/nse.html)"
164.164.164.164 - - [14/Sep/2022:19:14:36 -0400] "OPTIONS / HTTP/1.1" 405 182 "-" "Mozilla/5.0
(compatible; Nmap Scripting Engine; https://nmap.org/book/nse.html)"
164.164.164.164 - - [14/Sep/2022:19:14:36 -0400] "HEAD / HTTP/1.1" 200 0 "-" "Mozilla/5.0 (comp
atible; Nmap Scripting Engine; https://nmap.org/book/nse.html)"
164.164.164.164 - - [14/Sep/2022:19:14:36 -0400] "OPTIONS / HTTP/1.1" 405 182 "-" "Mozilla/5.0
(compatible; Nmap Scripting Engine; https://nmap.org/book/nse.html)"
164.164.164.164 - - [14/Sep/2022:19:14:36 -0400] "POST / HTTP/1.1" 405 182 "-" "Mozilla/5.0 (co
mpatible; Nmap Scripting Engine; https://nmap.org/book/nse.html)"
164.164.164.164 - - [14/Sep/2022:19:14:36 -0400] "OPTIONS / HTTP/1.1" 405 182 "-" "Mozilla/5.0
(compatible; Nmap Scripting Engine; https://nmap.org/book/nse.html)"
164.164.164.164 - - [14/Sep/2022:19:14:36 -0400] "OPTIONS / HTTP/1.1" 405 182 "-" "Mozilla/5.0
(compatible; Nmap Scripting Engine; https://nmap.org/book/nse.html)"
164.164.164.164 - - [14/Sep/2022:19:14:36 -0400] "OPTIONS / HTTP/1.1" 405 182 "-" "Mozilla/5.0
(compatible; Nmap Scripting Engine; https://nmap.org/book/nse.html)"
164.164.164.164 - - [14/Sep/2022:19:14:36 -0400] "OPTIONS / HTTP/1.1" 405 182 "-" "Mozilla/5.0
(compatible; Nmap Scripting Engine; https://nmap.org/book/nse.html)"
164.164.164.164 - - [14/Sep/2022:19:14:36 -0400] "GET / HTTP/1.0" 200 612 "-" "-"
164.164.164.164 - - [14/Sep/2022:19:14:36 -0400] "GET / HTTP/1.1" 200 612 "-" "-"
```

Figura 6.3 – Registros do arquivo access.log do agente OSSEC

Por mais que se note muitos erros do cliente (HTTP Status 405), o EDR não foi capaz de alertar com relação a este incidente.

Enquanto isso, o Wazuh foi capaz de alertar e ainda colocou detalhes importantes, como a requisição feita, os comandos adicionais passados, o arquivo de log que forneceu estas informações e, ainda, a correta classificação deste como a detecção de um ataque Shellshock.

6.6 Comparativo Final

A seguir, realiza-se um comparativo entre todos os EDRs utilizados, identificando se o EDR foi capaz de alertar sobre a tentativa de ataque ou não. Na tabela, a representação do “S” significa que o EDR foi capaz, “N” significa que o EDR não foi capaz e “-” significa que o ataque em questão não foi realizado para o EDR.

Como pode ser visto pela tabela, cada EDR apresentou um resultado diferente com relação aos ataques utilizados. Percebe-se que, para os ataques escolhidos, o OpenEDR mostrou-se como o mais fraco entre os analisados, falhando em identificar as ameaças nas duas situações em que foi

Tabela 6.1 – Resultado dos EDRs frente aos ataques executados

Ataque	OSSEC+	Wazuh	OpenEDR
Nmap Scan	S	S	N
SSH Brute-Force	S	S	-
RDP Brute-Force	-	-	N
Kernel-Mode Rootkit	S	-	-
Shellshock	N	S	-

testado. O OSSEC+, apesar de ter alarmado com relação a três dos quatro ataques executados quando ele foi o alvo, não classificou o ataque Kernel-Mode Rootkit corretamente. Observa-se também que o OSSEC+ não foi capaz de gerar alarmes na tentativa de um ataque do gênero Shellshock, diferentemente do Wazuh. Este, em comparativo, teve o melhor desempenho durante os testes, sendo capaz não só de alarmar, mas também de identificar todos os ataques realizados corretamente.

7 Conclusão

Neste projeto, foi desenvolvida uma topologia de rede virtual voltada para análise de desempenho de três soluções de EDRs gratuitas e de código aberto disponíveis no mercado frente a ameaças conhecidas nos ecossistemas das redes de computadores. Com o posicionamento de um atacante na rede externa, foi possível a simulação de cinco ataques distintos realizados ao todo em três sistemas operacionais diferentes. Como os três EDRs foram integrados com a pilha ELK, conseguiu-se visualizar todos os logs gerados por esses agentes de maneira centralizada, facilitando o gerenciamento dos endpoints da rede e a análise dos incidentes de segurança. No final dos testes, foi possível avaliar o comportamento dos EDRs frente aos ataques executados, analisando se foram capazes de produzir respostas compatíveis com a criticidade da ameaça que estava sendo posta em prática.

7.1 Trabalhos Futuros

