Universidade de Brasília
Instituto de Ciências Exatas
Departamento de Ciência da Computação

Metodologia para Produção de Inteligência de Ameaça Acionável em Nível Tático

Paulo Roberto da Paz Ferraz Santos

Tese apresentada como requisito parcial para
conclusão do Doutorado em Informática

Orientador
Prof. Dr. André Costa Drummond

Brasília
2025

Ficha Catalográfica de Teses e Dissertações

Está página existe apenas para indicar onde a ficha catalográfica gerada para dissertações de mestrado e teses de doutorado defendidas na UnB. A Biblioteca Central é responsável pela ficha, mais informações nos sítios:

http://www.bce.unb.br
http://www.bce.unb.br/elaboracao-de-fichas-catalograficas-de-teses-e-dissertacoes

Esta página não deve ser inclusa na versão final do texto.

Universidade de Brasília
Instituto de Ciências Exatas
Departamento de Ciência da Computação

Metodologia para Produção de Inteligência de Ameaça Acionável em Nível Tático

Paulo Roberto da Paz Ferraz Santos

Tese apresentada como requisito parcial para
conclusão do Doutorado em Informática

Prof. Dr. André Costa Drummond (Orientador)
CIC/UnB

Prof. Dr. André Costa Drummond
CIC/UnB

Prof. Dr. André Ricardo Abed Grégio
INF/UFPR

Prof. Dr. Robson de Oliveira Albuquerque
UCB

Prof. Dr. Rodrigo Bonifácio de Almeida
CIC/UnB

Prof.a Dr.a Cláudia Nalon
Coordenadora do Programa de Pós-graduação em Informática

Brasília, 18 de dezembro de 2025

Dedicatória

Aos meus pais, à minha esposa e aos meus filhos — alicerces do meu passado, presente e futuro — como testemunho de minha eterna gratidão, amor e inspiração.

Agradecimentos

Agradeço, antes de tudo, a Deus, pela vida, pela saúde e pela força concedida em todos os momentos desta jornada.

Aos meus pais, Carlos Alberto e Lúcia, por todo amor, exemplo e apoio incondicional ao longo da minha vida. À minha amada esposa, Mayumi, pelo companheirismo, paciência e incentivo constantes, que me sustentaram nos períodos mais desafiadores. Aos meus filhos, Daniela e Rafael, pela compreensão diante das longas horas de dedicação e por serem fonte diária de inspiração.

Expresso minha sincera gratidão ao meu orientador, Prof. Dr. André Drummond, pela orientação segura, confiança e constante estímulo à excelência acadêmica.

Ao Prof. Dr. João Gondim, coorientador convidado, manifesto meu mais profundo reconhecimento pela orientação técnica de excelência, pela dedicação exemplar e pela proximidade ao longo de toda a pesquisa. Sua competência, disponibilidade e rigor científico foram fundamentais para o amadurecimento deste trabalho e para o meu próprio crescimento como pesquisador.

Gratidão ao Prof. Dr. Paulo Costa, diretor do C5I Center da George Mason University (Fairfax, VA, EUA), pela oportunidade de realizar, em 2023, um período de pesquisa de seis meses naquele centro de excelência em cibersegurança. Agradeço também ao Prof. Dr. Michael Hieb e ao Prof. Dr. Alexandre de Barreto, da mesma instituição, pelas valiosas orientações e conselhos durante minha estadia no C5I Center.

Agradeço igualmente ao Coronel R1 Salles, tutor acadêmico do IME, e ao Tenente-Coronel Jorge Frederico, supervisor militar, pelo acompanhamento e apoio que me permitiram chegar até aqui, contribuindo de forma significativa para o êxito desta jornada.

Registro, ainda, meu agradecimento ao Coronel Seabra, Chefe do Centro de Pesquisa, Desenvolvimento e Inovação do Comando de Defesa Cibernética, pela compreensão e apoio fundamentais na etapa final do doutorado.

Por fim, agradeço a todas as pessoas que, de forma direta ou indireta, contribuíram para a realização desta tese, oferecendo apoio, incentivo e inspiração ao longo dessa caminhada.

Resumo

A Inteligência de Ameaça Cibernética (CTI) em nível tático descreve o comportamento adversário por meio de Táticas, Técnicas e Procedimentos (TTPs). Seu emprego em atividades de Threat Hunting oferece vantagens significativas em relação à detecção tradicional baseada em Indicadores de Comprometimento (IoCs); contudo, o elevado nível de abstração das TTPs dificulta sua tradução em regras de detecção, limitando a automação e a escalabilidade dessa abordagem. Para enfrentar esse desafio, esta pesquisa propõe uma metodologia estruturada para a produção de inteligência tática acionável, orientada à criação de regras de detecção de comportamentos adversários. Como contribuição preliminar, apresenta-se um estudo abrangente sobre taxonomias de ataques, no qual são identificados requisitos, atributos estruturais e critérios de avaliação que auxiliam a construção e a seleção de taxonomias mais eficazes. A principal contribuição consiste em uma metodologia que define um processo iterativo, apoiado na taxonomia MITRE ATT&CK e na expertise de analistas, para mapear TTPs, desenvolver e validar regras de detecção de forma sistemática, reduzindo a complexidade e favorecendo seu aprimoramento contínuo. Ademais, propõe-se um modelo de dados voltado à organização e padronização das informações geradas em formato de Runbook, o qual serve como fonte de inteligência tática para Threat Hunting e assegura interoperabilidade e reutilização do conhecimento. A proposta foi avaliada por meio de uma Prova de Conceito (PoC) baseada em estudos de caso, que demonstrou sua eficácia na geração de inteligência tática acionável e na criação de regras de detecção precisas, viabilizando a operacionalização da detecção baseada em TTPs. Por fim, são discutidos os benefícios, limitações e desafios observados, bem como apresentadas as perspectivas para pesquisas futuras.

Palavras-chave: inteligência tática de ameaça, CTI acionável, Threat Hunting, TTP, metodologia, Runbook, modelo de dados

Abstract

Cyber Threat Intelligence (CTI) at the tactical level describes adversarial behavior through Tactics, Techniques, and Procedures (TTPs). Its application in Threat Hunting activities offers significant advantages over traditional detection methods based on Indicators of Compromise (IoCs); however, the high level of abstraction of TTPs hinders their translation into detection rules, limiting the automation and scalability of this approach. To address this challenge, this research proposes a structured methodology for producing actionable tactical intelligence, aimed at creating detection rules for adversarial behaviors. As a preliminary contribution, it presents a comprehensive study on attack taxonomies, identifying requirements, structural attributes, and evaluation criteria that support the construction and selection of more effective taxonomies. The main contribution consists of a methodology that defines an iterative process, supported by the MITRE ATT&CK taxonomy and analyst expertise, to systematically map TTPs, develop, and validate detection rules, reducing complexity and fostering continuous improvement. In addition, a data model is proposed to organize and standardize the information generated in Runbook format, which serves as a source of tactical intelligence for Threat Hunting and ensures interoperability and knowledge reuse. The proposal was evaluated through a Proof of Concept (PoC) based on case studies, which demonstrated its effectiveness in generating actionable tactical intelligence and creating accurate detection rules, enabling the operationalization of TTP-based detection. Finally, the observed benefits, limitations, and challenges are discussed, as well as perspectives for future research.

Keywords: tactical threat intelligence, actionable CTI, Threat Hunting, TTP, methodology, Runbook, data model

Sumário

1 Introdução **1**

1.1 Problema e Motivação 2
1.2 Questões de Pesquisa e Desafios 5
1.3 Proposta e Objetivos 6
1.4 Contribuições e Publicações 8
1.5 Organização da Tese 10

2 Taxonomia de Ataques **11**

2.1 Requisitos das Taxonomias 13
2.2 Estrutura das Taxonomias 16
2.2.1 Organização: linear *versus* hierárquica 16
2.2.2 Esquema: plano *versus* multidimensional 17
2.2.3 Rotulagem: primitiva *versus* derivada 19
2.2.4 Abordagem: orientada por característica *versus* orientada por processo 20
2.3 Critérios de Avaliação de Taxonomias 21
2.3.1 Critérios de avaliação dos requisitos da taxonomia 21
2.3.2 Critérios de avaliação estrutural da taxonomia 24
2.4 Revisões de Taxonomias 26
2.4.1 Rede e computadores 26
2.4.2 Ataques DoS/DDoS 29
2.4.3 Ataques de phishing 31
2.4.4 Sistemas em nuvem 33
2.4.5 Aplicações Web 35
2.4.6 Aplicações P2P 36
2.4.7 Infraestrutura da internet 36
2.4.8 Resumo comparativo 37
2.4.9 Questões em aberto, desafios e possíveis direções futuras 38

3 Inteligência de Ameaça Cibernética **42**
3.1 Classificação e escopo da CTI 43
3.2 Limitações da CTI Técnica e relevância da CTI Tática 45
3.3 A Pirâmide da Dor 45

