UNIVERSIDADE FEDERAL RURAL DE PERNAMBUCO

Jonathas Felipe da Silva

Detecção e Modelagem de Ameaças Persistentes Avançadas na Fase de Movimentação Lateral: Uma Abordagem com Process Mining

Recife

2025

Jonathas Felipe da Silva

Detecção e Modelagem de Ameaças Persistentes Avançadas na Fase de Movimentação Lateral: Uma Abordagem com Process Mining

Monografia apresentada ao Curso de Bacharelado em Sistemas de Informação da Universidade Federal Rural de Pernambuco, como requisito parcial para obtenção do título de Bacharel em Sistemas de Informação.

Universidade Federal Rural de Pernambuco – UFRPE

Departamento de Estatística e Informática

Curso de Bacharelado em Sistemas de Informação

Orientador: Fernando Antonio Aires Lins

Coorientador: Milton Vinicius Morais de Lima

Recife

2025

Dados Internacionais de Catalogação na Publicação
Sistema Integrado de Bibliotecas da UFRPE
Bibliotecário(a): Auxiliadora Cunha – CRB-4 1134

S586d Silva, Jonathas Felipe da.
Detecção e modelagem de ameaças persistentes avançadas na fase de movimentação lateral: uma abordagem com Process Mining / Jonathas Felipe da Silva. – Recife, 2025.
55 f.; il.

Orientador(a): Fernando Antonio Aires Lins.
Co-orientador(a): Milton Vinicius Morais de Lima.

Trabalho de Conclusão de Curso (Graduação) – Universidade Federal Rural de Pernambuco, Bacharelado em Sistemas da Informação, Recife, BR-PE, 2025.

Inclui referências e apêndice(s).

1. Ciberterrorismo. 2. Vírus de computador. 3. Cibernética. 4. Fatores de qualidade de software em uso I. Lins, Fernando Antonio Aires, orient. II. Lima, Milton Vinicius Morais de, coorient. III. Título

CDD 004

JONATHAS FELIPE DA SILVA

Detecção e Modelagem de Ameaças Persistentes Avançadas na Fase de Movimentação Lateral: Uma Abordagem com Process Mining

Monografia apresentada ao Curso de Bacharelado em Sistemas de Informação da Universidade Federal Rural de Pernambuco, como requisito parcial para obtenção do título de Bacharel em Sistemas de Informação.

Aprovada em: 20 de Março de 2025.

BANCA EXAMINADORA

Fernando Antonio Aires Lins (Orientador)
Departamento de Computação
Universidade Federal Rural de Pernambuco

Milton Vinicius Morais de Lima (Coorientador)
Centro Integrado de Segurança em Sistemas Avançados
CESAR School

Gabriel Alves de Albuquerque Junior
Departamento de Estatística e Informática
Universidade Federal Rural de Pernambuco

Agradecimentos

Agradeço à A Deus, por me conceder forças, sabedoria e a oportunidade de concluir esta importante etapa da minha vida.

À minha esposa, Vitória, pelo amor, paciência e incentivo incondicional ao longo desta jornada, sendo minha base nos momentos de desafio.

Aos meus pais, Josival e Mauriceia, por todo amor, educação e valores que me transmitiram ao longo da minha caminhada.

Aos meus irmãos, Esther e Davih, por serem parte essencial da minha vida e do meu crescimento.

Ao meu orientador, Fernando Antonio Aires Lins, pela excelente orientação, paciência e valiosas contribuições, fundamentais para o desenvolvimento deste trabalho.

Ao meu coorientador, Milton Vinicius Morais de Lima, pela incansável disposição, assistência de alto nível e apoio contínuo durante esta pesquisa.

Aos professores da Universidade Federal Rural de Pernambuco (UFRPE), que compartilharam conhecimento e incentivaram minha busca pelo aprendizado e pela excelência acadêmica.

À UFRPE, pela infraestrutura e recursos disponibilizados para a realização deste estudo.

Aos colegas que compartilharam essa jornada acadêmica comigo, pelo apoio mútuo, troca de conhecimento e companheirismo.

E a todos que, de alguma forma, contribuíram direta ou indiretamente para a realização deste trabalho.

“A sabedoria é a coisa principal; adquire, pois, a sabedoria, e com tudo o que possuis adquire o entendimento.”

(Provérbios 4:7)

Resumo

A crescente ameaça de ataques cibernéticos complexos tem exigido estratégias avançadas de defesa, especialmente na detecção precoce de atividades suspeitas em redes comprometidas. Com isso, Ameaças Persistentes Avançadas (APTs) representam um desafio significativo para a segurança cibernética, caracterizando-se por ataques sofisticados e direcionados. Este trabalho tem como objetivo investigar a movimentação lateral dentro de redes comprometidas, utilizando mineração de processos para detectar padrões suspeitos de comportamento. Para isso, foi configurado um ambiente experimental com máquinas virtuais simulando um ataque APT. *Logs* do sistema e do Wazuh registraram as atividades, possibilitando a extração de eventos relevantes para o presente estudo. A metodologia consiste na coleta de dados em dois cenários: uso normal e ataque, seguida pela aplicação de algoritmos de *Process Mining*, como AlphaMiner, através da biblioteca pm4py. Com isso, foi possível identificar diferenças estruturais entre os processos normais e aqueles manipulados pelo invasor, possibilitando a criação de indicadores de comprometimento (IoCs). Os resultados contribuem para a melhoria de mecanismos de detecção e resposta a APTs, auxiliando na proteção de redes corporativas contra ataques avançados.

Palavras-chave: Ameaças Persistentes Avançadas, Mineração de Processos, Movimentação Lateral

Abstract

The growing threat of complex cyberattacks has demanded advanced defense strategies, particularly for the early detection of suspicious activities in compromised networks. In this context, Advanced Persistent Threats (APTs) represent a significant cybersecurity challenge, characterized by sophisticated and targeted attacks. This study aims to investigate lateral movement within compromised networks using process mining to detect suspicious behavioral patterns. To achieve this, an experimental environment was set up with virtual machines simulating an APT attack. System logs and Wazuh recorded the activities, enabling the extraction of relevant events for this study. The methodology consists of data collection in two scenarios: normal usage and attack, followed by the application of Process Mining algorithms, such as AlphaMiner, using the pm4py library. This approach allowed for the identification of structural differences between normal processes and those manipulated by the attacker, enabling the creation of Indicators of Compromise (IoCs). The results contribute to improving detection and response mechanisms for APTs, aiding in the protection of corporate networks against advanced attacks.

Keywords: Advanced Persistent Threat, Process Mining, Lateral Movement

Lista de ilustrações

Lista de tabelas

Lista de abreviaturas e siglas

APT	Advanced Persistent Threat
TTP	Táticas, Técnicas e Procedimentos
LM	Lateral Movement

Sumário

1 Introdução

A segurança cibernética tem se tornado uma área crítica no contexto das organizações inclusive governamentais devido ao crescimento exponencial de ameaças digitais. Com isso, as Ameaças Persistentes Avançadas destacam-se por sua complexidade e persistência, utilizando técnicas sofisticadas para se infiltrar e manter acesso não autorizado a sistemas por longos períodos (BUCHTA et al., 2024). Um dos estágios mais críticos das APTs é a movimentação lateral, que permite ao atacante expandir seu controle sobre a rede comprometida, aumentando o impacto e a dificuldade de detecção (SAKTHIVELU; KUMAR, 2023).

As Ameaças Persistentes Avançadas (APTs, do inglês Advanced Persistent Threat) são uma das maiores ameaças cibernéticas enfrentadas por empresas de diversos setores. Esses ataques são conduzidos por grupos altamente sofisticados, muitas vezes com motivações políticas ou financeiras, visando a espionagem industrial, roubo de dados sensíveis ou sabotagem de infraestruturas críticas. Um exemplo ocorreu na Alemanha, onde 68% das empresas foram vítimas de ataques cibernéticos nos últimos anos, incluindo um ataque direcionado a uma siderúrgica. O ataque explorou vulnerabilidades na in fraestrutura da empresa, causando falhas operacionais e impactos financeiros (REISDOERFER; ALCÂNTARA, 2020).

Outro caso é o de ataques direcionados a empresas petrolíferas, conforme documentado por Galoyan (2019). Empresas do setor energético são alvos frequentes devido ao valor estratégico das informações que detêm. Os invasores utilizam técnicas avançadas, como *spear phishing* e *malware* sofisticado, para infiltrar-se nos sistemas das empresas e manter o acesso de forma persistente. O impacto inclui o roubo de dados comerciais, interrupção de operações e até mesmo o comprometimento da segurança nacional em casos de infraestrutura crítica. Além disso, as vítimas frequentemente não percebem o ataque até que os danos já tenham sido causados.

A crescente sofisticação dos ataques APTs também foi observada em ataques a infraestruturas críticas nos Estados Unidos, atingindo agências governamentais e empresas privadas. De acordo com Junior (2024), esses ataques foram conduzidos por grupos altamente organizados, possivelmente ligados a interesses estatais. O ataque se destacou por sua complexidade, envolvendo múltiplos vetores de intrusão e exploração de vulnerabilidades em diferentes camadas da rede. A necessidade de estratégias de defesa mais eficazes, incluindo o uso de inteligência cibernética e machine learning, tornou-se essencial para combater essa ameaça crescente.

Dada a importância da detecção precoce para mitigar os danos de ataques APT,

este trabalho propõe a utilização de Mineração de Processos para identificar padrões suspeitos na fase de movimentação lateral. A mineração de processos tem se mostrado eficaz na análise de eventos registrados em *logs*, permitindo a reconstrução e interpretação do fluxo de execução de processos de um sistema (MACAK et al., 2022). Dessa forma, este trabalho busca contribuir para o aprimoramento dos mecanismos de detecção de APTs, oferecendo uma abordagem inovadora baseada em dados reais extraídos de ambientes simulados.

1.1 Objetivo

O objetivo principal deste trabalho é desenvolver uma abordagem baseada em mineração de processos para a detecção e modelagem de movimentação lateral em ataques APTs. Para isso, foram utilizados logs gerados em um ambiente experimental, simulando tanto o funcionamento normal do sistema quanto um ataque simulado. A partir desses dados, aplicamos técnicas de mineração de processos para identificar padrões anômalos de comportamento e mapear potenciais vetores de ataque.

1.2 Objetivos Específicos

- Projetar e implementar um ambiente controlado que permita simular ataques APTs de forma segura e realista.
- Coletar, organizar e estruturar dados de logs do sistema operacional relacionados aos cenários de ataque.
- Modelar uma notação do comportamento suspeito que permita a interpretação do fluxo dos logs.
- Aplicar técnicas de mineração de processos para mapear fluxos de atividade e identificar comportamentos suspeitos.
- Comparar padrões de movimentação lateral com o comportamento normal do sistema, destacando diferenças significativas.

Este trabalho está organizado da seguinte forma: no Capítulo 2 é apresentado a fundamentação teórica que norteou este trabalho, descrevendo APTs, movimentação lateral e Mineração de Processos. Em seguida, o Capítulo 3 descreve a metodologia adotada, incluindo a configuração do ambiente experimental e a coleta de dados. Já o Capítulo 4 descreve os resultados obtidos e sua análise. Por fim, o Capítulo 5 apresenta as conclusões e direções para trabalhos futuros.

2 Fundamentação Teórica

Esta seção apresenta a fundamentação teórica necessária para o entendimento do trabalho. Inicialmente, discute-se o conceito de Ameaças Persistentes Avançadas (APTs), destacando sua complexidade e persistência. Em seguida, aborda-se a movimentação lateral, uma fase crítica desses ataques, que permite a expansão na rede comprometida. Em seguida, explora-se a Mineração de Processos como uma abordagem para detecção de padrões suspeitos. Por fim, são descritas as ferramentas utilizadas na pesquisa, essenciais para a implementação e avaliação da solução proposta.

2.1 Ameaças Persistentes Avançadas

O termo *Advanced Persistent Threat* foi mencionado pela primeira vez em um pedido de patente nos Estados Unidos, registrado em 2007 e publicado em 2008 por Schmidt, Rattray e Fogle (2008). As APTs referem-se a ameaças cibernéticas altamente sofisticadas e contínuas, caracterizadas por ataques direcionados que visam comprometer sistemas de alto valor, como redes corporativas ou governamentais. Esses ataques são projetados para permanecerem ocultos por longos períodos, permitindo que os invasores realizem espionagem, roubem informações sensíveis ou até sabotem infraestruturas críticas. Para serem bem-sucedidas, as APTs exigem um elevado grau de furtividade e persistência ao longo do tempo (GHAFIR; PRENOSIL et al., 2014).

Alshamrani et al. (2019) resume as principais diferenças entre ataques APTs e ataques tradicionais em diferentes aspectos na Tabela 1.