Como trabalhos futuros, sugere-se: a implementação e comparação dos EDRs Wazuh e OSSEC no modo *agentless*. Dispositivos de rede os quais não são passíveis de instalação de um agente EDR, como roteadores, Firewalls e switches, por exemplo, também podem ser monitorados por ambas as ferramentas. A conexão entre os dispositivos monitorados e o gerenciador é feita via autenticação SSH e é possível coletar logs de sistema, realizar a checagem de integridade de arquivos e até monitorar a execução de comandos em dispositivos remotos [6] [5].

Como alternativa, sugere-se a comparação do OpenEDR com outras soluções de EDR gratuitas e de código aberto voltadas exclusivamente para sistemas operacionais Windows, como o WHIDS. Ataques e vulnerabilidades específicas desse sistema operacional que não foram abordadas neste projeto podem ser exploradas, proporcionando análises tanto a ameaças quanto funcionalidades novas.

Bibliografia

- [1] *25 Crucial Information Technology Statistics Facts to Know*. URL: <<https://connect.comptia.org/blog/information-technology-stats-facts#:~:text=General%5C%20Technology%5C%20Statistics&text=Zippia%5C%20reports%5C%20that%5C%20the%5C%20tech,are%5C%20collectively%5C%20worth%5C%20%5C%244%5C%20trillion>> (acesso em 22/08/2022).
- [2] *A pilha ELK*. URL: <<https://aws.amazon.com/pt/opensearch-service/the-elk-stack/>> (acesso em 04/09/2022).
- [3] *advanced persistent threat (APT)*. URL: <https://csrc.nist.gov/glossary/term/advanced_persistent_threat> (acesso em 29/08/2022).
- [4] *Agentes de dados lightweight*. URL: <<https://www.elastic.co/pt/beats/>> (acesso em 04/09/2022).
- [5] *Agentless monitoring*. URL: <<https://www.ossec.net/docs/manual/agent/agentless-monitoring.html#getting-started-with-agentless>> (acesso em 16/09/2022).
- [6] *Agentless monitoring - How it works*. URL: <<https://documentation.wazuh.com/current/user-manual/capabilities/agentless-monitoring/how-it-works.html>> (acesso em 16/09/2022).
- [7] A.P.A. ALARCÃO. *IMPLEMENTAÇÃO E ANÁLISE DE RESULTADOS DE FERRAMENTA DE DETECÇÃO E RESPOSTA PARA PROTEÇÃO DE ENDPOINTS EM AMBIENTE CONTROLADO*. pt-BR. Monografia de Projeto Final de Graduação. Brasília, DF, Brasil, 2021.
- [8] Ricardo Ávila et al. “Use of Security Logs for Data Leak Detection: A Systematic Literature Review”. Em: *Security and Communication Networks 2021* (2020), pp. 1–29. DOI: <<https://doi.org/10.1155/2021/6615899>>.
- [9] *Brute Force Password Attack*. URL: <https://csrc.nist.gov/glossary/term/brute_force_password_attack> (acesso em 14/09/2022).
- [10] *Centralize, transforme e oculte seus dados*. URL: <<https://www.elastic.co/pt/logstash/>> (acesso em 04/09/2022).
- [11] Ping Chen, Lieven Desmet e Christophe Huygens. “A Study on Advanced Persistent Threats”. Em: *Communications and Multimedia Security* (2014), pp. 63–72. DOI: <LNCS8735>.
- [12] *Como Surgiu a Internet?* URL: <<https://csrc.nist.gov/glossary/term/risk>> (acesso em 04/09/2022).
- [13] *Complete your OSSEC+ Install*. URL: <<https://www.ossec.net/finish-ossec-plus-install/>> (acesso em 31/08/2022).
- [14] *Conheça a história da Internet, sua finalidade e qual o cenário atual*. URL: <[Conhe%C3%A7a%20a%20hist%C3%B3ria%20da%20Internet,%20sua%20finalidade%20e%20qual%20o%20cen%C3%A1rio%20atual](https://www.conheca.com.br/historia-da-internet-sua-finalidade-e-qual-o-cenario-atual)> (acesso em 02/09/2022).

- [15] *Conheça os princípios da segurança da informação*. URL: <<https://dpinet.com.br/blog-principios-da-seguranca-da-informacao/>> (acesso em 25/08/2022).
- [16] *Critical Capabilities for Security Information and Event Management*. URL: <<https://www.gartner.com/doc/reprints?id=1-26S3MR9Q&ct=210714&st=sb>> (acesso em 25/08/2022).
- [17] *Cyber Threat*. URL: <https://csrc.nist.gov/glossary/term/cyber_threat> (acesso em 30/08/2022).
- [18] *Dados demais também são um perigo*. URL: <<https://www.kaspersky.com.br/blog/hash-o-que-sao-e-como-funcionam/2773/d>> (acesso em 24/08/2022).
- [19] *Diamorphine*. URL: <<https://github.com/m0nad/Diamorphine>> (acesso em 14/09/2022).
- [20] *Entenda o que são vulnerabilidades, as mais recorrentes e os mecanismos de segurança da informação para evitá-las*. URL: <<https://tripla.com.br/entenda-o-que-sao-vulnerabilidades/>> (acesso em 04/09/2022).
- [21] *EXPLORANDO O BASH COM O ATAQUE SHELLSHOCK*. URL: <<https://kryptus.com/explorando-o-bash-com-o-ataque-shellshock/>> (acesso em 15/09/2022).
- [22] *Firewall*. URL: <<https://docs.netgate.com/pfsense/en/latest/firewall/index.html>> (acesso em 04/09/2022).