4 Threat Hunting **47**
4.1 Abordagens de detecção de Threat Hunting 48
4.2 Classificação das soluções técnicas 49

5 MITRE ATT&CK **51**
5.1 Fontes e componentes de Dados 52
5.2 Sub-frameworks do MITRE ATT&CK 52
5.3 Relevância na pesquisa 55

6 Trabalhos Relacionados **56**
6.1 Extração de TTPs 56
6.2 Uso de TTPs na detecção de ameaças 58
6.3 Metodologias para detecção baseada em TTPs 59
6.4 Playbooks e Runbooks de Threat Hunting 61

7 Modelo de Dados **65**
7.1 Diagrama de Classes 66

8 Metodologia **71**
8.1 Estágio 1: Mapeamento Ameaça–TTP 73
8.1.1 Obtenção de Conhecimento 74
8.1.2 Mapeamento de TTP 76
8.1.3 Identificação de Relacionamentos 77
8.1.4 Ameaça Mapeada em TTPs 79
8.2 Estágio 2: Mapeamento TTP-Dados 83
8.2.1 Análise do Contexto 83
8.2.2 Mapeamento de Dados 86
8.2.3 Desenvolvimento 90
8.2.4 Regra de Detecção 91
8.3 Estágio 3: Validação 94
8.3.1 Planejamento 95
8.3.2 Execução 97
8.3.3 Avaliação 99
8.3.4 Regra de Detecção Validada 101

8.4 Estágio 4: Consolidação dos Resultados . . . 101
8.4.1 Criação do Runbook . . . 103

9 Formalização Matemática 107
9.1 Estágio 1: Mapeamento Ameaça-TTP . . . 108
9.2 Estágio 2: Mapeamento TTP-Dados . . . 111
9.3 Estágio 3: Validação . . . 114
9.4 Estágio 4: Consolidação dos Resultados . . . 115

10 Prova de Conceito 118
10.1 Estudo de Caso 1: Ataque LSASS Memory Credential Dump . . . 119
10.2 Estudo de Caso 2: Mosquito Malware . . . 130
10.3 Estudo de Caso 3: APT29 . . . 134

11 Avaliação e Discussão 146
11.1 Avaliação da Metodologia . . . 146
11.1.1 Eficácia . . . 146
11.1.2 Corretude . . . 147
11.2 Avaliação do Modelo de Dados . . . 148
11.3 Benefícios e Limitações . . . 149

12 Conclusão e Trabalhos Futuros 152

Referências 156

Apêndice 177

A TTPs mapeadas do Mosquito Malware 178

B Dados da ameaça e referências do APT29 185

C TTPs mapeadas do APT29 188

D Regras de Detecção do APT29 198

Lista de Figuras

1.1 Representação esquemática da proposta 7
1.2 Fases da metodologia baseadas no ciclo PDCA de melhoria contínua 7

2.1 Diagrama de relações das propriedades da taxonomia. 14
2.2 Organização linear (exemplo com 10 classes). 17
2.3 Organização hierárquica (exemplo com 3 níveis, 15 nós e 10 classes). 17
2.4 Esquema multidimensional (exemplo com 4 dimensões). 18
2.5 Exemplo do processo de classificação de categorias primitivas *versus* derivadas [1]. 19

3.1 Processo de produção de inteligência 43
3.2 Tipos de inteligência de ameaça 44
3.3 Pirâmide da Dor (PoP) [2] 46

4.1 Linha do tempo do Threat Hunting [3] 48

7.1 Estrutura simplificada do Runbook 66
7.2 Diagrama de entidades e relacionamentos do modelo de dados 66
7.3 Diagrama de Classes do Runbook 67

8.1 Estrutura hierárquica da metodologia 71
8.2 Diagrama macro da metodologia 72
8.3 Fluxo de ações do 1º estágio: 'Mapeamento Ameaça-TTP' 73
8.4 Exemplo de mapeamento de TTP de um trecho do relatório da ameaça "Operation Double Tap" [4] 79
8.5 Diagrama de classes da Ameaça Mapeada em TTP 81
8.6 Fluxo de ações do 2º estágio: 'Mapeamento TTP-Dados' 84
8.7 Exemplo do processo de desenvolvimento da *query* de uma regra de detecção usando o Estágio 2 da metodologia 92
8.8 Diagrama de classes da Regra de Detecção 94
8.9 Fluxo de ações do 3º estágio: 'Validação' 95

8.10 Exemplo de estrutura de laboratório de Threat Hunting 97
8.11 Diagrama de classes da Regra de Detecção Validada 101
8.12 Fluxo de ações do 4º estágio: 'Consolidação dos Resultados' 103
8.13 Interface gráfica da ferramenta Runbook Creator/Editor 105

10.1 Esquema de validação das Regras de Detecção 127
10.2 Inicialização da sessão Spark e carregamento do dataset 127
10.3 Validação da Regra de Detecção #1 (DTR-0017-2834-5168) 128
10.4 Validação da Regra de Detecção #2 (DTR-0017-2891-6445) 128
10.5 Validação da Regra de Detecção #3 (DTR-0017-2893-5282) 129
10.6 Validação da Regra de Detecção #4 (DTR-0017-2893-8676) 129
10.7 Legenda do diagrama de TTPs 133
10.8 Diagrama de TTPs referentes ao Instalador do malware Mosquito 134
10.9 Diagrama de TTPs referentes ao Lançador e ao Backdoor principal do malware Mosquito 135
10.10Diagrama de TTPs referentes às etapas 1 a 5 do APT29 139
10.11Diagrama de TTPs referentes às etapas 6 a 10 do APT29 141
10.12Inicialização da sessão Spark e carregamento do dataset do APT29 142
10.13Validação da Regra de Detecção DTR-0017-3103-4080 144

Lista de Tabelas

2.1 Propriedades da taxonomia 13
2.2 Requisitos das taxonomias de ataque 16
2.3 Critérios e subcritérios para avaliar os requisitos da taxonomia 22
2.4 Guia para avaliar os requisitos de taxonomia 23
2.5 Orientação sobre como avaliar os subcritérios 23
2.6 Guia para avaliar a estrutura da taxonomia 25
2.7 Comparação de taxonomias de ataque pela avaliação de requisitos 37
2.8 Comparação de taxonomias de ataque pela avaliação estrutural 39

5.1 Exemplos de *Data Sources* e *Data Components* no framework ATT&CK . . 53

6.1 Características abrangidas pelas metodologias analisadas 60
6.2 Comparação da estrutura metodológica entre as abordagens 60
6.3 Informações suportadas nativamente por diferentes modelos de Playbooks/Runbooks 63

8.1 Exemplos de comportamentos associados a *Data Components* 85
8.2 Eventos do Sysmon v15.15 [5] 88
8.3 Exemplo eventos relevantes do *Windows Security Auditing* para identificação de alguns comportamentos 89
8.4 Exemplo de eventos relevantes do *Sysmon* para identificação de alguns comportamentos 89
8.5 Campos do evento 13 (RegistryEvent – Value Set) do *Sysmon* [6] 90

9.1 Definição dos conjuntos de informações produzidos pela metodologia 116
9.2 Definição das funções utilizadas na geração dos conjuntos de informações . 116

10.1 Mapeamento TTP-Dados da Regra de Detecção #1 121
10.2 Mapeamento TTP-Dados da Regra de Detecção #2 123
10.3 Mapeamento TTP–Dados da Regra de Detecção #3 124

10.4 Resumo das principais informações das TTPs mapeadas do Mosquito Malware 132
10.5 Resumo das principais informações das TTPs mapeadas do APT29 138
10.6 Metadados resumidos das Regras de Detecção desenvolvidas para o APT29 140
10.7 Resumo dos testes de validação das regras de detecção do APT29 no dataset malicioso 143

Lista de Algoritmos

1 Pseudocódigo referente ao Estágio 1 da metodologia 82
2 Pseudocódigo referente ao Estágio 2 da metodologia 93
3 Pseudocódigo referente ao Estágio 3 da metodologia 102
4 Pseudocódigo referente ao Estágio 4 da metodologia 104
5 Pseudocódigo referente à execução completa de um ciclo da metodologia . 106

Lista de Abreviaturas e Siglas

API Application Programming Interface.

APT Advanced Persistent Threats.

ATT&CK Adversarial Tactics, Techniques and Common Knowledge.

C2 Command and Control.

CACAO Collaborative Automated Course of Action Operations.

CAR Cyber Analytics Repository.

CATRA Cloud Attack Taxonomy and Risk Assessment.

CERT Computer Emergency Response Team.

CISA Cybersecurity & Infrastructure Security Agency.

COA Course of Action.

CSV Comma-Separated Values.

CTI Cyber Threat Intelligence.

DAG Directed Acyclic Graph.

DDoS Distributed Denial of Service.

DoS Denial of Service.

IA Inteligência Artificial.

IaaS Infrastructure as a Service.

ICS Industrial Control Systems.

IDS Intrusion Detection System.

IoC Indicator of Compromise.

IoT Internet of Things.

JSON JavaScript Object Notation.

KG Knowledge Graph.

LDAP Lightweight Directory Access Protocol.

LLM Large Language Model.

LSASS Local Security Authority Subsystem Service.