Tabela 1 – Comparação entre Ataques Tradicionais e Ataques APT

Aspecto	**Ataques Tradicionais**	**Ataques APT**
Atacante	Geralmente uma única pessoa	Altamente organizado, sofisticado, determinado e com muitos recursos
Alvo	Não especificado, em geral sistemas individuais	Organizações específicas, instituições governamentais, empresas comerciais
Objetivo	Benefícios financeiros, demonstração de habilidades	Vantagens competitivas, benefícios estratégicos
Abordagem	Única tentativa, *smash and grab*, curto período	Tentativas repetidas, atua de forma discreta e adaptável para resistir às defesas, longo prazo

Fonte: Adaptado de (ALSHAMRANI et al., 2019).

Na literatura, o conceito de APT (Ameaça Persistente Avançada) é abordado de diferentes maneiras, refletindo a diversidade de perspectivas e interpretações presentes nos estudos sobre o tema (BUCHTA et al., 2024).

Ameaça. Uma ameaça é a presença de indivíduos ou grupos com a motivação, capacidade e oportunidade de realizar ações maliciosas. A intenção, combinada com a capacidade, é o elemento central que define e impulsiona a ameaça.

Avançada. Um grupo de invasores de ameaça avançado possui habilidades técnicas sofisticadas, sendo altamente competente em identificar ou criar vulnerabilidades em sistemas e elaborar métodos inovadores para explorá-las. Esses invasores podem empregar estratégias como o uso de *exploits* de dia zero, infiltração por agentes internos ou ataques à cadeia de suprimentos, especialmente quando as defesas do alvo apresentam desafios significativos.

Persistente. Uma ameaça persistente é caracterizada por sua determinação em alcançar seus objetivos, utilizando habilidades para se adaptar a ambientes hostis enquanto evita a detecção. Esse tipo de ameaça possui recursos variados para atualizar continuamente suas Táticas, Técnicas e Procedimentos (TTPs), superando contramedidas defensivas, estratégicas e táticas. Esses recursos vão além do financiamento e incluem ativos como capital humano qualificado, essencial para o planejamento e execução de operações. Normalmente, essas ameaças operam em equipes devido à amplitude e complexidade de suas ações, usando outros ativos, como infraestrutura, tempo e suporte financeiro, para sustentar suas operações a longo prazo.

O ciclo de vida de uma Ameaça Persistente Avançada é descrito na literatura, de acordo com Quintero-Bonilla e Rey (2020), por meio de diferentes divisões em múltiplos estágios, variando em quantidade conforme a abordagem adotada. As principais abordagens variam de três a onze estágios. Neste trabalho, adotaremos a interpretação que descreve um APT em cinco estágios.

Fases de uma Ameaça Persistente Avançada:

1. **Reconhecimento**. Os atacantes identificam e coletam informações sobre o alvo, como pontos fracos, sistemas, pessoas e tecnologias que possam ser explorados.

2. **Infiltração**. Utilizam técnicas como *phishing* ou exploração de vulnerabilidades para obter acesso inicial ao sistema ou rede do alvo.

3. **Movimentação lateral**. Uma vez dentro, os invasores exploram a rede, movimentando-se entre sistemas para aumentar seu alcance e obter privilégios elevados.

4. **Exfiltração**. Dados sensíveis são coletados e transferidos para fora do sistema comprometido, geralmente de forma discreta para evitar detecção.

5. **Pós-ataque**. Após atingir seus objetivos, os atacantes removem rastros, implantam *backdoors* ou criam mecanismos para manter acesso futuro.

A Movimentação Lateral refere-se à exploração interna de uma rede comprometida, onde o objetivo do atacante é expandir seu alcance, identificar nós críticos e coletar informações valiosas, como dispositivos, serviços e dados relevantes e sensíveis para a organização. Os comportamentos de movimentação lateral no nível da rede incluem o envio de solicitações de conexão para outros usuários ativamente, *sniffer* de rede, varredura de rede e assim por diante (LI et al., 2016).

A Figura 1 apresenta um fluxo estruturado e furtivo para comprometer sistemas sem ser detectada. Inicialmente, os invasores realizam reconhecimento para identificar alvos e explorar vulnerabilidades. Após a intrusão, estabelecem persistência e evitam detecção por meio de técnicas como ofuscação e escalonamento de privilégios. Na fase de movimentação lateral, exploram credenciais e acessam sistemas críticos, expandindo seu controle dentro da rede. Por fim, exfiltram dados ou executam ações maliciosas, como sabotagem.

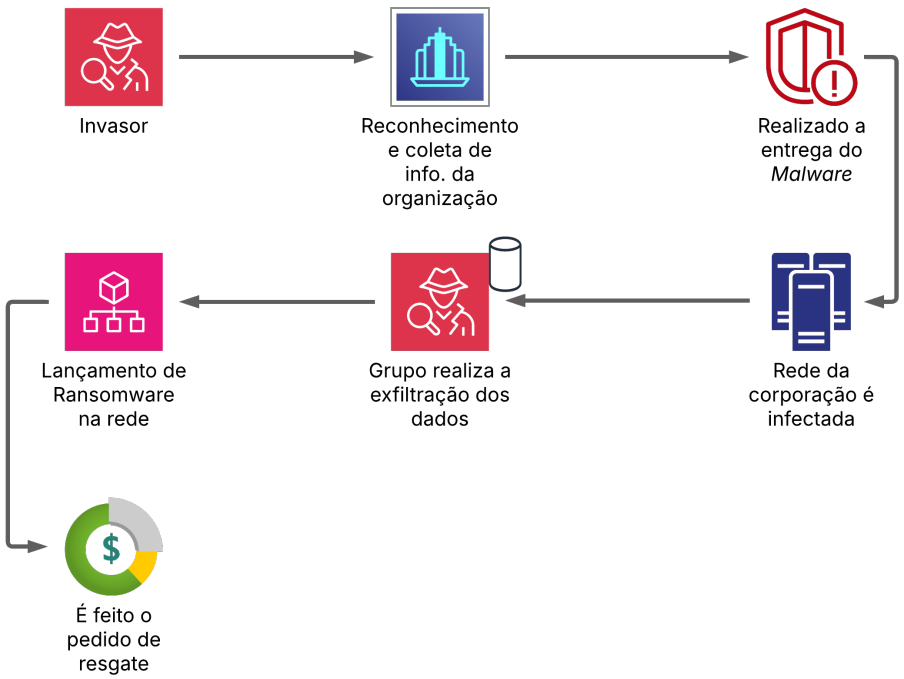

Figura 1 – Ameaça Persistente Avançada
Fonte: O autor.

As APTs são altamente sofisticados que realizam ataques cibernéticos de longo prazo com objetivos estratégicos. Para entender sua periculosidade, a seguir são apre-

sentadas três características fundamentais: a ameaça em si (Ameaças), sua persistência ao longo do tempo (Persistentes) e seu nível de sofisticação técnica (Avançadas).

2.2 Movimentação Lateral em APTs

De acordo com Lehto (2022a), na fase de movimentação lateral, o atacante busca explorar e compreender o ambiente TIC (Tecnologia da Informação e Comunicação) da vítima. Durante esse processo, o invasor trabalha para expandir a superfície de ataque, utilizando a rede alvo após comprometer alguns de seus sistemas iniciais. Além disso, o atacante tenta escalar privilégios, adquirindo credenciais, acessos ou outros recursos essenciais que possibilitem o controle de mais ativos e sistemas dentro do ambiente comprometido.

Conforme Lehto (2022b), o movimento lateral do ciclo de vida do APT, se destaca como um estágio crítico onde os intrusos aumentam os privilégios e se movem pela rede para expandir seu controle e acesso a dados confidenciais. A movimentação lateral de APTs também exige que os invasores utilizem estratégias furtivas para evitar a detecção durante sua exploração na rede. Isso inclui o uso de ferramentas customizadas e táticas de ofuscação que mascaram suas ações, permitindo que se misturem ao tráfego legítimo da rede. Além disso, os atacantes podem adotar técnicas como *Living off the Land* (LoL), utilizando ferramentas e processos legítimos do sistema-alvo para reduzir os sinais que poderiam alertar sistemas de defesa ou administradores.

Outro aspecto relevante é a dependência de infraestrutura comprometida pelos invasores. Durante essa fase, sistemas já infiltrados são usados como pontos de apoio para acessar outras partes da rede, esse processo é chamado de *pivoting*. Para isto, é frequentemente combinado com o uso de credenciais roubadas ou replicadas, como *hashes* e *tokens*, para autenticação em máquinas e serviços adicionais (SMILIOTOPOULOS; KAMBOURAKIS; KOLIAS, 2024). Tais práticas tornam a movimentação lateral uma fase progressiva, onde cada passo bem-sucedido aumenta o controle do atacante sobre a rede.

Por fim, a movimentação lateral não é apenas uma exploração técnica, mas também exige um profundo entendimento da arquitetura e dos ativos críticos da organização-alvo. Os invasores frequentemente analisam e priorizam dispositivos, identificando aqueles que possuem maior valor estratégico, como servidores de dados confidenciais ou sistemas relacionados à administração de identidade. Isso torna a fase de movimentação lateral uma combinação de inteligência operacional e execução técnica, um fator que ressalta ainda mais a sua importância no ciclo de vida de ataques APTs.

No contexto deste trabalho, a fase de movimentação lateral é o foco principal de análise e estudo, devido à sua importância estratégica dentro do ciclo de vida das

ameaças persistentes avançados (APTs). Esta etapa crítica destaca-se pela sua complexidade e pelo papel que desempenha na consolidação do controle do invasor sobre o ambiente comprometido. Assim, o escopo do trabalho será delineado para explorar em profundidade os métodos, ferramentas e técnicas empregadas nessa fase, bem como suas implicações na segurança cibernética.

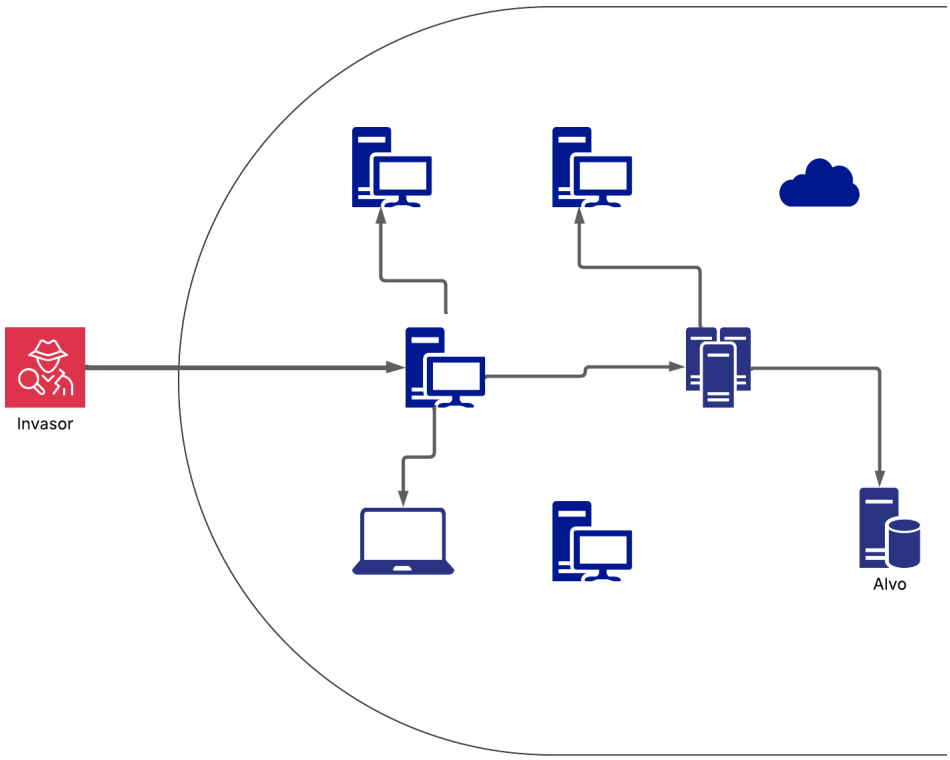

Figura 2 – Movimentação Lateral.
Fonte: O autor.

A Figura 2 demonstra o comportamento de um APT durante a fase de movimentação lateral. Nesse estágio, o atacante se espalha pela rede explorando vulnerabilidades e comprometendo dispositivos conectados. O invasor utiliza os sistemas previamente comprometidos como pontos de apoio para ampliar sua presença, movendo-se estrategicamente até localizar e comprometer seu alvo principal, como um servidor crítico ou um banco de dados sensível.