- [23] World Economic Forum. *The Global Risks Report 2022*. World Economic Forum, 2022.
- [24] *Get OSSEC*. URL: <<https://www.ossec.net/ossec-downloads/>> (acesso em 31/08/2022).
- [25] *Getting Started with GNS3*. URL: <<https://docs.gns3.com/docs/>> (acesso em 03/09/2022).
- [26] *Hacker*. URL: <<https://news.softpedia.com/news/hacker-has-job-offer-following-the-610m-crypto-heist-533817.shtml>> (acesso em 10/09/2022).
- [27] *História da Internet*. URL: <<https://www.todamateria.com.br/historia-da-internet/>> (acesso em 02/09/2022).
- [28] *How COVID-19 has pushed companies over the technology tipping point—and transformed business forever*. URL: <<https://www.mckinsey.com/business-functions/strategy-and-corporate-finance/our-insights/how-covid-19-has-pushed-companies-over-the-technology-tipping-point-and-transformed-business-forever>> (acesso em 22/08/2022).
- [29] *IBM Security QRadar SIEM*. URL: <<https://www.ibm.com/br-pt/qradar/security-qradar-siem#:~:text=que%5C%20%5C%3%5C%A9%5C%20SIEM%5C%3F-,SIEM%5C%20%5C%3%5C%A9%5C%20uma%5C%20solu%5C%3%5C%A7%5C%3%5C%A3o%5C%20de%5C%20seguran%5C%3%5C%A7a%5C%20que%5C%20ajuda%5C%20as%5C%20organiza%5C%3%5C%A7%5C%3%5C%B5es,interromper%5C%20as%5C%20opera%5C%3%5C%A7%5C%3%5C%B5es%5C%20de%5C%20neg%5C%3%5C%B3cios.>> (acesso em 30/08/2022).
- [30] *Introducing Open Source EDR*. URL: <<https://openedr.com/>> (acesso em 30/08/2022).
- [31] *Introduction*. URL: <<https://docs.netgate.com/pfsense/en/latest/general/index.html>> (acesso em 04/09/2022).

- [32] Jens Gustedt, Emmanuel Jeannot, Martin Quinson, Inria Nancy. “Experimental Validation in Large-Scale Systems: a Survey of Methodologies”. Em: (2009), pp. 1–17. DOI: <10.1142/S01296264090003044>.
- [33] George Karantzas e Constatinos Patsakis. “An Empirical Assessment of Endpoint Detection and Response Systems against Advanced Persistent Threats Attack Vectors”. Em: *Journal of Cybersecurity and Privacy* (2021), pp. 387–421. DOI: <https://www.mdpi.com/2624-800X/1/3/21>.
- [34] Hyeob Kim, HyukJun Kwon e Kyung Kyu Kim. “Modified cyber kill chain model for multimedia service environments”. Em: *Multimedia Tools and Applications* (2018), pp. 3153–3170. DOI: <https://doi.org/10.1007/s11042-018-5897-5>.
- [35] Sujeong Kim, Chanwoong Hwang e Taejin Lee. “Anomaly Based Unknown Intrusion Detection in Endpoint Environments”. Em: *Electronics* (2020). DOI: <10.3390/electronics9061022>.
- [36] Kyungroul Lee et al. “A Brief Survey on Rootkit Techniques in Malicious Codes”. Em: *Journal of Internet Services and Information Security* 3 (2012), pp. 134–147. DOI: <10.22667/JISIS.2012.11.31.134>.
- [37] Terry Liggett. *EVOLUTION OF ENDPOINT DETECTION AND RESPONSE PLATFORMS*. en-US. Dissertação. Utica, NY, United States, 2018.
- [38] *Log*. URL: <https://dictionary.cambridge.org/dictionary/english/log> (acesso em 24/08/2022).
- [39] CÉSAR MALERBA. *Vulnerabilidades e Exploits: técnicas, detecção e prevenção*. pt-BR. Monografia de Projeto Final de Graduação. Porto Alegre, Brasil, 2010.
- [40] *Mecanismos da Segurança: como funcionam e qual a importância?* URL: <https://blog.starti.com.br/mecanismos-da-seguranca/> (acesso em 10/09/2022).
- [41] NIST. *Guide to Computer Security Log Management*. Set. de 2006. URL: <https://nvlpubs.nist.gov/nistpubs/Legacy/SP/nistspecialpublication800-92.pdf>.
- [42] *Nmap*. URL: <https://nmap.org/> (acesso em 10/09/2022).
- [43] *O que é o Elasticsearch?* URL: <https://www.elastic.co/pt/what-is/elasticsearch> (acesso em 04/09/2022).
- [44] *O que é o ELK Stack?* URL: <https://www.elastic.co/pt/what-is/elk-stack> (acesso em 10/09/2022).
- [45] *O que é o Kibana?* URL: <https://www.elastic.co/pt/what-is/kibana> (acesso em 04/09/2022).
- [46] *O que é SIEM e quais suas principais funcionalidades?* URL: <https://seginfo.com.br/2020/09/03/o-que-e-siem-e-quais-suas-principais-funcionalidades/> (acesso em 08/09/2022).
- [47] *O que é SIEM e quais suas principais funcionalidades?* URL: <https://www.gcsec.com.br/o-que-e-siem-e-quais-suas-principais-funcionalidades/index.html#:~:text=SIEM%5C%20%5C%C3%5C%A9%5C%20a%5C%20combina%5C%C3%5C%A7%5C%C3%5C%A3o%5C%20de,SIM%5C%20%5C%E2%5C%80%5C%93%5C%20security%5C%20information%5C%20management).> (acesso em 25/08/2022).