ML Machine Learning.

NCSC National Cyber Security Centre.

NLP Natural Language Processing.

OSI Open System Interconnection.

OSSEM Open Source Security Events Metadata.

P2P Peer-to-peer.

PaaS Platform as a Service.

PDCA Plan, Do, Check, and Act.

PoC Proof of Concept.

PoP Pyramid of Pain.

QP Questão de Pesquisa.

ROP Return-Oriented Programming.

SaaS Software as a Service.

SAM Security Account Manager.

SO Sistema Operacional.

SQL Structured Query Language.

TI Tecnologia da Informação.

TIC Tecnologia da Informação e Comunicação.

TRAM Threat Report ATT&CK Mapper.

TTP Tácticas, Técnicas e Procedimentos.

UAC User Account Control.

UPX Ultimate Packer for eXecutables.

UTC Coordinated Universal Time.

WMI Windows Management Instrumentation.

XML eXtensible Markup Language.

XSS Cross Site Scripting.

YAML YAML Ain't Markup Language.

Capítulo 1

Introdução

Nas últimas décadas, vivenciamos uma profunda revolução nas Tecnologia da Informação e Comunicação (TIC), que tornou a sociedade mais interconectada e dependente de informações e sistemas computacionais. Paralelamente, os criminosos cibernéticos evoluíram técnica e organizacionalmente, tornando-se mais engenhosos, persistentes, bem financiados, menos previsíveis e altamente motivados por ganhos financeiros [7]. O surgimento de novas gerações de ataques — mais dinâmicos, coordenados e sofisticados — tem causado prejuízos crescentes a empresas, governos e instituições.

De acordo com a IBM Security e o Ponemon Institute, o custo médio global de uma violação de dados em 2024 foi de US$ 4,88 milhões, representando um aumento de 9,6% em relação a 2023 e um crescimento acumulado de 26,4% desde 2020 [8]. No mesmo ano, o custo global total de crimes cibernéticos foi estimado em US$ 9,22 trilhões, com previsão de atingir mais de US$ 15 trilhões até 2029 — um crescimento de aproximadamente 70% [9]. Esses números evidenciam que as medidas tradicionais de segurança têm se mostrado insuficientes diante da crescente sofisticação dos ataques, que exploram vulnerabilidades em pessoas, processos e tecnologias [7].

Entre os exemplos mais emblemáticos de ameaças sofisticadas estão as Ameaças Persistentes Avançadas (Advanced Persistent Threats – APT), caracterizadas por alto nível de furtividade e capacidade de evasão. Uma vez que obtêm acesso inicial, podem permanecer indetectáveis por longos períodos — meses ou até anos — operando de forma discreta e persistente. Sua natureza furtiva dificulta, ou mesmo inviabiliza, a detecção por mecanismos tradicionais, como os sistemas de detecção baseados em anomalias (*anomaly-based Intrusion Detection System*).

Nesse contexto, a Inteligência de Ameaças Cibernéticas (Cyber Threat Intelligence – CTI) consolidou-se como uma disciplina essencial da defesa cibernética moderna. A CTI fornece indicadores, análises e melhores práticas, além de oferecer uma compreensão aprofundada das táticas, técnicas e procedimentos (TTPs) empregados por adversários

[10]. Assim, tornou-se uma aliada fundamental na resposta ao crescente volume e à complexidade dos incidentes de segurança. Seu propósito central é produzir conhecimento baseado em evidências sobre ameaças, permitindo às organizações antecipar ataques, mitigar invasões e detectar atividades maliciosas de forma mais precoce e eficaz [7].

Embora a CTI produza informações valiosas para identificação de ameaças e atores maliciosos, ela nem sempre é suficiente para proteger organizações de ataques sofisticados, sobretudo aqueles que utilizam técnicas avançadas de evasão ou exploram vulnerabilidades desconhecidas (*zero-day exploits*). Nesse cenário, o *Threat Hunting* surge como uma abordagem proativa, voltada à identificação de ameaças que conseguiram contornar as defesas tradicionais antes que causem danos significativos.

O Threat Hunting e a CTI são atividades complementares. Enquanto a CTI fornece insumos estratégicos e táticos que orientam a caça a ameaças, o Threat Hunting contribui para validar e enriquecer as informações de inteligência, ao identificar comportamentos suspeitos que passam despercebidos pelos mecanismos de defesa tradicionais.

A efetividade do Threat Hunting, entretanto, depende de informações úteis, contextualizadas e suficientemente detalhadas que permitam uma ação de caça rápida e precisa. Tais informações são denominadas "*acionáveis*" [11, 12]. Tornar a CTI acionável, portanto, é um desafio estratégico fundamental para viabilizar operações efetivas de Threat Hunting.

1.1 Problema e Motivação

As defesas cibernéticas tradicionais ainda se apoiam majoritariamente em indicadores técnicos — como *hashes* de arquivos, endereços IP e domínios maliciosos — usados para criar regras automáticas de detecção [13]. Esses elementos, conhecidos como Indicadores de Comprometimento (Indicator of Compromise – IoC), constituem a base da CTI Técnica [14].

A CTI Técnica é o tipo mais difundido entre os fornecedores de inteligência [15], tanto pela facilidade de geração e distribuição de IoCs quanto por sua natureza diretamente aplicável a mecanismos de segurança [16]. Seus dados são normalmente disponibilizados em formatos padronizados e legíveis por máquina (*machine-readable feeds*), permitindo rápida integração com sistemas de detecção. Embora seu uso no Threat Hunting seja bastante comum, os IoCs apresentam limitações significativas que devem ser compreendidas.

Um dos principais problemas é a falta de contexto operacional: a maioria dos *feeds* públicos não informa o papel do IoC no ataque (*e.g.* se o IP corresponde a um servidor de comando e controle) nem a tática ou técnica associada [17, 13].

Problema 1: Alertas baseados em IoC não fornecem contexto sobre a tática ou técnica de ataque, dificultando a resposta efetiva do defensor.

Outro problema é que os IoC se baseiam em dados fáceis de serem alterados. Adversários podem evadir sistemas de detecção baseados em IoC alterando domínios, endereços IP, *hashes* de arquivos maliciosos ou outros artefatos [2]. Essa volatilidade reduz drasticamente sua validade e confiabilidade, impactando diretamente a efetividade do uso de CTIs Técnicas para a detecção de ataques mais sofisticados ou de longo prazo.

Problema 2: Regras baseadas em CTI Técnicas possuem validade reduzida e, consequentemente, menor confiabilidade.

Por outro lado, a CTI Tática — que descreve como os adversários conduzem seus ataques, incluindo ferramentas, metodologias e comportamentos observáveis — oferece informações mais duradouras e estáveis [14]. Essa inteligência compreende as Táticas, Técnicas e Procedimentos (TTP), mais difíceis de modificar do que simples indicadores técnicos [2]. Assim, operar no nível tático é mais eficaz para detectar ataques sofisticados e prolongados, como os ataques APT.

Entretanto, a aplicação prática das CTI Táticas enfrenta obstáculos. Apesar das iniciativas de padronização, como o framework ATT&CK [18], ainda não há mecanismos amplamente consolidados que permitam seu uso direto em sistemas de detecção [19].

Problema 3: As CTI Táticas, embora valiosas, ainda não são diretamente acionáveis pelos sistemas de detecção.

O MITRE ATT&CK fornece uma base de conhecimento estruturada de táticas e técnicas adversárias observadas em cenários reais, acompanhada de uma taxonomia organizada em formato de matriz. Entretanto, suas técnicas são descritas em linguagem natural e em um nível elevado de abstração, o que exige a atuação de analistas para sua interpretação e tradução em regras de detecção automatizáveis.

Problema 4: O alto grau de abstração das TTPs dificulta sua aplicação prática e reduz a eficiência da detecção.

No contexto de Threat Hunting, o grande volume de dados coletados de diversas fontes torna a eficiência de detecção altamente dependente da forma como as buscas são conduzidas. A busca por informações atômicas — como endereços IP, *hashes*, URLs ou nomes de arquivos — é relativamente simples e pode ser amplamente automatizada. Em contraste, as informações de natureza comportamental não possuem caráter atômico, o que

torna sua identificação significativamente mais complexa. Além disso, as Táticas, Técnicas e Procedimentos (TTPs) são, em geral, descritos em linguagem textual explicativa, exigindo interpretação minuciosa por parte dos analistas para sua tradução em consultas manuais. Diante desse cenário, a estratégia mais eficiente consiste em transformar as TTPs em regras ou padrões de busca específicos; contudo, essa tarefa é notoriamente desafiadora e requer elevado nível de expertise técnica e analítica.

Problema 5: Converter TTPs em regras e padrões de detecção é uma tarefa complexa e altamente dependente da expertise do analista.

Embora iniciativas como o Atomic Red Team [20], o MITRE CAR [21] e empresas como a Splunk[1] busquem mapear ameaças por meio de Táticas, Técnicas e Procedimentos (TTPs) e desenvolver regras de detecção correspondentes, essas iniciativas carecem de uma metodologia sistemática, detalhada e reprodutível.