A Tabela 2 apresenta uma visão organizada de métodos utilizados por invasores para expandir seu controle em rede comprometidas. Ela está divida em duas categorias principais: Acesso Remoto e Execução Remota, destacando tantos os mecanismos empregados quanto suas funcionalidades. Essas técnicas e ferramentas são utilizadas em sistemas operacionais Windows.

	Técnica	Descrição
Acesso Remoto	RDP	RDP é um protocolo da Microsoft que permite acesso remoto a um computador, usado para controle e suporte à distância.
	SSH	SSH (Secure Shell) é um protocolo de rede que permite acesso remoto seguro a servidores e dispositivos, garantindo confidencialidade e integridade dos dados através de criptografia.
	Map Network Shares	É um recurso de compartilhamento em uma rede, como uma pasta ou unidade, permitindo acesso fácil e direto a esse recurso a partir de um dispositivo remoto.
Execução Remota	PsExec	É uma ferramenta de linha de comando usada para executar programas ou scripts remotamente em dispositivos, iniciando serviços na máquina de destino.
	Schedule Tasks	É um mecanismo para automatizar operações, permitindo programar a execução de aplicativos ou scripts em horários específicos.
	Services	Em sistemas Windows, são processos em segundo plano que fornecem funcionalidades essenciais, como gerenciamento de hardware e conectividade de rede.
	WMI	É uma estrutura em sistemas Windows que permite gerenciar dispositivos remotos e modificar suas configurações na rede.
	PowerShell	É uma ferramenta de linha de comando e script usada para gerenciar sistemas Windows, oferecendo acesso a recursos como arquivos, registro, processos, serviços e WMI.

Tabela 2 – Ferramentas e técnicas comuns usadas por agentes de ameaças para movimentação lateral.
Fonte: Adaptado de (PÉREZ-GOMARIZ; CERDÁN-CARTAGENA; GARCÍA, 2025).

2.3 Mineração de Processos

Mineração de Processos (Process Mining) usa técnicas de ciência de dados e análise de processos de negócios, para fornecer uma visão detalhada sobre a execução real dos processos dentro de uma organização. Diferentemente das abordagens tradicionais baseadas em modelos teóricos, a mineração de processos utiliza dados reais extraídos de registros de eventos armazenados em sistemas de informações,

como *logs* de atividades (GARCIA et al., 2019).

A mineração de processos consolidou-se como um campo de pesquisa promissor, dedicado à análise de processos com base em dados de eventos. Ao contrário de métodos clássicos de mineração de dados, como classificação, agrupamento, regressão, regras de associação e análise de sequências, que se limitam a examinar etapas individuais de um processo, a mineração de processos foca em compreender o ciclo completo de ponta a ponta. Esse progresso foi possibilitado pela maior acessibilidade a registros de eventos e pelo surgimento de tecnologias avançadas para descoberta de processos e verificação de conformidade (AALST, 2012).

As notações geradas pela mineração de processos neste estudo usam redes de Petri (THONG; AMEEDEEN, 2015) para representar o fluxo de trabalho dos registros. As redes de Petri permitem representar graficamente as interações entre os processos e eventos do sistema, destacando tanto o comportamento esperado quanto as alterações resultantes de ataques sofisticados. Esses diagramas expõem as diferenças entre um sistema operando em condições normais e um sistema impactado por atividades maliciosas.

Os elementos dos diagramas são apresentados por:

- **Ficha (Token):** Uma ficha representa um elemento ou recurso que transita pelo sistema. A ficha está associada a um lugar e indica o estado atual do sistema.

Figura 3 – Ficha

  - **Exemplo:** Em um processo de login, uma ficha pode representar um usuário autenticado. Quando um usuário insere suas credenciais (ação que ativa uma transição), a ficha pode passar de um lugar, como Aguardando Autenticação, para outro, como Autenticação Concluída.

- **Arco (Arc):** Um arco é a ligação entre um lugar e uma transição (ou vice-versa) que define como as fichas se movem no sistema. O arco especifica a relação de entrada ou saída entre os elementos.

Figura 4 – Arco

  - **Exemplo:** O arco conectando o lugar Aguardando Login à transição Verificar Credenciais mostra que a verificação das credenciais só pode ocorrer quando o lugar Aguardando Login tem pelo menos uma ficha.

- **Lugar (Place):** Um lugar simboliza um estado ou condição dentro do sistema. O lugar é frequentemente representado como um círculo e pode conter fichas que indicam que o sistema está naquele estado.

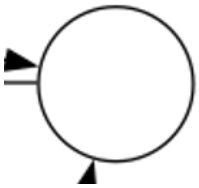

Figura 5 – Lugar

  - **Exemplo:** Aguardando Processamento pode ser um lugar em um sistema de processamento de dados. Quando as fichas estão ali, significa que os dados ainda estão aguardando ser processados.

- **Transição (Transition):** A transição é a ação ou evento que muda o estado do sistema, movendo as fichas entre os lugares. É representada como um retângulo ou barra nas redes de Petri.

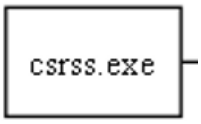

Figura 6 – Transição

  - **Exemplo:** No processo de um sistema, Processar Dados pode ser uma transição que consome fichas de Aguardando Processamento e as move para Processamento Concluído.

A crescente adoção da mineração de processos foi impulsionada pelo aumento na disponibilidade de dados digitais e pela evolução de tecnologias de coleta e análise de dados. Isso é particularmente relevante em ambientes corporativos onde sistemas ERP (Enterprise Resource Planning), CRM (Customer Relationship Management) e outros sistemas de informação documentam detalhadamente as atividades executadas. Esses registros permitem uma análise detalhada de atividades, desvios e gargalos (AALST, 2012). Além disso, técnicas de mineração de processos vão além da análise tradicional de dados ao permitir o estudo do comportamento sequencial e temporal das atividades. Ferramentas específicas, como ProM, Celonis e Disco, possibilitam a

criação de diagramas de fluxo que oferecem uma visão clara dos processos ponta a ponta, revelando lacunas que podem passar despercebidas em abordagens tradicionais (LEEMANS, 2018).

No contexto de transformações digitais, a mineração de processos desempenha um papel crucial na identificação de ineficiências, melhoria contínua e suporte à tomada de decisões baseada em dados. Mans et al. (2009) destacam que a mineração de processos pode revelar atividades redundantes, atrasos no fluxo de trabalho ou áreas de melhoria em processos críticos, promovendo ganhos de produtividade e qualidade.

Assim, a mineração de processos estabelece-se como uma abordagem robusta que não apenas analisa e descreve os processos existentes, mas também promove a conformidade, eficiência e inovação nos sistemas organizacionais.

2.4 Trabalhos Relacionados

O trabalho de (MACAK et al., 2022) apresenta uma revisão sistemática sobre a aplicação da mineração de processos em cibersegurança e confiabilidade de software, analisando 35 abordagens relevantes e identificando lacunas de pesquisa na área. Desse modo, o estudo destaca que, embora a mineração de processos tenha sido amplamente utilizada em setores como saúde, manufatura e logística, a sua aplicação em segurança cibernética ainda é limitada.

Outro ponto crítico identificado é que a maioria dos trabalhos analisados utilizam a mineração de processos para análises retrospectivas, ou seja, baseadas em eventos passados, sem suporte para monitoramento em tempo real. A introdução de mineração de processos em tempo real permitiria uma detecção precoce de ameaças, proporcionando respostas rápidas contra LM e outros ataques avançados. O estudo também enfatiza a necessidade de automação na detecção de padrões maliciosos e propões o uso de técnicas já aplicadas em outras áreas para aprimorar a segurança de sistemas críticos.

No intuito de poder usar os logs do sistema em tempo real, (SMILIOTOPOULOS; BARMPATSALOU ; KAMBOURAKIS, 2022) propõe a otimização da configuração do Sysmon para a identificação de LM no ecossistema MS Windows, estabelecendo critérios para determinar os recursos ideais de inicialização da ferramenta de monitoramento de eventos. Bem como, na utilização da base de conhecimento do MITRE ATT&CK para identificar os métodos mais comuns de LM. Ainda nesse sentido, (SMILIOTOPOULOS; KAMBOURAKIS; KOLIAS, 2024) apresenta uma revisão de literatura abrangente sobre a identificação de LM do ponto de vista do Sistema de Detecção de Intrusão.

No trabalho de (RODRíGUEZ; BETARTE; CALEGARI, 2024) é ressaltado a importância da coleta e análise de dados para entender as capacidades, intenções e comportamentos dos adversários cibernéticos com a finalidade de estabelecer medidas de segurança adequadas. Ainda assim, demonstra como a estrutura de conhecimento do MITRE ATT&CK é valiosa para entender detalhadamente as táticas, técnicas e procedimentos do invasor e nesse contento a utilização da mineração de processos pode ser usada para analisar eventos de tempo de execução de sistemas de informação, descobrindo assim relações causais entre esses eventos.

3 Metodologia

Este capítulo apresenta a proposta na seção 3.1 com uma breve descrição do método e a justificativa. Em seguinda, a seção 3.3 descreve o ambiente de testes configurado para a simulação dos ataques. Logo após, a seção 3.4 detalha as etapas do experimento. Por fim, a seção 3.5 discute as técnicas utilizadas para análise dos dados coletados.

O método adotado neste projeto é predominantemente quantitativa, focando na coleta e análise de dados numéricos para compreender e avaliar os fenômenos estudados. Essa abordagem é adequada ao propósito da deste trabalho, pois permite mensurar variáveis de maneira objetiva, identificar padrões e realizar comparações estatísticas que reforçam a confiabilidade dos resultados. Além disso, o método quantitativo possibilita a aplicação de testes e experimentos replicáveis, garantindo maior precisão e suporte empírico às conclusões obtidas no contexto do projeto (ECK et al., 2015)

3.1 Proposta

A cibersegurança tem se consolidado como uma área estratégica e com relevante importância para a proteção de empresas e instituições, especialmente diante do avanço das Ameaças Persistentes Avançadas. Essas ameaças representam um grande risco, pois exploram vulnerabilidades de forma sofisticada e prolongada, comprometendo redes e sistemas com extrema discrição. Nesse contexto, esta pesquisa se propõe a investigar a fase de movimentação lateral, uma etapa crítica no ciclo de vida das APTs. Por meio da aplicação de técnicas de mineração de processos, busca-se identificar padrões de comportamento malicioso, gerando conhecimento útil que podem fortalecer mecanismos de defesa e contribuir para a detecção precoce de ameaças que comprometem a integridade de redes corporativas.

3.1.1 Ferramentas Utilizadas

Nesta seção são apresentadas as principais ferramentas utilizadas no projeto, bem como suas características e funções. A escolha dessas ferramentas se baseia na necessidade de coletar, processar e analisar dados de eventos do sistema, de modo a identificar padrões de movimentação lateral em ataques APTs.

O Kali Linux[1] é uma distribuição baseada em Debian voltada para testes de

[1] https://www.kali.org/

intrusão e segurança ofensiva. Ele contém diversas ferramentas utilizadas para exploração de vulnerabilidades, movimentação lateral e escalonamento de privilégios. No projeto, o Kali Linux foi utilizado como máquina atacante para simular ações de um invasor, testando diferentes técnicas de movimentação lateral.

O pm4py[2] é uma biblioteca utilizada para mineração de processos, permitindo a análise e visualização de fluxos de eventos a partir de *logs* de execução. Essa ferramenta é essencial para identificar padrões de ataque e detectar desvios no comportamento esperado do sistema. A utilização do pm4py no projeto permite transformar os eventos coletados em modelos de processo, facilitando a análise das ações do atacante.

O Wazuh[3] é uma plataforma de segurança de código aberto utilizada para monitoramento de integridade de arquivos, detecção de ameaças, análise de *logs* e resposta a incidentes. Ele permite a coleta de eventos do sistema operacional, ajudando na detecção de atividades suspeitas. No contexto deste projeto, o Wazuh foi configurado para coletar *logs* das máquinas vítimas e do atacante, fornecendo uma visão detalhada das ações realizadas na rede.

3.1.2 Resumo das Ferramentas

A Tabela 3 apresenta um resumo das principais ferramentas utilizadas no projeto e suas funcionalidades.

Tabela 3 – Ferramentas utilizadas no projeto
Fonte: O autor.

Ferramenta	**Descrição**
Kali Linux	Sistema operacional utilizado para testes de intrusão e simulação de ataques.
Máquinas Virtuais Windows	Ambientes vítimas utilizados para captura de *logs* e execução de experimentos.
pm4py	Biblioteca para mineração de processos, utilizada na análise de movimentação lateral.
Wazuh	Plataforma de monitoramento de segurança, utilizada para coletar eventos e detectar ameaças.

Com essas ferramentas, foi possível criar um ambiente de testes controlado, permitindo a análise detalhada da movimentação lateral dentro da rede. Os dados coletados serão utilizados para modelar os ataques e identificar padrões de comportamento suspeitos.