- [48] *O que é uma ameaça em segurança da informação? Como calcular o seu impacto?* URL: <<https://www.siteware.com.br/seguranca/o-que-e-uma-ameaca-em-seguranca-da-informacao/#:~:text=Podemos%5C%20definir%5C%20o%5C%20que%5C%20%5C%A9,e%5C%20informa%5C%20%5C%A7%5C%20sobre%5C%20a%5C%20empresa.>> (acesso em 02/09/2022).
- [49] *OpenEDR*. URL: <<https://github.com/ComodoSecurity/openedr?key5sk1=78f940f9ceb5623a325033d62bdaf=7639>> (acesso em 30/08/2022).
- [50] *OpenEDR*. URL: <<https://github.com/ComodoSecurity/openedr?key5sk1=4406edb5ceba3f4a6ceb8890942>> (acesso em 02/09/2022).
- [51] *OpenSSH: Como Utilizar Para Criar Um Servidor SSH No Linux Com Diversas Camadas De Segurança*. URL: <<https://e-tinet.com/linux/openssh/>> (acesso em 14/09/2022).
- [52] *OSSEC Architecture*. URL: <<https://www.ossec.net/docs/docs/manual/ossec-architecture.html>> (acesso em 31/08/2022).
- [53] *pfSense - introduction to the most powerfull router operating system*. URL: <<https://teklager.se/en/pfsense-introduction-open-source-router-firewall/>> (acesso em 04/09/2022).
- [54] *Qual a história e o futuro da segurança de rede?* URL: <<https://www.avast.com/pt-br/business/resources/future-of-network-security#pc>> (acesso em 23/08/2022).
- [55] *Rules Classification*. URL: <<https://www.ossec.net/docs/manual/rules-decoders/rule-levels.html>> (acesso em 16/09/2022).
- [56] *Rules Classification*. URL: <<https://documentation.wazuh.com/current/user-manual/ruleset/rules-classification.html>> (acesso em 16/09/2022).
- [57] *SIEM*. URL: <https://www.precisely.com/app/uploads/2019/11/SIEM-Graphic_2020-800x432.png> (acesso em 10/09/2022).
- [58] *Supported Systems*. URL: <<https://www.ossec.net/docs/docs/manual/supported-systems.html>> (acesso em 31/08/2022).
- [59] *Take A Tour Getting Started*. URL: <<https://www.pfsense.org/getting-started/>> (acesso em 04/09/2022).
- [60] *Take A Tour of pfSense*. URL: <<https://www.pfsense.org/about-pfsense/>> (acesso em 04/09/2022).
- [61] *The FreeBSD Project*. URL: <<https://www.freebsd.org/>> (acesso em 04/09/2022).
- [62] *The Secure Shell (SSH) Transport Layer Protocol*. URL: <<https://datatracker.ietf.org/doc/html/rfc4253>> (acesso em 14/09/2022).
- [63] *UK Online Shopping and E-Commerce Statistics for 2017*. URL: <<https://www.nasdaq.com/articles/uk-online-shopping-and-e-commerce-statistics-2017-2017-03-14>> (acesso em 22/08/2022).
- [64] Wajih Ul Hassan, Adam Bates e Daniel Marino. “Tactical Provenance Analysis for Endpoint Detection and Response Systems”. Em: *2020 IEEE Symposium on Security and Privacy* (2020). DOI: <10.1109/SP40000.2020.00096>.

- [65] *vulnerability*. URL: <<https://csrc.nist.gov/glossary/term/vulnerability>> (acesso em 04/09/2022).
- [66] VyOS maintainers and contributors. *VyOS Documentation - Release 1.4.x (sagitta)*. Set. de 2022. URL: <<https://buildmedia.readthedocs.org/media/pdf/vyos/latest/vyos.pdf>>.
- [67] *Wazuh Architecture*. URL: <<https://documentation.wazuh.com/current/getting-started/architecture.html>> (acesso em 03/09/2022).
- [68] *Wazuh Quickstart*. URL: <<https://documentation.wazuh.com/current/quickstart.html>> (acesso em 03/09/2022).
- [69] *Wazuh Use Cases*. URL: <<https://documentation.wazuh.com/current/getting-started/use-cases/index.html>> (acesso em 03/09/2022).
- [70] *What does pfSense stand for/mean?* URL: <<https://docs.netgate.com/pfsense/en/latest/general/what-is-pfsense.html>> (acesso em 04/09/2022).
- [71] *What Is a Firewall?* URL: <<https://www.cisco.com/c/en/us/products/security/firewalls/what-is-a-firewall.html#~types-of-firewalls>> (acesso em 04/09/2022).
- [72] *What is FreeBSD?* URL: <<https://freebsdoundation.org/freebsd-project/what-is-freebsd/>> (acesso em 04/09/2022).
- [73] *What is Snort?* URL: <<https://www.snort.org/>> (acesso em 04/09/2022).
- [74] *Wild Wide Web*. URL: <<https://reports.weforum.org/global-risks-report-2020/wild-wide-web/>> (acesso em 22/08/2022).
- [75] Tarun Yadav e Arvind Mallari Rao. “Technical Aspects of Cyber Kill Chain”. Em: (2015), pp. 438–452. DOI: <10.1007/978-3-319-22915-7_40>.
- [76] Yanping Zhang et al. “A survey of security visualization for computer network logs”. Em: *Security and Communication Networks* (2011), pp. 404–421. DOI: <10.1002/sec.324>.

ANEXO A – Instalação e configuração do Servidor OSSEC+ no Ubuntu 16.04

Antes de começar a instalação é necessário criar uma conta de usuário a partir da seguinte URL:

```
https://www.ossec.net/register-for-ossec/
```

Logo após, é necessário coletar a licença gratuita do OSSEC+ a partir da seguinte URL:

```
https://www.atomicorp.com/amember/member
```

Acessando a plataforma, é necessário ir à seção Add/Renew Licenses e adicionar a licença do OSSEC+. Depois, é necessário clicar em View Cart e, finalmente, em Checkout. Em seguida, no terminal do Ubuntu, é necessário obter privilégios de super usuário e inserir os seguintes comandos:

```
# apt-get update  
# wget -q -O - https://updates.atomicorp.com/installers/oum | bash
```

Nesta parte será necessário informar o usuário e senha cadastrados anteriormente. Após a inserção da credencial, realiza-se os comandos

```
# oum configure  
# oum update
```

A instalação do servidor está concluída. Para rodá-lo é necessário ir até o caminho `/var/bin/ossec` no terminal do Ubuntu e digitar o seguinte comando:

```
# ./ossec-control start
```

No mesmo caminho, pode-se iniciar a configuração dos agentes com o seguinte comando

```
# ./manage-agents
```

```
*****
* OSSEC HIDS v3.6.0 Agent manager.      *
* The following options are available: *
*****
(A)dd an agent (A).
(E)xtract key for an agent (E).
(L)ist already added agents (L).
(R)emove an agent (R).
(Q)uit.
Choose your action: A,E,L,R or Q: E
```

Figura A.1 – Gerenciador de agentes do OSSEC+

Com o terminal aberto, digita-se “A” para adicionar o agente. Aqui é especificado o nome, o endereço de IP do agente e o seu identificador (ID).

```
Choose your action: A,E,L,R or Q: a
- Adding a new agent (use '\q' to return to the main menu).
Please provide the following:
* A name for the new agent: nomeagente
* The IP Address of the new agent: 192.168.110.20
* An ID for the new agent[002]: 005
Agent information:
ID:005
Name:nomeagente
IP Address:192.168.110.20
Confirm adding it?(y/n): y
Agent added with ID 005.
```

Figura A.2 – Adição do agente ao servidor OSSEC+

Em seguida, extrai-se a chave para o pareamento entre o gerenciador e o agente digitando “E” no terminal do Agent Manager. É necessário informar o ID do agente para obtê-la.