Problema 6: Ausência de uma metodologia estruturada e reprodutível para a produção de regras de detecção baseadas em TTP.

A principal motivação para tornar as CTI Táticas acionáveis reside na necessidade de aumentar a eficácia da detecção por meio da observação de comportamentos adversários. Em operações de Threat Hunting, a detecção de uma ameaça indica que ela já ultrapassou as barreiras de segurança convencionais. Nesse contexto, o uso de regras de detecção baseadas em TTPs, em vez de IoCs, proporciona uma visão mais detalhada da ameaça. Essa abordagem permite identificar as ações específicas do atacante — como reconhecimento, movimento lateral ou exfiltração de dados — e oferece um embasamento mais sólido para uma resposta a incidentes mais precisa e eficiente.

Outra motivação importante é que as CTI Táticas tendem a ser mais duráveis que as CTI Técnicas, pois descrevem padrões de comportamento relativamente estáveis ao longo do tempo [13]. Assim, regras de detecção baseadas em TTPs mantêm-se válidas por períodos mais longos, exigindo menos atualizações e oferecendo maior confiabilidade operacional.

Além disso, as CTI Técnicas, como os IoCs, são informações facilmente contornáveis pelos adversários, o que reduz sua vida útil e aumenta o risco de falsos positivos. A enorme quantidade de IoCs gerada também amplia consideravelmente o volume de regras de detecção, elevando os custos de manutenção e o consumo de recursos computacionais.

Por fim, a criação de uma base de CTI Táticas acionáveis — composta por regras e padrões de detecção estruturados — permite aprimorar substancialmente a capacidade de detecção de comportamentos maliciosos. Tal base fornece aos analistas de Threat

[1] `https://www.splunk.com`

Hunting um instrumento poderoso para identificar, compreender e responder a ameaças de forma mais precisa, eficiente e antecipada, reduzindo o impacto potencial de incidentes de segurança.

1.2 Questões de Pesquisa e Desafios

Com base nos problemas apresentados na seção anterior, e visando aprimorar a detecção de ameaças com base em comportamentos adversários, foram formuladas as seguintes Questões de Pesquisa (QPs), que nortearam o desenvolvimento desta tese:

QP1: Como tornar as CTI Táticas acionáveis para aplicação em operações de Threat Hunting?

QP2: Como identificar comportamentos adversários (informações de alto nível) a partir de dados coletados em sistemas e fontes diversas (baixo nível)?

QP3: Como associar os comportamentos descritos nas TTPs do Framework ATT&CK a informações observáveis em fontes de dados coletáveis?

QP4: Como produzir regras ou padrões de busca aplicáveis ao Threat Hunting, a partir de CTI Táticas e outros conhecimentos, para detecção de TTPs?

Para responder a essas questões, esta pesquisa enfrentou um conjunto de desafios conceituais e práticos, descritos a seguir.

As CTIs Táticas, diferentemente dos indicadores técnicos, não são diretamente acionáveis, pois descrevem padrões de comportamento adversário em vez de eventos discretos facilmente observáveis em registros de log. Por representarem a forma como os atacantes operam, as TTPs são predominantemente documentadas em formato textual e descritivo, em fontes como o Framework ATT&CK [18] e relatórios de CTI. No entanto, essa natureza abstrata dificulta sua aplicação direta na detecção de ameaças, exigindo a atuação de especialistas para interpretar, contextualizar e traduzir essas informações em ações concretas de busca e correlação de eventos.

Outro desafio central reside na criação de regras acionáveis a partir da inteligência tática, capazes de detectar comportamentos adversários com base nas TTPs. Essa tarefa é complexa e altamente dependente da expertise dos analistas, pois envolve interpretação contextual, correlação de múltiplas fontes de dados e validação empírica. Atualmente, esse processo ocorre de maneira não sistematizada, variando conforme o conhecimento e a experiência individual de cada profissional. A ausência de um método estruturado para transformar TTPs em regras de detecção prejudica a operacionalização do Threat

Hunting baseado em TTPs, além de limitar seu potencial de automação e aplicação em larga escala.

Adicionalmente, há o desafio de estruturar e organizar as informações derivadas da inteligência tática de forma reutilizável e interoperável. A falta de um modelo de dados padronizado compromete a correlação eficiente entre TTPs, e restringe a integração da CTI Tática em diferentes plataformas e ferramentas de Threat Hunting, reduzindo sua aplicabilidade operacional e dificultando a consolidação de conhecimento compartilhável.

1.3 Proposta e Objetivos

Diante da motivação apresentada na Seção 1.1 e com o propósito de superar os desafios identificados na Seção 1.2, esta tese tem como objetivo geral propor uma metodologia estruturada para a produção de CTI Tática acionável voltada ao Threat Hunting, com foco na operacionalização da abordagem baseada em TTP.

A inteligência tática produzida pela metodologia visa fornecer, simultaneamente, o mapeamento das TTPs que compõem a ameaça e as regras capazes de detectar, nos dados coletados, os comportamentos maliciosos decorrentes das técnicas mapeadas. O objetivo principal da metodologia é, portanto, estabelecer um processo sistemático que oriente os analistas na criação de regras de detecção fundamentadas nas TTPs da ameaça.

Para isso, a metodologia utiliza conhecimentos derivados de CTIs táticas, a expertise de analistas e a taxonomia do MITRE ATT&CK [18], a fim de produzir um documento denominado Runbook de Threat Hunting, conforme ilustrado na Figura 1.1. O Runbook registra e organiza, de forma padronizada, as regras de detecção desenvolvidas, juntamente com as TTPs mapeadas, suas referências e demais informações relevantes. Assim, constitui uma fonte estruturada de inteligência tática acionável destinada a subsidiar analistas de Threat Hunting na configuração de estratégias baseadas em regras para a detecção de ameaças em ambientes operacionais.

Desenvolver regras acionáveis a partir de inteligência de ameaça em nível tático, capazes de detectar comportamentos adversários com base em TTPs, é uma tarefa complexa. O objetivo da metodologia é reduzir essa complexidade por meio da especificação de etapas claras e objetivas que orientem os analistas na criação dos Runbooks de Threat Hunting.

As regras de detecção desenvolvidas têm como propósito servir como instruções diretamente aplicáveis às ferramentas de Threat Hunting, permitindo aos analistas identificar de forma rápida — e sem necessidade de interpretação adicional — as táticas e técnicas adversárias em grandes volumes de dados provenientes de múltiplas fontes.

Outro aspecto fundamental relacionado às regras de detecção, e que constitui um requisito essencial da segurança cibernética, é a necessidade de manutenção e atualiza-

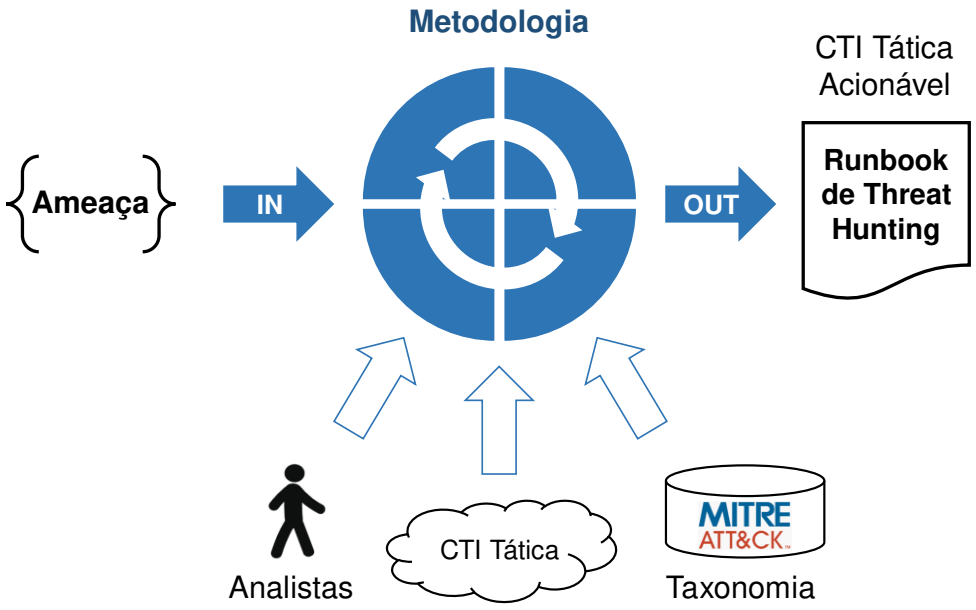


Figura 1.1: Representação esquemática da proposta

ção contínuas, de modo a preservar a eficácia das operações de Threat Hunting frente à constante evolução dos agentes de ameaça. Para isso, a metodologia proposta adota um modelo de quatro fases inspirado no ciclo PDCA (*Plan, Do, Check, Act*), ilustrado na Figura 1.2, reconhecido por sua eficácia em promover a melhoria contínua.