[2] https://pypi.org/project/pm4py/
[3] https://wazuh.com/

3.2 Desenvolvimento

Nesta seção, serão apresentadas três classes fundamentais (WindowsProcessMonitor, ProcessMining e LogsToXes) para a conversão, análise e monitoramento de eventos no contexto da mineração de processos e segurança cibernética. Cada uma dessas ferramentas desempenha um papel específico na coleta e processamento de dados, contribuindo para a estruturação do pipeline de análise.

A Figura 7 apresenta as ferramentas que constituem a proposta deste trabalho.

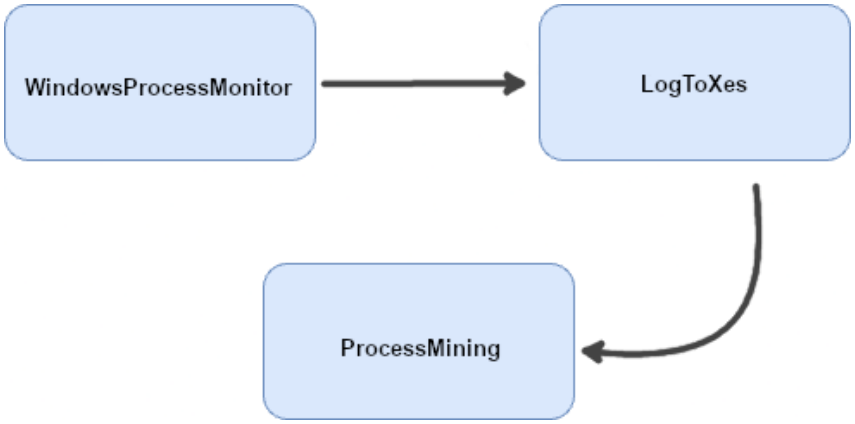

Figura 7 – Ferramentas desenvolvidas para a detecção da movimentação lateral.
Fonte: O autor.

3.2.1 WindowsProcessMonitor

O objetivo dessa ferramenta WindowsProcessMonitor (Lista 3.1) é monitorar e registrar eventos e processos do sistema de forma eficiente, garantindo a coleta organizada de dados relevantes para análises de segurança e auditorias, com uso de threads para execução simultânea. As etapas e explicação dessa classe:

- Importação das bibliotecas necessárias: O código utiliza bibliotecas importantes para captura e processamento de logs:
  - `win32evtlog`: Para acessar e filtrar logs de eventos do Windows.
  - `psutil`: Para monitorar processos em execução no sistema.
  - `time`: Para controle de execução e criação de timestamps.
  - `pandas`: Para manipular e salvar logs em formato CSV.
  - `signal` e `threading`: Para lidar com interrupções (Ctrl+C) e executar tarefas em threads paralelas.
- Definição dos eventos de interesse: Conjuntos são criados para identificar eventos específicos:

  - `SECURITY_EVENTS`: Eventos de segurança, como falhas de autenticação e escalada de privilégios.
  - `SMB_EVENTS`: Eventos relacionados a compartilhamento de arquivos na rede SMB.
  - `SSH_EVENTS`: Eventos associados ao protocolo SSH.
  - `ALL_EVENTS`: União de todos os conjuntos para capturar todos os eventos relevantes.

- Variáveis globais para coleta:
  - `windows_logs`: Lista que armazena eventos capturados do Windows.
  - `process_logs`: Lista que registra processos em execução no sistema.
  - `running`: Variável de controle para interromper a execução corretamente.

- Função `get_filtered_windows_logs`: Captura eventos do log de segurança do Windows:
  - Acessa o log de eventos do Windows usando o `win32evtlog`.
  - Filtra eventos de interesse definidos em `ALL_EVENTS`.
  - Armazena os dados no formato de dicionário, com informações como `EventID`, `TimeGenerated`, `SourceName` e mensagens relacionadas.

- Função `monitor_processes`: Monitora processos em execução no sistema:
  - Usa o `psutil.process_iter` para iterar sobre os processos ativos.
  - Captura informações como `pid`, `name` e `username` de cada processo.
  - Salva os dados em `process_logs` com timestamps.

- Função `save_logs`: Salva os logs capturados em arquivos CSV:
  - Transforma as listas `windows_logs` e `process_logs` em `pandas.DataFrame`.
  - Salva os dados em arquivos `filtered_windows_logs.csv` e `process_logs.csv`.

- Função `stop_execution`: Lida com o encerramento do programa:
  - Captura o sinal de interrupção (`Ctrl+C`).
  - Define a variável `running` como `False`, interrompendo os loops de coleta.

- Execução principal (`__main__`):
  - Associa o sinal `SIGINT` à função `stop_execution`.

- Cria e inicia duas threads para:
  - `get_filtered_windows_logs`: Captura eventos do Windows.
  - `monitor_processes`: Monitora processos em execução.
- Mantém o programa em execução até que `Ctrl+C` seja pressionado.
- Após a interrupção, chama a função `save_logs()` para salvar os dados coletados.

```
import win32evtlog
import psutil
import time
import pandas as pd
import signal
import threading

SECURITY_EVENTS = {4624, 4625, 4648, 4673, 4674, 4688, 4697, 4776}
SMB_EVENTS = {5140, 5145}
SSH_EVENTS = {22}
ALL_EVENTS = SECURITY_EVENTS | SMB_EVENTS | SSH_EVENTS

windows_logs = []
process_logs = []
running = True

def get_filtered_windows_logs(log_type="Security"):

    global running, windows_logs
    server = "localhost"
    flags = win32evtlog.EVENTLOG_BACKWARDS_READ | win32evtlog.
        EVENTLOG_SEQUENTIAL_READ
    log_handle = win32evtlog.OpenEventLog(server, log_type)

    try:
        while running:
            events_chunk = win32evtlog.ReadEventLog(log_handle, flags, 0)
            if not events_chunk:
                break
            for event in events_chunk:
                if event.EventID in ALL_EVENTS:
                    windows_logs.append({
                        "EventID": event.EventID,
                        "TimeGenerated": event.TimeGenerated.Format(),
                        "SourceName": event.SourceName,
                        "Message": event.StringInserts
                    })
            time.sleep(1)
```

```
38              print('coletando windows_logs')
39      except Exception as e:
40          print(f"Erro ao capturar eventos: {e}")
41      finally:
42          win32evtlog.CloseEventLog(log_handle)
43
44  def monitor_processes():
45      global running, process_logs
46      try:
47          while running:
48              for proc in psutil.process_iter(attrs=['pid', 'name', 'username
                    ']):
49                  if not running:
50                      break
51                  try:
52                      process_logs.append({
53                          "timestamp": time.time(),
54                          "process": proc.info['name'],
55                          "pid": proc.info['pid'],
56                          "user": proc.info['username']
57                      })
58                  except (psutil.NoSuchProcess, psutil.AccessDenied):
59                      continue
60              time.sleep(1)
61              print('coletando process_logs')
62      except Exception as e:
63          print(f"Erro ao monitorar processos: {e}")
64
65  def save_logs():
66      if windows_logs:
67          df_win = pd.DataFrame(windows_logs)
68          df_win.to_csv("filtered_windows_logs.csv", index=False)
69          print("Logs filtrados do Windows salvos.")
70
71      if process_logs:
72          df_proc = pd.DataFrame(process_logs)
73          df_proc.to_csv("process_logs.csv", index=False)
74          print("Logs de processos salvos.")
75
76  def stop_execution(signum, frame):
77      global running
78      print("\nParando a coleta de logs...")
79      running = False
80
81  if __name__ == "__main__":
82      signal.signal(signal.SIGINT, stop_execution)
83
```

```
84      t1 = threading.Thread(target=get_filtered_windows_logs, daemon=True)
85      t2 = threading.Thread(target=monitor_processes, daemon=True)
86
87      t1.start()
88      t2.start()
89
90      try:
91          while running:
92              time.sleep(1)
93      except KeyboardInterrupt:
94          stop_execution(None, None)
95
96      save_logs()
97      print("Coleta finalizada.")
```

Listing 3.1 – WindowsProcessMonitor
Fonte: O autor.

3.2.2 LogsToXes

O objetivo da ferramenta LogsToXes (Lista 3.2) é realizar a conversão de arquivos CSV para o formato XES, possibilitando sua utilização em mineração de processos. Ela assegura o ajuste e a organização dos dados, garantindo conformidade com os requisitos do formato XES e facilitando análises subsequentes. As etapas e explicação dessa classe:

- Importação de bibliotecas: As bibliotecas necessárias são carregadas para manipulação de arquivos, ajuste de dados e conversão de formatos.
- Definição da função `convert_csv_to_xes`: A função central responsável por realizar a conversão de um arquivo CSV para o formato XES, com as seguintes etapas:
  - Leitura do arquivo CSV: O arquivo CSV é carregado em um DataFrame para processamento posterior.
  - Verificação de arquivo vazio: Caso o arquivo CSV não possua dados, a função detecta e informa que o processo não pode continuar.
  - Identificação de colunas obrigatórias: Garante que as colunas essenciais para a conversão estejam presentes no DataFrame, como os campos de caso, tempo e evento.
  - Adiciona `case:concept:name` se ausente: Se a coluna correspondente a `case:concept:name` não existir, ela é criada e preenchida com valores apropriados.

  - Conversão de formatos e ajuste do DataFrame: Os dados no DataFrame são reorganizados e transformados em conformidade com os requisitos do formato XES.
  - Conversão do DataFrame para log de eventos: A estrutura do DataFrame é convertida em um log de eventos, que é compatível com o padrão XES.
  - Exportação do log para o formato XES: Finalmente, o log de eventos é exportado para um arquivo XES para ser usado em análises de mineração de processos.

- Chamada da função com exemplos de uso: A função é chamada com exemplos práticos de um arquivo CSV, demonstrando o funcionamento e gerando o arquivo XES correspondente.

```
import pandas as pd
from pm4py.objects.log.util import dataframe_utils
from pm4py.objects.conversion.log import converter as conversion_factory
from pm4py.objects.log.exporter.xes import exporter as xes_exporter

def convert_csv_to_xes(csv_filename, xes_filename):
    df = pd.read_csv(csv_filename)

    if df.empty:
        print(f"O arquivo {csv_filename} está vazio, não será convertido.")
        return

    if "EventID" in df.columns:
        df["concept:name"] = df["EventID"].astype(str)
        df["time:timestamp"] = pd.to_datetime(df["TimeGenerated"])

    elif "process" in df.columns:
        df["concept:name"] = df["process"]
        df["time:timestamp"] = pd.to_datetime(df["timestamp"], unit="s")

    else:
        print(f"Erro: Arquivo {csv_filename} não possui colunas
            reconhecidas!")
        return

    if 'case:concept:name' not in df.columns:
        df['case:concept:name'] = 'default_case'

    df = dataframe_utils.convert_timestamp_columns_in_df(df)
    df = df[["case:concept:name", "concept:name", "time:timestamp"]]

    event_log = conversion_factory.apply(df)
```

```
32
33      xes_exporter.apply(event_log, xes_filename)
34      print(f"Arquivo {xes_filename} salvo com sucesso!")
35
36  convert_csv_to_xes("filtered_windows_logs.csv", "filtered_windows_logs.xes"
        )
37  convert_csv_to_xes("process_logs.csv", "process_logs.xes")
```

Listing 3.2 – LogsToXes
Fonte: O autor.

3.2.3 ProcessMining

O objetivo da ferramenta ProcessMining (Lista 3.3) é processar logs em formato XES para descobrir e visualizar modelos de processos, representados como redes de Petri, além de calcular a conformidade entre os logs e os modelos gerados. Isso permite análises detalhadas e avaliação da aderência do comportamento real ao modelo identificado. As etapas e explicação dessa classe:

- Importação dos módulos necessários: Os módulos do pacote `pm4py` são importados para permitir as operações relacionadas à análise e descoberta de processos:
  - `xes_importer`: Para carregar arquivos no formato XES.
  - `alpha_miner`: Para descobrir redes de Petri a partir dos logs.
  - `pn_visualizer`: Para visualizar redes de Petri como gráficos.
  - `token_replay`: Para calcular a conformidade do log com a rede de Petri descoberta.
- Definição da lista de nomes de logs: A variável `logs_names` contém os nomes dos arquivos XES a serem processados, como `process_logs` e `filtered_windows_logs`.
- Configuração dos parâmetros de visualização: Um dicionário chamado `parameters_visualization` é criado para definir o formato da saída (`png`), o mecanismo de renderização (`dot`) e a orientação (`portrait`) da visualização.
- Iteração sobre os nomes dos logs: O código processa cada log listado em `logs_names`. Para cada arquivo:
  - Carregamento do log XES: O log é carregado usando o método `apply` do `xes_importer`, e o número de eventos é exibido no console.