```
Choose your action: A,E,L,R or Q: e
Available agents:
ID: 005, Name: nomeagente, IP: 192.168.110.20
Provide the ID of the agent to extract the key (or '\q' to quit): 005
Agent key information for '005' is:
MDA1IG5vbWVhZ2VudGUgMTkyLjE2OC4xMTAuMjAgMWUzNTZjZjliOTMxOGY3MjZhZDVlNGM0MmFmMWM3
NTNiNWExZjc1Y2NjZBmNTVlNmIzZDRhZmNlNDA1MzRhMQ==
```

Figura A.3 – Obtenção da chave de pareamento entre o gerenciador e o agente

ANEXO B – Instalação e configuração do Agente OSSEC no Ubuntu 16.04

Primeiramente é necessário obter privilégios de super usuário. Em seguida digita-se os seguintes comandos.

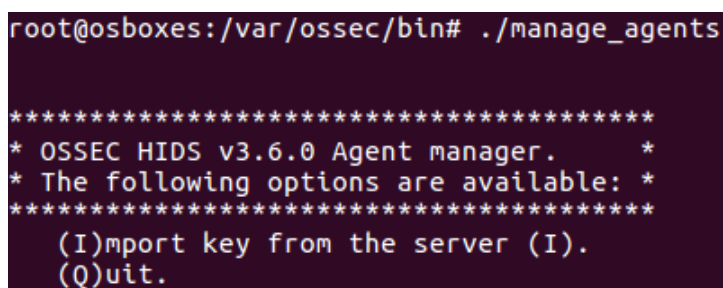
```
# wget -q -O - https://updates.atomicorp.com/installers/atomic | sudo bash
# apt-get update
# sudo apt-get install ossec-hids-agent
```

Após a instalação do agente é necessário iniciá-lo. No diretório `/var/ossec/bin`, executa-se o seguinte comando:

```
#. /ossec-control start
```

Em seguida, é necessário realizar o pareamento entre o agente e o servidor.

```
# ./manage-agents
```



```
root@osboxes:/var/ossec/bin# ./manage_agents
*****
* OSSEC HIDS v3.6.0 Agent manager.      *
* The following options are available:  *
*****
  (I)mport key from the server (I).
  (Q)uit.
  (R)estart agent (R).
```

Figura B.1 – Importação da chave de pareamento no agente OSSEC

Em seguida, digita-se “I” e insere-se a chave obtida a partir do servidor. Após a confirmação da adição, é necessário reiniciar o agente OSSEC com o seguinte comando:

```
# ./ossec-control restart
```

No servidor OSSEC+ é possível verificar se o agente está conectado corretamente a ele ao digitar o seguinte comando:

```
# /var/ossec/bin/agent_control - lc
```

ANEXO C – Instalação do Agente OpenEDR no Windows 10

Na seguinte URL, escolher a versão 2.0.0.0 de 64 bits do instalador do agente e seguir os passos da instalação.

<https://github.com/ComodoSecurity/openedr/releases/>

ANEXO D – Instalação do Servidor Wazuh no CentOS7

Nesta instalação, mostram-se os passos para conseguir instalar o Wazuh Manager, Wazuh API e também o Filebeat, que será utilizado para repassar os logs obtidos do agente para o servidor ELK. Importante ressaltar que os comandos devem ser realizados tendo-se o privilégio de administrador da máquina.

```
sudo su
```

Para que os downloads sejam feitos, é necessário acrescentar o repositório do Wazuh na máquina a ser utilizada.

```
# cat > /etc/yum.repos.d/wazuh.repo « EOF
[wazuh_repo]
gpgcheck=1
gpgkey=https://packages.wazuh.com/key/GPG-KEY-WAZUH
enabled=1
name=Wazuh repository
baseurl=https://packages.wazuh.com/4.x/yum/
protect=1
EOF
```

A fim de instalar o Wazuh Manager e iniciá-lo, basta rodar os comandos a seguir:

```
# yum -y install wazuh-manager
# systemctl start wazuh-manager
```

Em seguida, são demonstrados os comando que possibilitam a criação de uma senha. Esta será utilizada pelos agentes que querem se conectar ao servidor. Também habilita-se a permissão para o auto-registro dos agentes, que possuindo as configurações corretas, serão pareados automaticamente ao servidor. Importante ressaltar que “<SenhaEscolhida>” no comando, deve ser substituída pela senha de sua preferência.

```
# grep “<use_password>” -B7 -A8 /var/ossec/etc/ossec.conf
# sed -i “s/<use_password>no/<use_password>yes/” /var/ossec/etc/ossec.conf
# grep “<use_password>” -B7 -A8 /var/ossec/etc/ossec.conf
# echo “<SenhaEscolhida>” > /var/ossec/etc/authd.pass
# systemctl restart wazuh-manager
```

Em seguida, é possível ver que o serviço do Wazuh Manager está funcionando. Nota-se também que o serviço utiliza as portas 1514 e 1515.