Figura 1.2: Fases da metodologia baseadas no ciclo PDCA de melhoria contínua

O ciclo PDCA, amplamente empregado na gestão de processos e projetos, possui uma estrutura iterativa que favorece a avaliação e o aprimoramento contínuo [22]. Essas

características são essenciais para garantir a produção de Runbooks de alta qualidade para uso em Threat Hunting. As quatro fases da metodologia proposta são descritas a seguir:

1. **Obter informações comportamentais da ameaça:** Esta fase inicial corresponde ao *Plan* do ciclo PDCA. Envolve a coleta e a análise de informações de inteligência de ameaça (CTI) com o objetivo de compreender o comportamento adversário e identificar as táticas, técnicas e procedimentos (TTPs) utilizados pelo atacante. Essa etapa realiza o mapeamento Ameaça–TTP.

2. **Definir regras de detecção:** Correspondente ao *Do* do ciclo PDCA, esta fase consiste na definição de regras capazes de identificar os comportamentos adversários nos dados coletados de fontes específicas. As regras resultam do mapeamento TTP–dados.

3. **Validar as regras de detecção:** Relativa ao *Check* do ciclo PDCA, esta fase compreende a validação das regras definidas anteriormente, garantindo que sejam eficazes na detecção dos comportamentos adversários correspondentes às TTPs analisadas.

4. **Compor o Runbook de Threat Hunting:** A fase final corresponde ao *Act* do PDCA. Ela consolida os resultados das fases anteriores, utilizando um modelo de dados para estruturar as informações no Runbook de Threat Hunting.

O modelo de dados utilizado é apresentado no Capítulo 7, enquanto a metodologia proposta é detalhada no Capítulo 8.

1.4 Contribuições e Publicações

A presente pesquisa apresenta diversas contribuições que podem ser agrupadas em duas áreas principais:

1. Taxonomias de ataques;

2. Inteligência tática de ameaça.

Cronologicamente, a pesquisa iniciou-se com um estudo sobre taxonomias de ataques, considerando o papel crucial que essas taxonomias desempenham na compreensão e prevenção de incidentes cibernéticos. Nesse estudo, verificou-se que a ausência de métodos de avaliação abrangentes representa uma lacuna significativa na literatura, dificultando o desenvolvimento e a aplicabilidade dessas taxonomias.

Para abordar essa lacuna, foi realizado um *survey* envolvendo 20 taxonomias de ataques publicadas entre 2011 e 2022, avaliadas a partir de um novo conjunto de critérios qualitativos e quantitativos, propostos com base em requisitos fundamentais de taxonomia e em atributos estruturais essenciais.

No processo de formulação da metodologia e de seus critérios de avaliação, foram investigadas as principais propriedades de taxonomias mencionadas na literatura, bem como suas dependências e inter-relações. Essa investigação permitiu extrair os requisitos fundamentais para uma taxonomia de ataques relevante e amplamente aceita pela comunidade de segurança cibernética. Aspectos estruturais importantes — como organização, esquema, rotulagem e abordagem — também foram analisados, considerando seu impacto na eficácia e na aplicabilidade das taxonomias.

A pesquisa supramencionada, descrita com mais detalhes no Capítulo 2, apresenta diversas contribuições significativas, entre as quais destacam-se:

- Identificação de requisitos relevantes para a construção de taxonomias úteis e com maior aceitação pela comunidade científica e profissional.
- Análise de questões estruturais com potencial para impactar a eficácia e a aplicabilidade das taxonomias.
- Proposição de critérios de avaliação claros e objetivos para auxiliar profissionais de segurança na escolha da taxonomia de ataques mais adequada às suas aplicações, bem como apoiar desenvolvedores na avaliação e aprimoramento de suas propostas.
- Revisão de 20 taxonomias de ataques publicadas entre 2011 e 2022, desenvolvidas para diferentes contextos de aplicação, incluindo redes e computadores, ataques de DoS e DDoS, *phishing*, computação em nuvem, aplicações Web, redes P2P e infraestrutura da Internet.
- Avaliação das taxonomias com base em critérios qualitativos e quantitativos fundamentados em requisitos e atributos estruturais, apresentando os resultados em tabelas comparativas.

O restante desta pesquisa concentra-se na aplicação da Inteligência de Ameaça Cibernética (CTI) ao Threat Hunting, com foco no uso da inteligência tática de ameaça para a detecção de comportamentos adversários com base nas TTPs. As principais contribuições desta etapa da pesquisa são:

- Definição de uma metodologia estruturada para a produção de inteligência tática acionável voltada ao Threat Hunting baseado em TTPs.
- Proposição de um modelo de dados que formaliza essa inteligência em um formato padronizado, reprodutível e interoperável.

- Formalização matemática da metodologia proposta.
- Avaliação da metodologia por meio de uma prova de conceito composta por estudos de caso que demonstram a eficácia da inteligência tática produzida, evidenciando sua capacidade de apoiar a detecção de ameaças reais.

Como resultado das pesquisas realizadas sobre taxonomias de ataques e sobre a produção de inteligência tática acionável para Threat Hunting, foram produzidos os seguintes artigos:

- P. R. da P. F. Santos, P. A. A. Resende, J. J. C. Gondim, and A. C. Drummond, "Towards Robust Cyber Attack Taxonomies: A Survey with Requirements, Structures, and Assessment", *ACM Computing Surveys*, vol. 57, no. 8, p. 199:1-199:36, Mar. 2025, doi: 10.1145/3717606. [23].
- P. R. da P. F. Santos, A. C. Drummond, and J. J. C. Gondim, "Producing actionable tactical intelligence for rule-based Threat Hunting: methodology and data model", *Security and Privacy*. (Aceito para publicação).

1.5 Organização da Tese

O restante deste trabalho está organizado da seguinte forma:

O Capítulo 2 apresenta a pesquisa sobre taxonomias de ataques e os critérios desenvolvidos para sua avaliação. O Capítulo 3 aborda de forma sucinta a Inteligência de Ameaça Cibernética (*Cyber Threat Intelligence*) e seus subdomínios. O Capítulo 4 descreve a atividade de Threat Hunting e as principais abordagens de detecção utilizadas. O Capítulo 5 apresenta o *framework* ATT&CK da MITRE e sua utilidade para o mapeamento de comportamentos adversários. O Capítulo 6 discute os trabalhos relacionados a esta pesquisa.

O Capítulo 7 descreve o modelo de dados proposto para representar e formalizar a inteligência tática produzida. O Capítulo 8 detalha a metodologia proposta nesta pesquisa. O Capítulo 9 apresenta a formalização matemática dessa metodologia. O Capítulo 10 descreve a prova de conceito (PoC), composta por estudos de caso. O Capítulo 11 discute e avalia os resultados obtidos. Por fim, a tese é concluída no Capítulo 12.

Capítulo 2

Taxonomia de Ataques

O campo da segurança cibernética é vasto e abrange diversas subáreas em constante evolução. A organização e descrição sistemáticas desses temas são essenciais para o avanço e amadurecimento do conhecimento nesse domínio [24]. Um dos meios mais eficazes para alcançar essa organização é o uso de taxonomias.

O conceito de taxonomia foi originalmente concebido por Carolus Linnaeus [25] para agrupar e classificar organismos em uma hierarquia de níveis fixos [26]. Em termos gerais, taxonomia é a ciência da classificação [27]. O termo deriva das palavras gregas *taxis* ($\tau\acute{\alpha}\xi\iota\varsigma$), que significa 'ordem', 'arranjo' ou 'organização', e *nomos* ($\nu\acute{o}\mu\varsigma$), que significa 'lei', 'norma' ou 'regra' [28]. Inicialmente aplicada à botânica e à zoologia, a taxonomia surgiu para classificar sistemática e metodicamente animais e plantas de acordo com suas semelhanças e diferenças [29, 30, 31].

Enquanto a classificação consiste no processo de separar ou ordenar algo em classes com base em semelhanças [32], a taxonomia oferece uma sistemática organizacional [33], conferindo à classificação significado teórico e consistência [34]. Ela integra termos, princípios, metodologias e estruturas de organização, permitindo uma compreensão mais eficaz das inter-relações entre conceitos em campos específicos [35, 36]. Além disso, a taxonomia possibilita a identificação inequívoca de fenômenos ou objetos e facilita a formulação de inferências úteis, auxiliando na detecção de lacunas de conhecimento [26, 37]. Ao segmentar o contínuo em categorias discretas adequadas à análise detalhada, a taxonomia também apoia a tomada de decisão [26]. Ao fornecer terminologia comum, ela facilita o compartilhamento de conhecimento e a comparação de informações [38, 26].

As primeiras taxonomias em segurança cibernética surgiram na década de 1970, com o objetivo de caracterizar falhas em sistemas operacionais [39, 40, 41]. Posteriormente, foram desenvolvidas taxonomias para classificar falhas [42, 43], erros [44, 45] e vulnerabilidades [46, 47] em software, bem como para redes [48, 49], protocolos [50, 51], firewalls [52] e Sistemas de Detecção de Intrusão (Intrusion Detection System – IDS) [53, 54].

As taxonomias de segurança não se limitam a sistemas ou software, abrangendo também hardware [55, 56, 57], Internet das Coisas (Internet of Things – IoT) [58, 59, 60], engenharia social [61, 62], autenticação biométrica [63], blockchain [64], dispositivos móveis [65, 66], veículos automotores [67, 68], Smart Grid [69], entre outros contextos.