- Descoberta da rede de Petri com Alpha Miner: A rede de Petri é gerada a partir do log. O resultado inclui a estrutura da rede (`net`), a marcação inicial (`initial_marking`) e a marcação final (`final_marking`).
- Visualização do modelo descoberto: A rede de Petri é visualizada usando o módulo `pn_visualizer`. Parâmetros adicionais ajustam a orientação para retrato (`rankdir` = 'TB'), e o gráfico é exibido diretamente na tela.
- Salvando a visualização como arquivo PNG: A visualização é salva em um arquivo nomeado com o mesmo prefixo do log e com a extensão `png`.
- Cálculo da conformidade com token replay: A conformidade entre o log e a rede de Petri é avaliada usando `token_replay`. O resultado, que inclui informações sobre o nível de conformidade, é armazenado em `replayed_traces`.
- Exportação dos resultados de conformidade: O resultado da conformidade (`replayed_traces`) é escrito em um arquivo de texto chamado `{logN}_conformidade.txt`.

```
1  from pm4py.objects.log.importer.xes import importer as xes_importer
2  from pm4py.algo.discovery.alpha import algorithm as alpha_miner
3  from pm4py.visualization.petri_net import visualizer as pn_visualizer
4  from pm4py.algo.conformance.tokenreplay import algorithm as token_replay
5
6  logs_names = ['process_logs', 'filtered_windows_logs']
7
8  parameters_visualization = {
9      "format": "png",
10     "engine": "dot",
11     "orientation": "portrait"
12 }
13
14 for logN in logs_names:
15     log = xes_importer.apply(f"{logN}.xes")
16     print(f"Log␣carregado␣com␣{len(log)}␣eventos!")
17
18     net, initial_marking, final_marking = alpha_miner.apply(log)
19
20     gviz = pn_visualizer.apply(net, initial_marking, final_marking,
           parameters=parameters_visualization)
21
22     gviz.graph_attr['rankdir'] = 'TB'
23     pn_visualizer.view(gviz)
24
25     gviz.render(filename=f'{logN}_output', format='png')
26
27     replayed_traces = token_replay.apply(log, net, initial_marking,
           final_marking)
```

```
28
29      with open(f'{logN}_conformidade.txt', 'w') as arquivo:
30          arquivo.write(str(replayed_traces))
```

Listing 3.3 – ProcessMining
Fonte: O autor.

3.2.4 Experimento

A pesquisa adota um modelo experimental, estruturado nas seguintes etapas principais:

1. **Configuração do Ambiente de Testes**: Configurar um ambiente virtualizado, composto por máquinas virtuais Windows conectadas em uma subrede controlada, associadas a um atacante operando a partir de um sistema Kali Linux.
2. **Simulação de Cenários de Ataque**: Realizar a simulação de APTs, com foco na movimentação lateral entre máquinas comprometidas, para replicar situações reais de invasão.
3. **Coleta e Monitoramento de Dados**: Registrar as atividades das máquinas comprometidas utilizando ferramentas específicas, como o Wazuh, para monitoramento de logs de eventos.
4. **Mineração e Análise de Dados**: Utilizar a biblioteca **pm4py** para realizar a mineração de processos e extrair padrões de comportamento relevantes.
5. **Validação dos Resultados**: Comparar os padrões de comportamentos, avaliando a viabilidade da detecção de comportamentos maliciosos e anômalos.

3.3 Cenários

O ambiente de testes foi configurado para simular um cenário de ataque APT, com foco na fase de movimentação lateral. A Figura 8 apresenta a visão geral do cenário do experimento.

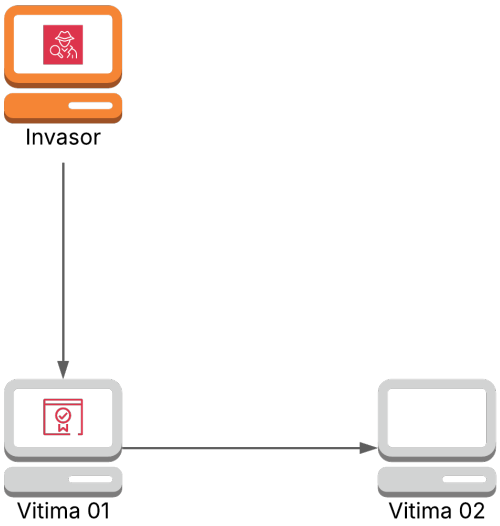

Figura 8 – Ambiente de testes do primeiro cenário
Fonte: O autor.

A infraestrutura consiste em máquinas virtuais organizadas dentro de uma sub-rede privada (192.168.2.0/24), sendo:

- Kali Linux (Atacante): 192.168.2.5
  - Sistema Operacional: Kali Linux
  - Memória RAM: 6 GB
  - Processador: 12th Gen Intel(R) Core(TM) i3-1215U, 4 núcleos
  - Armazenamento: 50 GB
- Windows 1 (Cenário 1): 192.168.2.6
  - Sistema Operacional: Windows 10
  - Memória RAM: 4 GB
  - Processador: 12th Gen Intel(R) Core(TM) i3-1215U, 2 núcleos
  - Armazenamento: 50 GB
- Windows 2 (Cenário 2): 192.168.2.7
  - Sistema Operacional: Windows 10
  - Memória RAM: 4 GB
  - Processador: 12th Gen Intel(R) Core(TM) i3-1215U, 2 núcleos
  - Armazenamento: 50 GB

As máquinas Windows foram configuradas com usuários padrão e serviços essenciais ao funcionamento do ambiente, o que, dentro de um cenário realista, pode ser explorado por um agente mal-intencionado para a movimentação lateral. Cada sistema conta com um diretório compartilhado, denominado espaco n (onde n representa o número do cenário), refletindo uma configuração comum em redes corporativas para o compartilhamento interno de arquivos.

Para realizar o experimento, foi executada uma simulação do uso das maquinas vítimas, sendo: navegação na internet, *download* de documentos, criação e edição de arquivos. Considerando assim o comportamento normal esperado em situações reais. Essa simulação foi conduzida em dois cenários distintos: o primeiro, sem qualquer ocorrência de movimentação lateral, e o segundo, contemplando a presença da movimentação lateral provocada pelo ataque.

Durante o desenvolvimento do ataque, após o invasor obter o acesso inicial à primeira máquina, realizou o reconhecimento da infraestrutura da rede. Por meio desse reconhecimento, foi possível mapear e identificar recursos compartilhados ou acessíveis no ambiente. Aproveitando permissões existentes na rede, o atacante estabeleceu uma conexão com a segunda máquina. Essa etapa permitiu não somente o envio de um *payload* malicioso, mas também sua execução remota. Como resultado, o invasor consolidou sua presença no sistema, criando um ponto de apoio estratégico que possibilitou o avanço do ataque e sua propagação pelos recursos comprometidos da rede, apresentado na Figura 9.

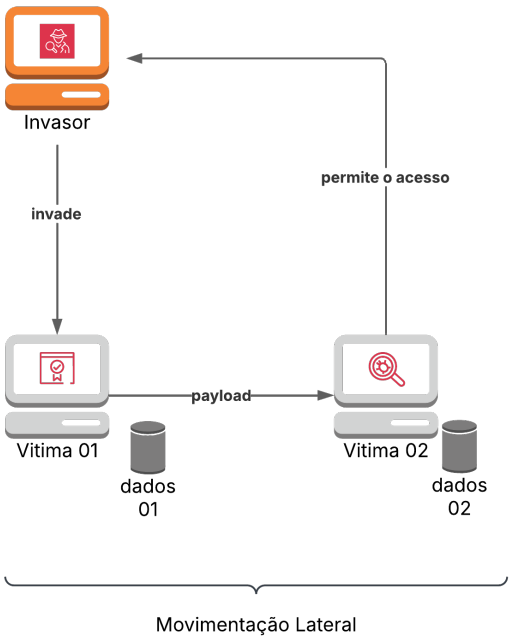

Figura 9 – Ambiente com movimentação lateral
Fonte: O autor.

Os eventos do sistema operacional são monitorados por meio de *Event Logs* do Windows junto com o Wazuh, permitindo a captura de registros detalhados das atividades do sistema. Além disso, foram desenvolvidos *scripts* em Python para monitorar, coletar e processar esses eventos, assegurando a extração estruturada das informações relevantes. Esses dados são então preparados e aplicados em técnicas de mineração de processos, viabilizando a identificação de padrões e possíveis anomalias no comportamento do sistema durante o experimento.

3.4 Etapas do Experimento

O experimento foi estruturado em etapas definidas para garantir a organização e a reprodutibilidade dos procedimentos. Cada etapa é planejada para realizar a configuração do ambiente, executar o ataque simulado e coletar dados para análise posterior. Em seguida, são descritas as principais etapas:

1. **Configuração do ambiente:** O primeiro passo consistiu na preparação do ambiente de teste, garantindo que todas as máquinas virtuais (VMs) fossem configuradas de uma forma funcional. As tarefas incluíram a instalação e configuração inicial das VMs, a integração das máquinas Windows ao sistema de monitoramento Wazuh e a definição de permissões específicas e compartilhamentos de arquivos para simular um ambiente corporativo.

    - Instalação e configuração das VMs: Criação e ajuste das máquinas virtuais com base nos requisitos do experimento.
    - Integração das máquinas virtuais Windows ao Wazuh: Conexão ao sistema para captura de eventos e monitoramento.
    - Definição das permissões e compartilhamentos de arquivos: Configuração de políticas de acesso para simular um ambiente corporativo.

2. **Implementação de *script* para realizar a coleta:** Um *script* foi desenvolvido para capturar os dados relevantes durante as simulações. Este *script* opera com duas *threads*, permitindo a coleta simultânea de logs de processos e logs de segurança, otimizando o processo de monitoramento.

    - *Script* atua com duas *threads*: Permite a coleta simultânea de logs de processos e logs de segurança, otimizando o desempenho e eficiência do monitoramento.
    - Captura dos logs de processos e logs de segurança: Coleta de informações importantes para análise futura.

3. **Coleta de dados do uso normal do sistema:** Antes da execução do ataque, foi realizada uma etapa de monitoramento das atividades normais dos usuários nas máquinas-alvo. Nesta etapa, foram simuladas ações comuns e rotineiras realizadas em um ambiente típico, com o objetivo de capturar os *logs* gerados sem interferência de ameaças. Essas atividades forneceram uma base de comparação para avaliar os impactos do ataque.

    - Monitoramento de eventos durante atividades normais dos usuários: Incluiu navegação na internet, *download* de documentos e criação e edição de arquivos, representando cenários reais de uso.
    - Registro de *logs* sem interferência de ataques: Geração de um conjunto de dados que reflete o comportamento legítimo do sistema, garantindo a precisão na análise comparativa.

4. **Execução do ataque APT:** Nesta etapa, foi simulado um ataque de ameaça persistente avançada (APT). O atacante, utilizando a máquina Kali Linux, comprometeu o primeiro cenário (Windows 1) e explorou vulnerabilidades para acessar o segundo cenário (Windows 2). O ataque incluiu a execução de um *payload* malicioso via compartilhamento de arquivos.

    - Invasão do primeiro cenário (192.168.2.6) a partir do Kali Linux: Acesso inicial ao sistema vulnerável.
    - Exploração de vulnerabilidades para acessar o segundo cenário (192.168.2.7): Propagação do ataque.
    - Execução de *payload* no segundo cenário via compartilhamento de arquivos: Execução do código malicioso para comprometer o sistema.

5. **Coleta de dados pós-ataque:** Após o ataque, foram coletados os logs para análise da movimentação lateral e dos eventos registrados pelo Wazuh e pelo Windows Event Logs. Esses dados são fundamentais para identificar os rastros deixados pelo atacante e os impactos do ataque.

    - Registro de eventos relacionados à movimentação lateral: Identificação do progresso do atacante entre as máquinas.
    - Extração de logs do Wazuh e do *Windows Event Logs*: Coleta dos dados gerados durante o ataque.

6. **Desenvolvimento de *script* para preparação dos dados para mineração de processos:** Finalmente, os logs coletados foram processados por meio de um *script* para limpeza, formatação e conversão para um formato adequado à ferramenta pm4py. Essa etapa preparou os dados para análises mais detalhadas.

- Limpeza e formatação dos *logs*: Remoção de informações redundantes ou irrelevantes.
- Conversão para um formato adequado para o pm4py: Ajuste dos dados para compatibilidade com a ferramenta.
- Geração de *logs* de eventos para análise: Produção de um conjunto organizado de dados para mineração de processos.