```
# systemctl status wazuh-manager
# netstat -natp | egrep "(1514|1515)"
```

Vale ressaltar que, para que o pareamento entre o Wazuh Agent e o Wazuh Manager seja possível, deve-se, ainda, realizar a correta configuração com relação às portas utilizadas.

Para o próximo passo, acrescenta-se o repositório do Wazuh na máquina em questão.

```
# rpm -import https://packages.wazuh.com/key/GPG-KEY-WAZUH
# cat /etc/yum.repos.d/wazuh.repo « EOF
[wazuh]
gpgcheck=1
gpgkey=https://packages.wazuh.com/key/GPG-KEY-WAZUH
enabled=1
name=EL-$releasever - Wazuh
baseurl=https://packages.wazuh.com/4.x/yum/
protect=1
EOF
```

Faz-se, então, a instalação do filebeat.

```
# yum install filebeat-7.10.2
```

O próprio repositório do Wazuh disponibiliza um arquivo base de configuração do Filebeat para o envio dos alertas para a pilha ELK. Obtém-se o arquivo e são dados os privilégios de leitura.

```
# curl -so /etc/filebeat/filebeat.yml https://raw.githubusercontent.com/wazuh/wazuh/v4.3.6/extensions/filebeat/7.x/filebeat.yml
# chmod go+r /etc/filebeat/filebeat.yml
```

Realiza-se a importação o módulo do Filebeat, que foi desenvolvido para ser utilizado junto ao Wazuh.

```
# curl -s https://packages.wazuh.com/4.x/filebeat/wazuh-filebeat-0.1.tar.gz | sudo tar -xvz
-C /usr/share/filebeat/module
```

Configura-se o arquivo do Filebeat com o IP do servidor ELK, para onde deverão ser enviados os alertas. Ressalta-se que “<SeuIpConfiguradoNoELK>”, deverá ser substituído no comando pelo IP configurado na máquina contendo a pilha ELK.

```
# sed -i "s/YOUR_ELASTIC_SERVER_IP/<SeuIpConfiguradoNoELK>/" /etc/filebeat/filebeat.yml
```

Com estas configurações, pode-se, então, iniciar o Filebeat e verificar seu status.

```
# systemctl daemon-reload
# systemctl enable filebeat.service
# systemctl start filebeat.service
# systemctl status filebeat.service
```

Por fim, para que não ocorram atualizações automáticas, que podem gerar conflitos nas configurações utilizadas, usa-se o comando a seguir.

```
# sed -i "s/^enabled=1/enabled=0/" /etc/yum.repos.d/wazuh.repo
```

ANEXO E – Instalação do Agente Wazuh no CentOS7

Para realizar a instalação, em primeira instância, é necessário possuir privilégios de administrador na máquina a ser utilizada.

```
sudo su
```

Para que os downloads sejam feitos, é necessário acrescentar o repositório do Wazuh na máquina a ser utilizada.

```
# cat > /etc/yum.repos.d/wazuh.repo « EOF
[wazuh_repo]
gpgcheck=1
gpgkey=https://packages.wazuh.com/key/GPG-KEY-WAZUH
enabled=1
name=Wazuh repository
baseurl=https://packages.wazuh.com/4.x/yum/
protect=1
EOF
```

Com estas configurações feitas, pode-se, portanto, baixar os arquivos necessários e realizar a instalação. O comando a seguir já leva em consideração os parâmetros necessários para que haja a conexão entre o Wazuh Agent e o Wazuh Manager, para isto, são passadas as configurações contendo o IP do Wazuh Manager, além da senha configurada no mesmo e o protocolo a ser utilizado, o TCP. Ressalta-se que “<IpWazuhManager>” deve ser substituído pelo IP configurada na máquina do Wazuh Manager, e “<SenhaEscolhida>” deve ser substituída pela senha escolhida ao configurar o Wazuh Manager.

```
# WAZUH_MANAGER="<IpWazuhManager>" WAZUH_REGISTRATION_PASSWORD="<SenhaEscolhida>" WAZUH_PROTOCOL="tcp" yum -y install wazuh-agent
```

Obtendo-se sucesso, pode-se iniciar o Wazuh Agent e visualizar o status do mesmo.

```
# systemctl start wazuh-agent
# systemctl status wazuh-agent
```

Com o Wazuh Agent funcionando, verifica-se o status da conexão existente entre o servidor e o agente.

```
# grep ^status /var/ossec/var/run/wazuh-agentd.state
```

Caso o status não esteja conectado é preciso verificar as configurações realizadas, um motivo para que o pareamento não seja possível é a configuração existente na máquina do servidor, que pode estar bloqueando a comunicação com a porta utilizada.

Por fim, para que não ocorram atualizações automáticas que podem gerar conflitos nas configurações utilizadas, usa-se o comando a seguir.

```
# sed -i "s/^enabled=1/enabled=0/" /etc/yum.repos.d/wazuh.repo
```

ANEXO F – Instalação da pilha ELK no CentOS7

Para que a realização da instalação da pilha ELK ocorra sem maiores problemas, é necessário atentar-se a quantidade de memória RAM disponível na máquina utilizada. No caso em questão, utilizou-se uma máquina com 8GB de RAM.

A fim de realizar a instalação, em primeira instância, é necessário possuir privilégios de administrador na máquina a ser utilizada.