Em cada contexto, as taxonomias podem abordar distintas facetas, como vulnerabilidades [70], ameaças [71], ataques [72, 57], contramedidas [73, 74], mecanismos de defesa [75, 76], estratégias [69], ferramentas [77, 78], frameworks [79, 80], resposta a incidentes [81, 82] e conjuntos de dados de segurança [60], entre outros. Assim, diferentes combinações entre contexto e faceta podem ser exploradas. Por exemplo, no contexto da engenharia social, há taxonomias que se concentram tanto na caracterização de ataques quanto na definição de mecanismos de defesa [83, 84].

Essa multiplicidade de taxonomias evidencia a amplitude e a complexidade inerentes à área de segurança cibernética, que decorrem da dicotomia fundamental entre defesa e ataque. Compreender essa dualidade é essencial para o avanço da pesquisa na área.

Do ponto de vista dos defensores, esforços contínuos são feitos para preservar propriedades essenciais, como disponibilidade, integridade, confidencialidade e o Triplo-A (autenticação, autorização e *accountability*). Por outro lado, do ponto de vista dos invasores, o cenário é distinto. Segundo a ISO/IEC 27000/2018, um ataque é "uma tentativa de destruir, expor, alterar, desativar, roubar ou obter acesso não autorizado ou fazer uso não autorizado de um ativo" [85]. Simplificando, trata-se de um método específico para explorar intencionalmente e de forma maliciosa uma vulnerabilidade [86]. Enquanto os invasores procuram continuamente pontos fracos exploráveis, os defensores trabalham para fortalecer as barreiras de proteção. Ao compreenderem os mecanismos defensivos, os invasores aprimoram suas táticas, técnicas e procedimentos, desenvolvendo métodos mais eficazes de evasão e furtividade com base nas estratégias de proteção existentes.

De maneira complementar, a compreensão aprofundada de ataques cibernéticos contribui para aprimorar as defesas. Por essa razão, a taxonomia de ataques desempenha papel central na segurança cibernética, oferecendo uma estrutura para a classificação sistemática de ataques com base em suas características [87]. Isso proporciona uma visão clara, estruturada e compreensível do panorama de ameaças [88], permitindo estudos sistemáticos que geram insights relevantes para pesquisa e desenvolvimento. Esses esforços visam fortalecer operações essenciais de segurança, incluindo prevenção, detecção e mitigação de ataques.

2.1 Requisitos das Taxonomias

Os desenvolvedores de taxonomias enfrentam, há muito tempo, o desafio de atingir um objetivo comum: *como classificar e organizar fenômenos ou objetos com base em suas características compartilhadas*. Essa busca deu origem a uma ampla variedade de taxonomias, cada uma estruturada segundo diferentes critérios de concepção. No entanto, a variabilidade desses critérios resultou em taxonomias com propriedades divergentes, o que, por sua vez, afeta significativamente sua usabilidade e, consequentemente, sua aceitação na comunidade científica.

A heterogeneidade resultante leva a uma questão fundamental: *quais propriedades uma taxonomia deve possuir para ser bem-sucedida?* Em resposta a essa questão, diversos pesquisadores se empenharam em identificar e articular um conjunto de propriedades que possam orientar o desenvolvimento de taxonomias e servir como critérios de avaliação. Um resumo das propriedades desejáveis de uma taxonomia satisfatória, propostas por diferentes autores ao longo das últimas três décadas, é apresentado na Tabela 2.1.

Tabela 2.1: Propriedades da taxonomia

Propriedade	**Descrição**	**Proposta por**
Aceita [89, 38, 88, 90, 72, 1, 91, 92]	A taxonomia deve ser aceita pela comunidade em geral.	[89]
Apropriada [93, 88, 90]	A taxonomia deve classificar adequadamente os ataques com base em suas características, considerando quaisquer restrições pertinentes.	[93]
Completa [93, 88, 90, 72, 1, 91, 92]	A taxonomia deve abranger todos os ataques possíveis dentro de seu escopo.	[93]
Compreensível [88, 90, 72, 1, 91, 92]	A taxonomia deve ser compreendida tanto por especialistas quanto por profissionais menos familiarizados com a área de segurança.	[88]
Determinismo [42, 90, 72, 91, 92]	O procedimento de classificação deve ser claramente definido.	[42]
Exaustiva [89, 88, 38, 90, 72, 1, 91]	As categorias devem incluir todas as possibilidades, garantindo que cada ataque se enquadre em pelo menos uma delas.	[89, 88]
Representatividade humana [94]	A taxonomia deve abranger ataques que exploram o elemento humano da cibersegurança.	[94]
Ameaças internas *versus* externas [93, 88, 90]	A taxonomia deve diferenciar entre ameaças internas e externas.	[93]
Mutuamente exclusivas [89, 88, 38, 90, 72, 1, 91, 92]	A taxonomia deve classificar cada ataque em apenas uma categoria em cada dimensão.	[89, 88]
Objetividade [42, 90]	A escolha da categoria deve ser orientada por observações claras, sem inferências subjetivas.	[42]
Primitiva [95, 90]	O processo de classificação deve ser simples, baseado em respostas binárias (sim/não).	[95]
Objetiva [96]	A taxonomia deve ter um objetivo claramente definido.	[96]
Repetível [89, 88, 38, 90, 72, 1, 91, 92]	Classificações repetidas devem gerar o mesmo resultado, independentemente de quem classifica.	[89, 88]
Similaridade [95, 90]	Profissionais distintos devem classificar ataques semelhantes na mesma categoria.	[95]
Específica [42, 90]	As categorias devem ser únicas e inequívocas.	[42]
Terminologia em conformidade com a terminologia de segurança estabelecida [88, 90, 72, 91, 92]	A taxonomia deve empregar terminologia compatível com o conhecimento e padrões de segurança existentes.	[88]
Termos bem definidos [95, 90, 72, 1, 91]	Os termos devem possuir significados precisos e consistentes.	[95]
Inequívoca [89, 38, 42, 90, 72, 1, 91]	As categorias devem ser definidas de forma clara e precisa, permitindo classificação consistente independente de quem a realiza.	[89]
Útil [89, 88, 38, 90, 72, 1, 91]	A taxonomia deve ser aplicável e atender às necessidades da comunidade de segurança.	[89, 88]
Versátil [94]	A taxonomia deve possuir mecanismos para incorporar novas categorias, mantendo-se atualizada frente a novos tipos de ataque.	[94]

A falta de consenso entre os autores, aliada à existência de semelhanças e dependências entre determinadas propriedades, sugere a necessidade de análises adicionais para determinar os requisitos mínimos — livres de redundâncias e sobreposições — que caracterizem uma taxonomia de ataques satisfatória e amplamente aceita pela comunidade de segurança cibernética. Assim, nesta pesquisa, a propriedade "*aceita*" deixa de ser considerada uma característica intrínseca da taxonomia, passando a representar um estado de sucesso, *i.e.*, o objetivo a ser alcançado. A Figura 2.1 ilustra a inter-relação entre as propriedades da taxonomia para atingir esse objetivo.

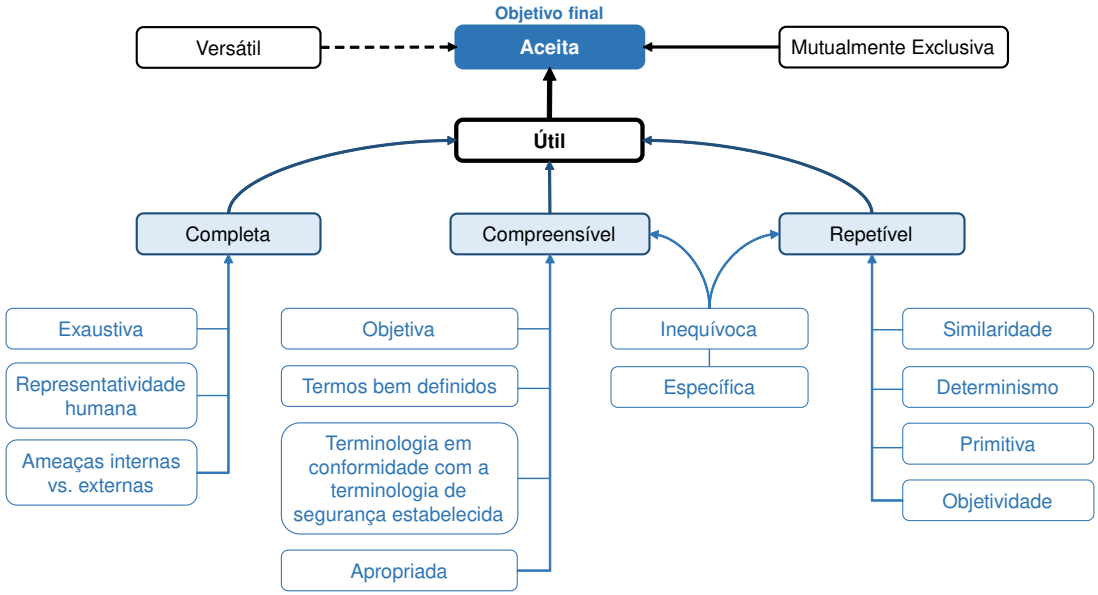


Figura 2.1: Diagrama de relações das propriedades da taxonomia.