3.5 Técnicas de Análise

A análise dos dados coletados foi realizada com base em um método estruturado que envolve múltiplos estágios, desde o pré-processamento até a interpretação dos resultados. O uso da biblioteca pm4py, reconhecida por suas capacidades de mineração de processos, foi essencial para identificar padrões comportamentais e anomalias relacionadas aos ataques simulados. Cada etapa é descrita a seguir:

- **Pré-processamento dos Dados:** Esta etapa preparou os *logs* brutos para serem analisados pela ferramenta pm4py. O objetivo foi estruturar os dados de maneira compatível.
  - Extração de *logs* do Windows (*Event Logs* e Wazuh): Recuperação dos eventos pertinentes ao experimento.
  - Filtragem de eventos relacionados a movimentação lateral: Identificação de ocorrências ligadas à propagação do ataque entre máquinas.
  - Conversão dos dados para um formato compatível com o pm4py: Transformação dos *logs* em um modelo processável pela ferramenta.
- **Aplicação de algoritmos de mineração de processos:** Com os dados pré-processados, foi utilizado a biblioteca pm4py para mapear e entender os fluxos de eventos registrados. Essa análise busca identificar padrões e desvios comportamentais durante o experimento.
  - Utilização do *Alpha Miner* para descoberta de processos: Aplicação do algoritmo para modelar o comportamento registrado nos *logs*.
  - Avaliação de padrões comportamentais antes e depois do ataque: Comparação entre o uso normal do sistema e os fluxos registrados durante o ataque.
  - Comparação entre *logs* de uso normal e de movimentação lateral: Identificação das diferenças entre os cenários com e sem ataque.

- **Interpretação dos resultados:** Os resultados obtidos pela análise foram avaliados para compreender os impactos do ataque e validar a eficácia da abordagem. Isso incluiu tanto a detecção de anomalias quanto a comparação com métodos tradicionais.
  - Identificação de desvios nos fluxos de eventos: Reconhecimento de alterações no comportamento padrão do sistema, indicando atividades maliciosas.
  - Detecção de padrões anômalos indicativos de ataques APT: Identificação de características específicas relacionadas às ações do invasor.

Com essa abordagem, espera-se demonstrar como a mineração de processos pode ser aplicada na identificação de ataques APT, especificamente na fase de movimentação lateral.

4 Resultados

Este capítulo apresenta e analisa os principais resultados obtidos a partir das etapas realizadas no experimento. Além disso, interpretar os dados coletados, identificando padrões comportamentais e anomalias que surgiram durante as simulações e ataques. Adicionalmente, são avaliadas as implicações desses resultados no contexto de segurança e monitoramento em sistemas Windows. As abordagens analíticas aplicadas forneceram *insights*, contribuindo para um melhor entendimento do impacto das ameaças e da eficácia das técnicas de detecção utilizadas.

4.1 Apresentação dos resultados da mineração de processos

A mineração de processos em ambiente Windows gerou resultados por meio da análise dos logs de eventos e processos do sistema. Essa análise permite a extração de dados essenciais, representados em diagramas para visualizar as interações complexas entre os diversos componentes do sistema operacional.

Nos apêndices A, B, C e D, são apresentados os diagramas de fluxo, modelados utilizando redes de Petri, gerados para diferentes cenários: com e sem a presença de Ameaça Persistente Avançada. Esses diagramas expõem as diferenças entre um sistema operando em condições normais e um sistema impactado por ataques sofisticados.

Ao longo desse processo, foi possível identificar padrões recorrentes que refletem o funcionamento esperado do sistema, servindo como referência para comportamentos normais. Além disso, a análise revelou anomalias que, ao serem destacadas nos diagramas, levantaram suspeitas sobre possíveis comportamentos maliciosos ou atividades irregulares no sistema. Tais anomalias podem incluir acessos não autorizados, execução de processos fora do padrão ou outras irregularidades que comprometam a segurança e a integridade do ambiente.

Essa abordagem baseada na mineração de processos não só viabiliza a compreensão das dinâmicas internas do sistema, como também se apresenta como uma ferramenta relevante para a detecção de ameaças e a otimização do desempenho em ambientes Windows. Isso reflete a importância de métodos analíticos avançados na gestão de sistemas complexos, proporcionando *insights* valiosos para profissionais de tecnologia e segurança da informação.

4.1.1 Características dos *Logs* sem Movimentação Lateral

O Apêndice A apresenta o fluxo de trabalho registrado no primeiro cenário, onde não foi detectada a presença de uma Ameaça Persistente Avançada (APT). O modelo reflete o comportamento típico dos eventos capturados. Os principais processos registrados incluem acessos rotineiros e interações previsíveis, sem desvios anômalos ou sinais de interferência externa. O diagrama ilustra uma interação simples e previsível entre processos do sistema.

O Apêndice B apresenta o fluxo de trabalho registrado no cenário, onde também não foi detectada a presença de uma Ameaça Persistente Avançada (APT). O modelo reflete interações lineares entre processos, similares ao registrado no Apêndice A, destacando a consistência no comportamento esperado em um ambiente seguro. O gráfico exibe um padrão rotineiro de eventos, sem sinais de processos ou acessos suspeitos.

Nos casos sem movimentação lateral, apresentados nas Apêndices A e B, os diagramas revelam:

- **Interações simples**. Fluxos lineares e previsíveis com um número reduzido de nós e conexões.
- **Padrões consistentes**. Eventos como execução de processos do sistema (`explorer.exe` e `svchost.exe`) são predominantes.
- **Ausência de anomalias**. Nenhum comportamento anormal foi identificado, indicando a operação típica do sistema.

Essas características refletem a estabilidade do sistema em condições normais, oferecendo um padrão confiável para comparação com cenários anômalos. Por outro lado, os *logs* sem movimentação lateral, apresentaram uma redução significativa de eventos suspeitos. Os processos registrados eram predominantemente de natureza rotineira, como:

- **Execução de Processos do Sistema**. Processos conhecidos, como `explorer.exe`, `svchost.exe` e outros diretamente associados ao funcionamento normal do Windows.
- **Interações Simples e Previsíveis**. Diagramas indicaram que as relações entre os processos eram mais estáveis, com poucas conexões complexas ou ramificações, típicas de um sistema que opera em condições normais.

A Figura 10 apresenta os *logs* de segurança filtrados no ambiente monitorado, considerando a ausência de movimentação lateral. Esse diagrama ilustra a atividade

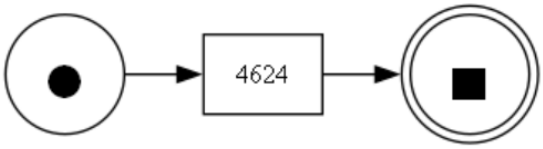

Figura 10 – Windows 1 e 2 sem *LM*.
Fonte: O autor.

registrada nos dois cenários distintos, correspondentes às máquinas Windows 1 e Windows 2, destacando o evento 4624, que representa um login bem-sucedido de um usuário legítimo, sem envolvimento em acessos remotos ou atividades maliciosas.

Essa distinção entre logs com e sem LM reflete diretamente na eficácia da mineração de processos para destacar padrões relevantes, ajudando a diferenciar o tráfego legítimo de atividades potencialmente maliciosas.

Os diagramas gerados permitiram observar sequências específicas que sugerem tentativas coordenadas de ataque. Por exemplo, a combinação de eventos de logon bem-sucedidos, seguida de criação de processos, aponta para um fluxo típico de movimentação lateral dentro de uma rede. Além disso, nos logs com LM, foi detectada a presença de processos suspeitos que não são comumente utilizados em ambientes operacionais normais, como `psexec.exe`, frequentemente associado a movimentação lateral em ataques reais.

4.1.2 Características dos *Logs* com movimentação lateral

Nos casos de movimentação lateral, como representado nas Apêndices C e D, os diagramas mostram:

O Apêndice C apresenta o fluxo de trabalho do primeiro cenário, impactada pela presença de uma Ameaça Persistente Avançada (APT). O modelo evidencia anomalias, incluindo interações complexas entre processos e a presença de atividades suspeitas que sugerem tentativas de exploração do sistema.O diagrama destaca atividades anômalas e ciclos não convencionais que indicam comportamento malicioso.

O Apêndice D apresenta o fluxo de trabalho registrado no segundo cenário, onde houve a presença de uma Ameaça Persistente Avançada (APT). O modelo reflete conexões e padrões de comportamento similares ao observado no Cenário 01 (Apêndice C), com múltiplos processos anômalos e interações não esperadas. O diagrama reflete eventos complexos e interações extensivas sugerindo movimentação lateral e exploração do sistema.

- **Eventos críticos**. Como mostrado, houve um aumento significativo de eventos como Logon Falhado (4625) e Criação de Processos (4688), destacando tentati-

vas de força bruta e execução de *scripts* maliciosos.

- **Caminhos intricados**. Os fluxos apresentam conexões entre nós que fogem dos padrões esperados, como ciclos envolvendo processos específicos.
- **Atividades suspeitas**. Wmi identificado no Apêndice D sugere comportamentos avançados de invasores em ambientes corporativos.

As Figuras 11 e 12 apresentam os *logs* de segurança coletados e filtrados, destacando eventos críticos para a análise da movimentação lateral. Na primeira máquina Windows, o evento 4648 indica o uso explícito de credenciais para autenticação em um recurso remoto, sugerindo uma tentativa de acesso entre sistemas. Já na segunda máquina Windows, o evento 4688 registra a criação de um novo processo, evidenciando a execução de comandos ou programas possivelmente vinculados à fase de exploração do ataque.

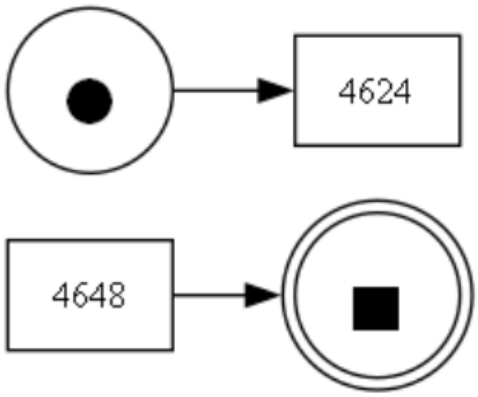

Figura 11 – Windows 1 com *LM*
Fonte: O autor.

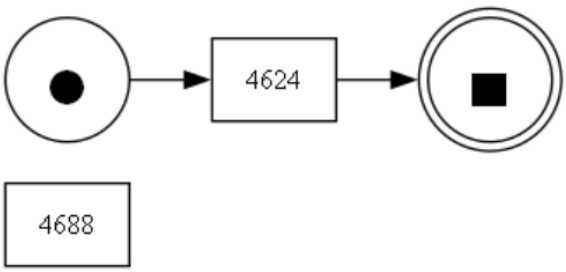

Figura 12 – Windows 2 com *LM*
Fonte: O autor.

Essas características são comuns em ataques direcionados, reforçando a necessidade de análise contínua para proteger o ambiente.

4.2 Comparação entre o tráfego normal e os ataques detectados

Os diagramas apresentados demonstram as diferenças entre tráfego normal e tráfego associado a movimentação lateral. Os apêndices C e D destacam como as ameaças cibernéticas afetam a complexidade do fluxo do sistema, enquanto as Figuras A e B mostram um comportamento previsível e seguro.

A mineração de processos demonstrou ser uma abordagem poderosa para diferenciar comportamentos normais do sistema e tráfego anômalo associado a ataques cibernéticos. Os logs com movimentação lateral revelaram padrões de tráfego que indicam tentativas de explorar o sistema, enquanto os logs sem movimentação lateral apresentaram fluxos previsíveis e confiáveis.

4.2.1 *Logs* sem Movimentação Lateral

Por outro lado, os logs sem movimentação lateral apresentaram poucos eventos críticos e nenhuma sequência que pudesse ser interpretada como uma tentativa coordenada de ataque. Os processos registrados eram amplamente rotineiros e previsíveis, refletindo as operações normais do sistema operacional.

Essa comparação destaca a eficácia do monitoramento contínuo e da mineração de processos como ferramentas de detecção de padrões que diferem do comportamento esperado, permitindo que atividades suspeitas sejam identificadas em tempo hábil.

4.2.2 *Logs* com Movimentação Lateral

Os logs filtrados para movimentação lateral indicaram uma maior ocorrência de eventos como 4624 (logon bem-sucedido) e 4625 (falha de logon), muitas vezes em sucessão rápida, sugerindo ataques de força bruta ou exploração de vulnerabilidades. Além disso, a criação de processos (4688) e o uso explícito de credenciais (4648) indicaram atividades avançadas por parte de invasores, como a execução remota de comandos utilizando ferramentas administrativas.