```
sudo su
```

Faz-se, então, o comando necessário para atualizar o sistema.

```
# yum update
```

Realiza-se, em seguida, a instalação de dependências necessárias.

```
# yum install -assumeyes java-11-openjdk java-11-openjdk-devel  
# yum install curl unzip wget
```

Para que os downloads sejam feitos, é necessário acrescentar o repositório do Elasticsearch na máquina a ser utilizada.

```
# rpm -import https://packages.elastic.co/GPG-KEY-elasticsearch  
# cat > /etc/yum.repos.d/elastic.repo « EOF  
[elasticsearch-7.x]  
name=Elasticsearch repository for 7.x packages  
baseurl=https://artifacts.elastic.co/packages/7.x/yum  
gpgcheck=1  
gpgkey=https://artifacts.elastic.co/GPG-KEY-elasticsearch  
enabled=1  
autorefresh=1  
type=rpm-md  
EOF
```

Passa-se, então, para, de fato, a instalação do Elasticsearch, inicializando-o e verificando o seu status.

```
# yum -y install elasticsearch-7.10.2
# systemctl daemon-reload
# systemctl enable elasticsearch.service
# systemctl start elasticsearch.service
# systemctl status elasticsearch.service
```

A seguir são feitas configurações para otimizar o Elasticsearch, possibilitando a fragmentação otimizada do processo de indexação e também os valores de memória heap que serão alocados.

```
# sed -i "s/#bootstrap.memory_lock: true/bootstrap.memory_lock: true/" /etc/elasticsearch/elasticsearch.yml
# mkdir -p /etc/systemd/system/elasticsearch.service.d/
# echo -e "[Service]\nLimitMEMLOCK=infinity" > /etc/systemd/system/elasticsearch.service.d/elasticsearch.conf
# sed -i "s/-Xms.* / -Xms3g;/s/-Xmx.* / -Xmx3g/" /etc/elasticsearch/jvm.options
# systemctl daemon-reload
# systemctl restart elasticsearch.service
# systemctl status elasticsearch.service
```

Em seguida, realiza-se a instalação do Kibana.

```
# yum install -y kibana-7.10.2
```

Mudam-se as características do diretório para que este pertença ao Kibana.

```
# mkdir /usr/share/kibana/data
# chown -R kibana:kibana /usr/share/kibana
```

Baixa-se, então, o plugin do Wazuh a ser utilizado pelo Kibana.

```
# cd /usr/share/kibana/
# sudo -u kibana bin/kibana-plugin install https://packages.wazuh.com/4.x/ui/kibana/wazuh_kibana-4.3.6_7.10.2-1.zip
```

Realiza-se a configuração para mudar a porta utilizada pelo Kibana e permitir conexões externas.

```
# cat » /etc/kibana/kibana.yml « EOF
server.host: "0.0.0.0"
server.port: 443
EOF
```

Para que o Kibana seja capaz de realizar as conexões necessárias, é preciso torná-lo root. Em seguida, pode-se iniciar o serviço, verificando seu status.

```
# setcap "CAP_NET_BIND_SERVICE=+eip" /usr/share/kibana/node/bin/node
# systemctl daemon-reload
# systemctl enable kibana.service
# systemctl start kibana.service
# systemctl status kibana.service
```

Realiza-se a configuração para que o Kibana seja capaz de se comunicar com a API do Wazuh, que fica disponibilizada na máquina em que foi feita a instalação do Wazuh Manager. A API utiliza a porta 55000, por isso, para que a comunicação seja possível, é importante lembrar de realizar as configurações necessárias no firewall do sistema operacional em que se encontra instalado o Wazuh Manager.

```
# cat > /usr/share/kibana/data/wazuh/config/wazuh.yml « EOF
hosts:
  - wazuhapi:
      url: https://<IpWazuhManager>
      port: 55000
      username: wazuh-wui
      password: wazuh-wui
EOF
```

Para que não ocorram atualizações automáticas que podem gerar conflitos nas configurações utilizadas, usa-se o comando a seguir.

```
# sed -i "s/^enabled=1/enabled=0/" /etc/yum.repos.d/wazuh.repo
```

Por fim, é possível configurar para que o Kibana seja acessado somente com o uso de senha.

```
# echo "xpack.security.enabled: true" » /etc/elasticsearch/elasticsearch.yml
# systemctl restart elasticsearch
# cd /usr/share/elasticsearch/bin/
# ./elasticsearch-setup-passwords interactive
# cat » /etc/kibana/kibana.yml « EOF
xpack.security.enabled: true
elasticsearch.username: "elastic"
elasticsearch.password: "<SenhaColocadaNoPassoAnterior>"
EOF
```