O requisito mais relevante para a aceitação de uma taxonomia é a utilidade. Uma taxonomia *útil* é aquela que cumpre satisfatoriamente seu propósito para uma aplicação específica. Para isso, deve ser *completa*, *compreensível* e *repetível*.

Uma taxonomia *completa* é capaz de classificar todos os ataques dentro de seu escopo definido. Para tanto, suas categorias devem ser *exaustivas*, abrangendo todas as formas potenciais de ataque — incluindo aqueles que exploram o elemento humano e aqueles provenientes de ameaças internas e externas. Em essência, uma taxonomia é considerada *completa* quando é *exaustiva*, possui *representatividade humana* e diferencia *ameaças internas e externas*.

Uma taxonomia *compreensível* deve ser clara para especialistas e não especialistas. Um objetivo bem definido e o uso consistente de terminologia reconhecida pela comunidade de segurança contribuem para essa clareza. Para favorecer a compreensão, as categorias devem ser organizadas de forma coerente com as características dos ataques observados, além de possuírem rótulos únicos e inequívocos. Assim, uma taxonomia é considerada

altamente compreensível quando satisfaz as seguintes propriedades: *objetiva*, *termos bem definidos*, *terminologia em conformidade*, *inequívoca*, *específica* e *apropriada*.

Uma taxonomia *repetível* deve permitir a categorização consistente de ataques semelhantes, produzindo resultados idênticos independentemente do classificador, conforme o princípio da *similaridade* [95]. Para isso, o processo de classificação deve ser simples, objetivo e claramente definido, conforme o princípio do *determinismo* [42]. A *objetividade* exige que as categorias sejam atribuídas com base em observações diretas, sem inferências. A simplicidade é assegurada quando as categorias são *primitivas*, permitindo classificações por meio de respostas binárias (sim/não) [95]. Além disso, as categorias devem ser *específicas* e *inequívocas*, evitando classificações incorretas. Essas propriedades contribuem para taxonomias *compreensíveis* e *repetíveis*, como mostrado na Figura 2.1.

Uma taxonomia *mutuamente exclusiva* assegura que cada ataque pertença a uma única categoria em cada dimensão, evitando sobreposições. Caso um ataque se enquadre simultaneamente nas categorias A e B, a exclusividade é violada; nesse caso, uma solução possível é introduzir uma nova categoria C, tal que $C = A \cap B$, e redefinir as categorias originais como $A' = A - C$ e $B' = B - C$. Todavia, esse processo tende a aumentar a complexidade geral da taxonomia. Para mitigar esse problema, os desenvolvedores podem restringir o escopo ou refinar o objetivo da taxonomia. Embora algumas aplicações não exijam exclusividade mútua, sua importância tem crescido, especialmente com o uso crescente de técnicas de aprendizado de máquina (Machine Learning – ML). Assim, neste estudo, a propriedade *mutuamente exclusiva* é considerada um requisito obrigatório, mesmo que em trabalhos anteriores tenha sido tratada como apenas desejável [35].

A propriedade *versátil* refere-se à capacidade de uma taxonomia adaptar-se à evolução do cenário de ameaças, permitindo a inclusão de novas categorias [94]. Essa característica, também denominada "extensível" ou "atualizável", requer que os procedimentos de atualização sejam claramente documentados. Embora essencial em um campo dinâmico como a segurança cibernética, a extensibilidade pode introduzir ambiguidades e comprometer a *compreensibilidade* e a *repetibilidade*. Além disso, modificações podem impactar alguns tipos de aplicações, como sistemas baseados em aprendizado de máquina, que exigiriam retreinamento dos modelos. Diante da possibilidade de comprometer requisitos anteriormente satisfeitos, corromper sistemas dependentes, reprocessar conjuntos de dados rotulados e retreinar modelos de ML, a *versatilidade* é tratada nesta pesquisa como um requisito desejável, mas não obrigatório — representado pela seta tracejada na Figura 2.1.

Nesta seção, foram compiladas, analisadas e discutidas as principais propriedades das taxonomias, com o objetivo de identificar os requisitos mínimos obrigatórios para taxonomias de ataques, conforme resumido na Tabela 2.2. Para serem úteis e amplamente aceitas pela comunidade, tais taxonomias devem ser *completas*, *compreensíveis*, *repetíveis*

e *mutuamente exclusivas*. A próxima seção discute sobre aspectos estruturais observados nas taxonomias de ataques, com ênfase em suas características, benefícios e limitações.

Tabela 2.2: Requisitos das taxonomias de ataque

Requisito	**Obrigatório**	**Contribui para ser**	**Depende das propriedades** (descritas na Tabela 2.1)
Versátil	não	aceita	versátil.
Mutuamente exclusiva	sim	aceita	mutuamente exclusiva.
Útil	sim	aceita	completa, compreensível, repetível.
Completa	sim	útil	exaustiva, representatividade humana, ameaças internas *versus* externas.
Compreensível	sim	útil	objetivo, termos bem definidos, terminologia em conformidade com a terminologia de segurança estabelecida , inequívoca, específica, apropriada.
Repetível	sim	útil	similaridade, determinismo, primitiva, objetividade, inequívoca, específica, apropriada.

2.2 Estrutura das Taxonomias

Uma taxonomia é essencialmente um conjunto de categorias que compartilham propriedades comuns e são organizadas de forma estruturada. A análise de sua estrutura oferece insights sobre fatores que afetam diretamente sua eficácia e aplicabilidade, sendo útil para desenvolvedores, usuários e pesquisadores. No contexto desta pesquisa, tal análise permite classificar com maior precisão as taxonomias estudadas e compreender as razões pelas quais determinadas não satisfazem certos requisitos.

Esta seção apresenta uma análise da estrutura das taxonomias de ataque sob aspectos como organização, esquema, rotulagem e abordagem. Para cada aspecto, são descritas e discutidas duas possibilidades de implementação, com ênfase nas diferenças entre elas e nas suas relações com as propriedades discutidas na seção anterior. Por fim, são analisadas as principais opções de visualização empregadas pelos autores na apresentação de suas taxonomias, destacando-se as vantagens e limitações de cada uma e sua relação com os atributos estruturais da taxonomia.

2.2.1 Organização: linear *versus* hierárquica

A estrutura organizacional mais simples de uma taxonomia é uma lista de termos ou categorias, chamada de *linear* (ou horizontal) por conter uma disposição plana de classes únicas, conforme ilustrado na Figura 2.2. Dois exemplos de taxonomias lineares de ataque podem ser encontrados em [97] e [98].

Taxonomias lineares são simples, porém limitadas, pois sua utilidade se restringe à compreensão das características de um ataque [35]. O principal desafio em seu desenvolvimento está na definição dos termos [89, 38]. Termos mais abrangentes aumentam a

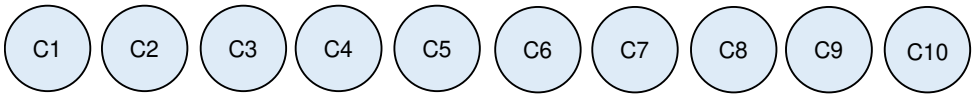


Figura 2.2: Organização linear (exemplo com 10 classes).

cobertura de ataques, mas podem introduzir ambiguidade e sobreposição; já termos muito restritos resultam em categorias *não exaustivas*. Além disso, taxonomias lineares não expressam relações entre diferentes tipos de ataques [89, 38], o que limita sua finalidade [86]. Por esses motivos, nenhuma taxonomia linear obteve aceitação generalizada [89, 38].

Taxonomias *hierárquicas* (ou em camadas) surgiram para superar essas limitações. Sua estrutura multinível permite um processo de classificação mais claro e objetivo [35], podendo ser representada por uma árvore [99]. O processo de classificação percorre a árvore da raiz até a folha [86], com cada decisão determinando a categoria mais adequada até chegar à classe final. Cada folha da árvore representa uma categoria — referida como *classe* — possível para o ataque. Para maior eficácia, uma taxonomia hierárquica deve iniciar com categorias gerais e refinar-se progressivamente em categorias mais específicas [35], conforme ilustrado na Figura 2.3.