4.3 Interpretação dos padrões descobertos

A análise dos padrões identificados nos logs demonstrou a importância de uma abordagem visual e baseada em dados para a segurança cibernética. Os diagramas gerados pela mineração de processos foram essenciais para mapear as interações entre os elementos do sistema, destacando as etapas típicas de movimentação lateral.

4.3.1 Padrões de Movimentação Lateral

Os padrões observados em logs com movimentação lateral incluíram:

1. Tentativas de Logon Repetidas: Eventos 4625, indicando tentativas falhas de acesso, muitas vezes seguidas por logons bem-sucedidos (4624), sugerem ataques de força bruta.
2. Criação de Novos Processos: Eventos 4688, especialmente associados a processos incomuns, indicam tentativas de invasores de executar comandos ou ferramentas administrativas.
3. Uso Explícito de Credenciais: Eventos 4648 evidenciam o uso de credenciais roubadas para acessar recursos adicionais.

Esses padrões refletem as etapas comuns de um ataque cibernético, desde o reconhecimento até a exploração e a escalada de privilégios dentro de uma rede corporativa.

4.3.2 Padrões de Comportamento Normal

Nos logs sem movimentação lateral, os padrões identificados incluíam interações simples e previsíveis entre processos rotineiros do sistema, como `explorer.exe` e `svchost.exe`. Esses padrões servem como referência para diferenciar o comportamento legítimo de atividades suspeitas.

5 Conclusão e Trabalhos Futuros

Este capítulo apresenta um resumo dos principais achados do estudo, discute a relevância dos resultados obtidos para a detecção de Ameaças Persistentes Avançadas (APTs) e sugere direções para futuras pesquisas na área de Cyber Threat Intelligence.

O estudo abordou a detecção de APTs através de Mineração de Processos, com ênfase na fase de movimentação lateral. Para isso, foi desenvolvido um ambiente de testes contendo máquinas virtuais simulando um ataque real, no qual o atacante comprometia um dispositivo inicial e, a partir desse dispositivo, se deslocava para outros dispositivos na rede.

As contribuições incluem:

- Viabilidade de coletar e correlacionar logs de eventos do sistema para identificar padrões de comportamento.
- Uso da Mineração de Processos na reconstrução do fluxo de atividades e na distinção entre operações normais e maliciosas.
- A possibilidade de automação da detecção de ataques por meio da modelagem de processos do ambiente comprometido.

Esses achados reforçam a necessidade de abordagens baseadas em análise de processos para melhorar a detecção de APTs, que tradicionalmente são difíceis de identificar devido à sua natureza furtiva e prolongada.

A detecção de APTs continua sendo um desafio crítico para a segurança cibernética, dado que esses ataques são meticulosamente planejados e executados com técnicas avançadas para evitar a identificação. Os resultados deste estudo demonstram que a mineração de processos pode ser uma ferramenta eficaz na compreensão dos fluxos de atividade em um ambiente comprometido, permitindo a identificação de desvios e comportamentos atípicos.

O uso de logs do sistema e eventos coletados pelo Wazuh mostrou-se crucial para a análise, destacando a importância de uma coleta de dados robusta para alimentar algoritmos de detecção. Além disso, a possibilidade de automatizar a identificação de padrões de movimentação lateral demonstra o potencial de aplicações práticas dessa metodologia em ambientes reais.

Os resultados também apontam para a necessidade de integração entre Mineração de Processos e outras técnicas de detecção baseadas em aprendizado de máquina e análise comportamental, a fim de criar sistemas de segurança mais robustos e capazes de identificar ameaças em tempo hábil.

5.1 Limitações do estudo

Embora os resultados apresentados sejam significativos, algumas limitações devem ser destacadas:

1. **Coleta de Dados**: A representatividade dos dados coletados foi limitada ao ambiente de teste, o que pode não refletir a complexidade de redes corporativas maiores e mais diversificadas.
2. **Ferramentas Utilizadas**: As ferramentas de mineração de processos possuem limitações inerentes, como capacidade de processamento de grandes volumes de dados e funcionalidades restritas para análise avançada.
3. **Análise Contextual**: A ausência de uma contextualização mais ampla dos resultados com as políticas e práticas específicas da organização pode dificultar a aplicação prática das recomendações.

Este estudo demonstrou o potencial da mineração de processos como uma abordagem para monitorar e analisar atividades em ambiente Windows. A capacidade de identificar padrões de movimentação lateral e tráfego anômalo destaca a relevância desta técnica para a segurança cibernética. No entanto, as limitações do estudo sugerem a necessidade de aprimoramentos na coleta de dados, ferramentas e análise, para ampliar a aplicabilidade e eficácia da abordagem em diferentes contextos.

5.2 Trabalhos Futuros

Embora os resultados obtidos sejam promissores, há diversas possibilidades para avançar nessa linha de pesquisa. Algumas sugestões incluem:

- **Expansão do conjunto de dados**: A coleta de mais dados de diferentes cenários e ambientes pode ajudar a validar e refinar os modelos utilizados na detecção de APTs.
- **Aplicabilidade em ambientes reais**: Testar a abordagem proposta em ambientes empresariais reais permitiria avaliar sua eficácia contra ataques autênticos.

- **Integração com aprendizado de máquina**: A exploração de modelos de aprendizado supervisionado e não supervisionado poderia melhorar a capacidade de detecção e reduzir falsos positivos.
- **Análise de outras fases do ataque**: Embora este estudo tenha focado na movimentação lateral, outras fases do ciclo de vida de um APT, como a exfiltração de dados, também podem se beneficiar da Mineração de Processos.
- **Automatização da resposta a incidentes**: Explorar maneiras de integrar a detecção baseada em Mineração de Processos com respostas automáticas a incidentes pode aumentar a eficiência na mitigação de ameaças.

Com essas direções, espera-se que futuras pesquisas possam contribuir para a evolução das estratégias de detecção e resposta a APTs, tornando os sistemas mais resilientes contra esse tipo de ameaça sofisticada.

A Cenário 01 sem APT

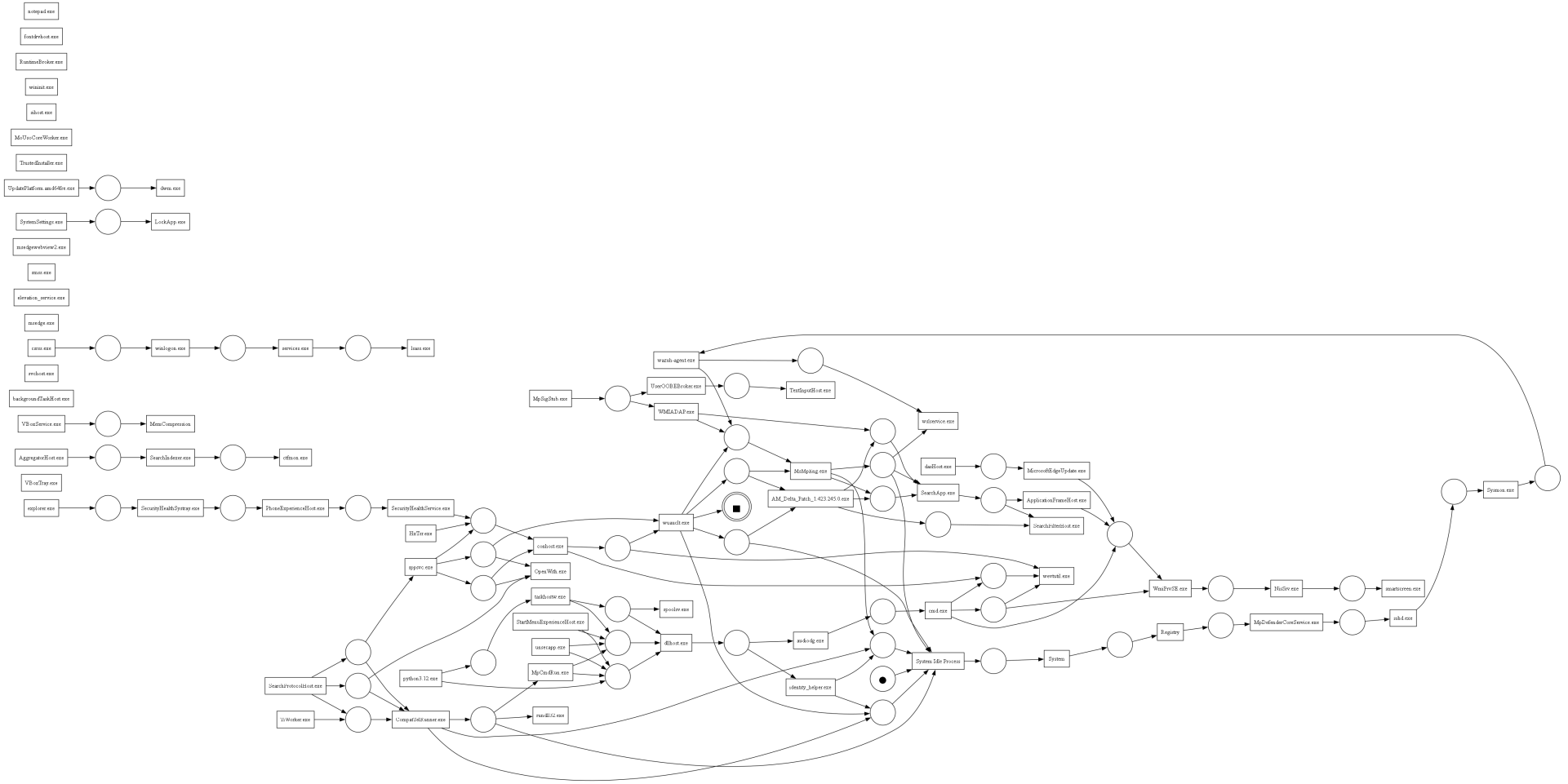

Figura 13 – Fluxo de trabalho da Vítima 01 sem APT.
Fonte: O autor.

B Cenário 02 sem APT

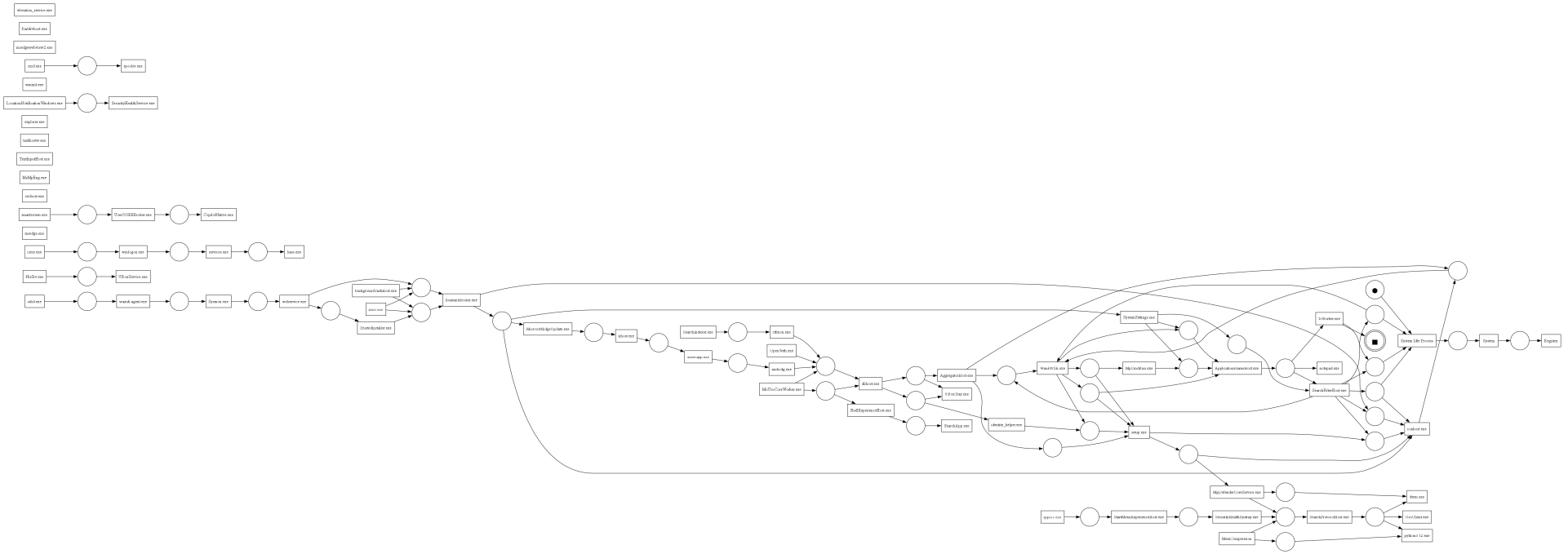

Figura 14 – Fluxo de trabalho da Vítima 02 sem APT.
Fonte: O autor.

C Cenário 01 com APT

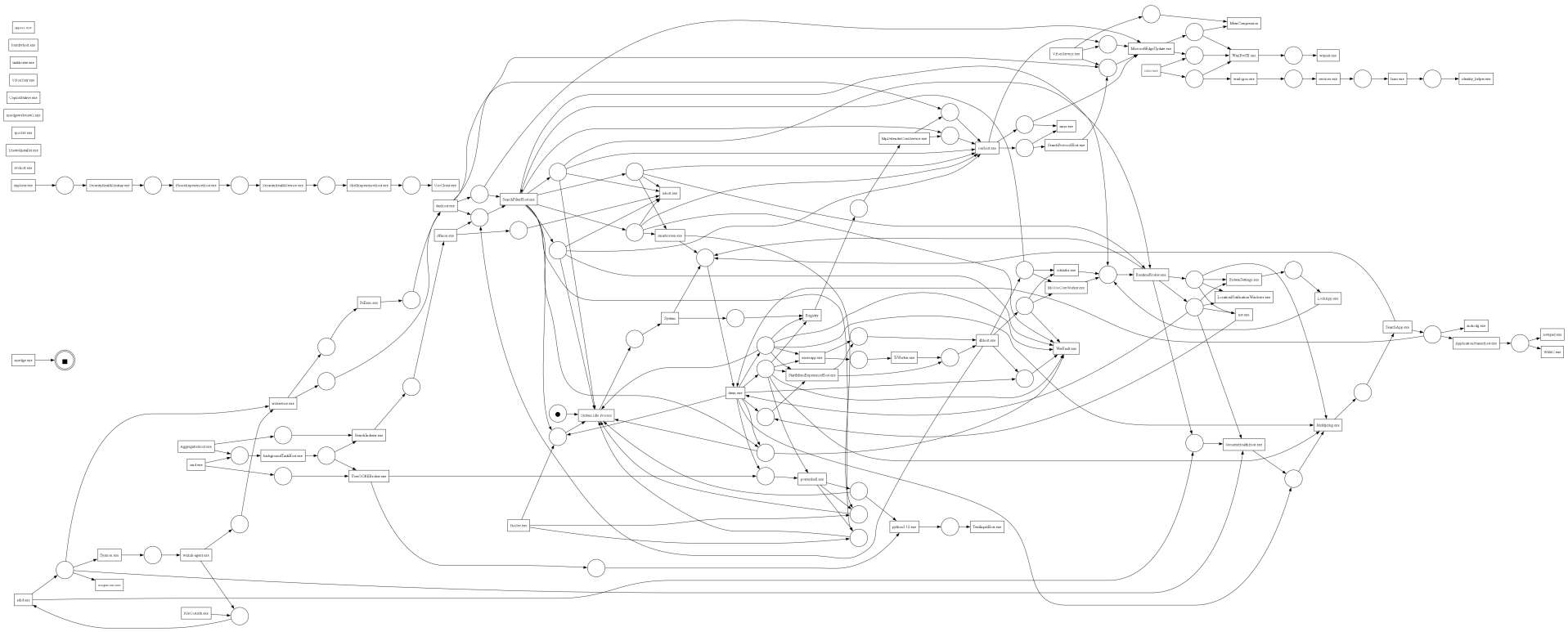

Figura 15 – Fluxo de trabalho da Vítima 01 impactado por APT.
Fonte: O autor.

D Cenário 02 com APT

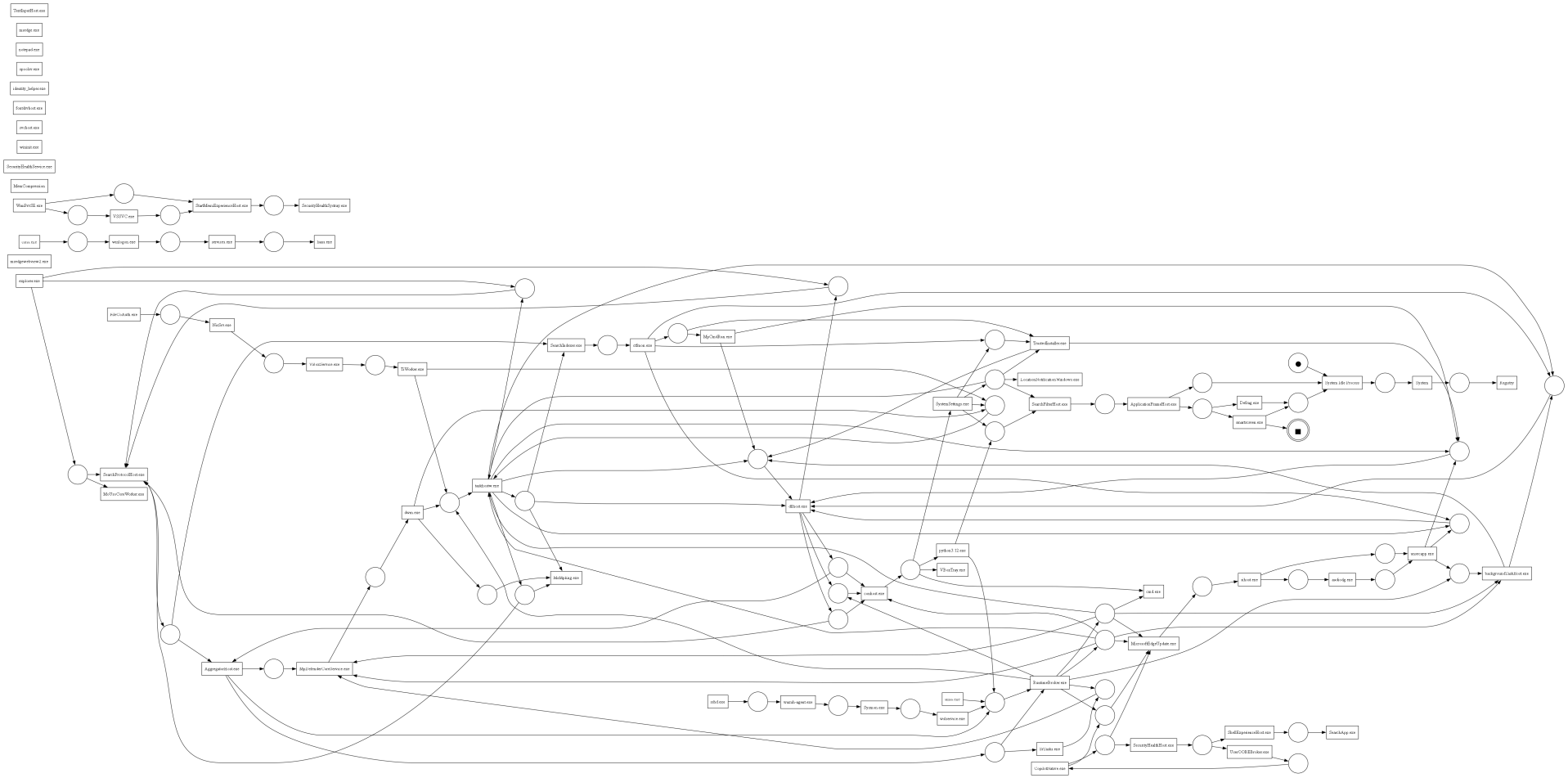

Figura 16 – Fluxo de trabalho da Vítima 02 impactado por APT.
Fonte: O autor.

Referências

AALST, W. V. D. Process mining: Overview and opportunities. *ACM Transactions on Management Information Systems (TMIS)*, ACM New York, NY, USA, v. 3, n. 2, p. 1–17, 2012. Citado 2 vezes nas páginas 21 e 22.

ALSHAMRANI, A. et al. A survey on advanced persistent threats: Techniques, solutions, challenges, and research opportunities. *IEEE Communications Surveys & Tutorials*, IEEE, v. 21, n. 2, p. 1851–1877, 2019. Citado na página 15.

BUCHTA, R. et al. Advanced persistent threat attack detection systems: A review of approaches, challenges, and trends. *Digital Threats: Research and Practice*, ACM New York, NY, v. 5, n. 4, p. 1–37, 2024. Citado 2 vezes nas páginas 13 e 16.

ECK, M. L. V. et al. Pm: a process mining project methodology. In: SPRINGER. *International conference on advanced information systems engineering*. [S.l.], 2015. p. 297–313. Citado na página 25.

GALOYAN, A. Segurança cibernética no âmbito das relações internacionais. *Brasília, DF: Universidade de Brasília*, 2019. Citado na página 13.

GARCIA, C. dos S. et al. Process mining techniques and applications–a systematic mapping study. *Expert Systems with Applications*, Elsevier, v. 133, p. 260–295, 2019. Citado na página 21.

GHAFIR, I.; PRENOSIL, V. et al. Advanced persistent threat attack detection: an overview. *Int J Adv Comput Netw Secur*, v. 4, n. 4, p. 5054, 2014. Citado na página 15.

JUNIOR, H. F. R. A inteligência cibernética e a proteção das infraestruturas críticas de interesse do país. Escola Superior de Guerra, 2024. Citado na página 13.

LEEMANS, S. J. Robust process mining with guarantees. In: SPRINGER. *BPM (Dissertation/Demos/Industry)*. [S.l.], 2018. p. 46–50. Citado na página 23.

LEHTO, M. Apt cyber-attack modelling: Building a general model. In: ACADEMIC CONFERENCES INTERNATIONAL LIMITED. *International Conference on Cyber Warfare and Security*. [S.l.], 2022. v. 17, n. 1, p. 121–129. Citado na página 18.

LEHTO, M. Apt cyber-attack modelling: Building a general model. In: ACADEMIC CONFERENCES INTERNATIONAL LIMITED. *International Conference on Cyber Warfare and Security*. [S.l.], 2022. v. 17, n. 1, p. 121–129. Citado na página 18.

LI, M. et al. The study of apt attack stage model. In: IEEE. *2016 IEEE/ACIS 15th International Conference on Computer and Information Science (ICIS)*. [S.l.], 2016. p. 1–5. Citado na página 17.

MACAK, M. et al. Process mining usage in cybersecurity and software reliability analysis: A systematic literature review. *Array*, Elsevier, v. 13, p. 100120, 2022. Citado 2 vezes nas páginas 14 e 23.

MANS, R. S. et al. Application of process mining in healthcare–a case study in a dutch hospital. In: SPRINGER. *Biomedical Engineering Systems and Technologies: International Joint Conference, BIOSTEC 2008 Funchal, Madeira, Portugal, January 28-31, 2008 Revised Selected Papers 1*. [S.l.], 2009. p. 425–438. Citado na página 23.

PÉREZ-GOMARIZ, M.; CERDÁN-CARTAGENA, F.; GARCÍA, J. Lm-hunter: An nlp-powered graph method for detecting adversary lateral movements in apt cyber-attacks at scale. *Computer Networks*, Elsevier, p. 111181, 2025. Citado na página 20.

QUINTERO-BONILLA, S.; REY, A. Martín del. A new proposal on the advanced persistent threat: A survey. *Applied Sciences*, MDPI, v. 10, n. 11, p. 3874, 2020. Citado na página 16.

REISDOERFER, B.; ALCÂNTARA, B. Alemanha como líder na determinação de ameaças cibernéticas na união europeia. *Revista Carta Internacional*, v. 15, n. 2, p. 163–189, 2020. Citado na página 13.

RODRíGUEZ, M.; BETARTE, G.; CALEGARI, D. A process mining-based method for attacker profiling using the mitre att&ck taxonomy. *Journal of Internet Services and Applications*, v. 15, n. 1, p. 212–232, 2024. Citado na página 24.

SAKTHIVELU, U.; KUMAR, C. V. Advanced persistent threat detection and mitigation using machine learning model. *Intelligent Automation & Soft Computing*, v. 36, n. 3, 2023. Citado na página 13.

SCHMIDT, R.; RATTRAY, G. J.; FOGLE, C. J. *Methods and apparatus for developing cyber defense processes and a cadre of expertise*. [S.l.]: Google Patents, 2008. US Patent App. 11/947,655. Citado na página 15.

SMILIOTOPOULOS, C.; BARMPATSALOU , K.; KAMBOURAKIS, G. Revisiting the detection of lateral movement through sysmon. *Applied Sciences*, v. 12, n. 15, 2022. Citado na página 23.

SMILIOTOPOULOS, C.; KAMBOURAKIS, G.; KOLIAS, C. Detecting lateral movement: A systematic survey. *Heliyon*, Elsevier, v. 10, n. 4, 2024. Citado 2 vezes nas páginas 18 e 23.

THONG, W. J.; AMEEDEEN, M. A survey of petri net tools. In: SPRINGER. *Advanced Computer and Communication Engineering Technology: Proceedings of the 1st International Conference on Communication and Computer Engineering*. [S.l.], 2015. p. 537–551. Citado na página 21.