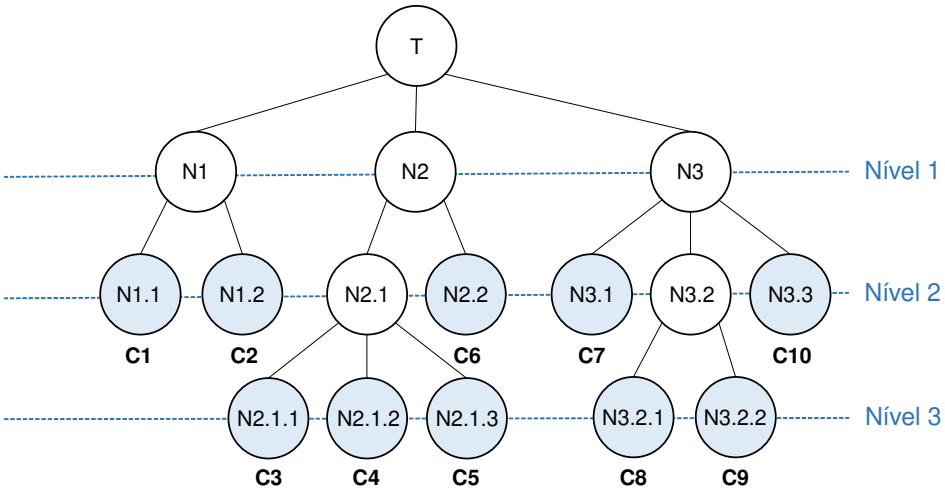


Figura 2.3: Organização hierárquica (exemplo com 3 níveis, 15 nós e 10 classes).

Embora a organização hierárquica seja mais complexa do que a linear (compare Figuras 2.2 e 2.3), sua estrutura em árvore facilita o processo de decisão para uma classificação mais preciso e eficiente [42]. É mais confiável tomar várias decisões simples do que uma única decisão com múltiplas possibilidades.

2.2.2 Esquema: plano *versus* multidimensional

As primeiras taxonomias buscavam classificar ataques segundo um único critério geral [92], sendo chamadas de *planas* ou *unidimensionais*. Exemplos incluem [100, 101, 102,

103, 104]. Embora adequadas em contextos simples, elas se mostram insuficientes para ataques combinados, compostos por múltiplos subataques que exploram diferentes vulnerabilidades [81, 92]. Referências cruzadas ou árvores recursivas tornariam tais taxonomias confusas e pouco compreensíveis [72]. Para tratar ataques complexos mantendo a clareza estrutural, passou-se a adotar o esquema *multidimensional*.

O conceito de *dimensão* [88] — comum em áreas como matemática e física — é aqui empregado para representar um aspecto, característica ou atributo do ataque. Por exemplo, a taxonomia de Hansman [72] classifica ataques segundo quatro dimensões: vetor[1], alvo, vulnerabilidade e *payload*.

Cada dimensão funciona como uma taxonomia plana independente, geralmente hierárquica, com suas próprias categorias. Combinadas, oferecem uma visão mais holística do ataque [72]. A Figura 2.4 ilustra uma taxonomia com quatro dimensões, sendo a 1ª, 2ª e 4ª hierárquicas (dois níveis cada) e a 3ª linear.

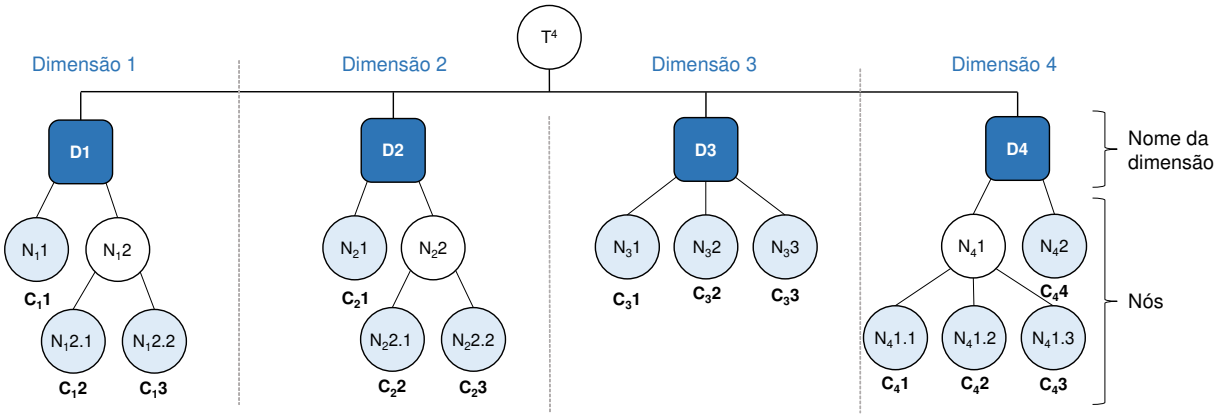


Figura 2.4: Esquema multidimensional (exemplo com 4 dimensões).

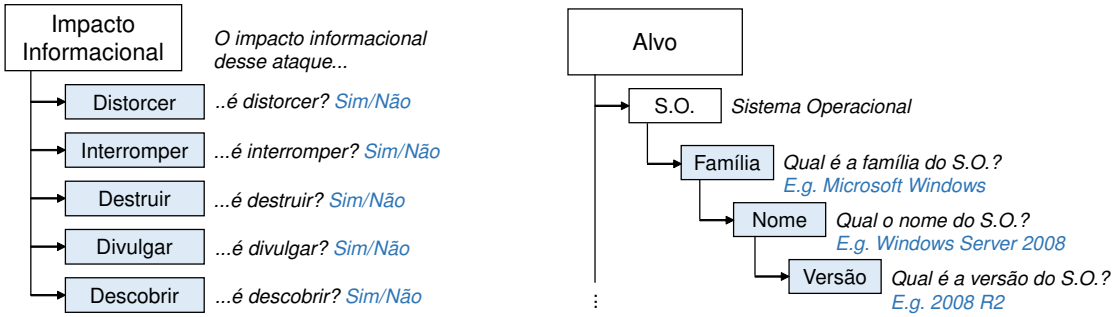
A classificação de um ataque em uma taxonomia multidimensional *mutuamente exclusiva* pode ser representada matematicamente por uma n-tupla $(d_1, d_2, \ldots, d_n)$, em que d_i é a classe escolhida na dimensão D_i de uma taxonomia n-dimensional T^n. O número total de classificações possíveis é obtido pelo produto do número de classes em cada dimensão. Assim, uma taxonomia multidimensional tem maior capacidade de classificação do que uma plana com o mesmo número total de classes. Por exemplo, uma taxonomia plana com 13 classes distingue no máximo 13 ataques, enquanto a taxonomia T^4 da Figura 2.4, também com 13 classes, pode classificar $3 \times 3 \times 3 \times 4 = 108$ ataques distintos. Contudo, o aumento de dimensões pode tornar a taxonomia mais complexa e difícil de manter, comprometendo alguns requisitos.

[1]Método usado para explorar o alvo [72].

2.2.3 Rotulagem: primitiva *versus* derivada

A definição das categorias é um dos aspectos mais críticos na construção de uma taxonomia, pois os rótulos representam atributos ou propriedades que caracterizam os ataques. Assim, a escolha da nomenclatura influencia diretamente a clareza conceitual e o processo de classificação.

Segundo Bishop [95], a formulação das categorias deve ser tão elementar quanto responder a uma pergunta de sim/não. As categorias que atendem a essa propriedade são denominadas *primitivas* e passam por um processo de classificação binária. Um exemplo de categorias *primitivas*, extraído da taxonomia AVOIDIT [1], é apresentado na Figura 2.5a. No processo de classificação dessas categorias, se o atributo observado corresponder ao rótulo da categoria, a resposta será "sim" e o rótulo será designado como valor da classe.



(a) Categorias primitivas (b) Categorias derivadas

Figura 2.5: Exemplo do processo de classificação de categorias primitivas *versus* derivadas [1].

Algumas categorias, entretanto, não são formadas por atributos ou características *primitivas*; em vez disso, consistem em termos genéricos que exigem a atribuição de um valor durante o processo de classificação. Essas categorias *não primitivas* são chamadas de *derivadas*, uma vez que o valor de classificação não é o rótulo em si, mas um novo valor derivado dele.

Conforme ilustrado na Figura 2.5b, as categorias "Família", "Nome" e "Versão" são *derivadas*, pois requerem a inclusão de informações detalhadas sobre o Sistema Operacional (SO) durante a classificação do Alvo na taxonomia AVOIDIT [1]. O processo de classificação das categorias *derivadas* utiliza perguntas do tipo "Qual/O quê/Por que/-Como" para determinar o valor da classe. No exemplo anterior, as perguntas poderiam ser "Qual é a família do SO?", "Qual é o nome do SO?" e "Qual é a versão do SO?", conforme mostrado na Figura 2.5b. As respostas a essas perguntas são, então, atribuídas como valores de classe para esse ramo da taxonomia.

Embora as categorias *primitivas* sejam predominantes nas taxonomias de ataque, alguns autores optam por incluir categorias *derivadas* como forma de conferir flexibilidade

ao modelo, uma vez que essas categorias não possuem valores fixos. Além disso, o uso de categorias *derivadas* pode tornar a taxonomia mais enxuta, pois requer menos níveis e, consequentemente, menos categorias para representar todos os atributos necessários. Essa característica simplifica o desenvolvimento de taxonomias, reduzindo o tempo necessário para definir categorias *exaustivas*.

Por outro lado, as categorias *derivadas* aumentam a complexidade do processo de classificação, pois, em vez de simplesmente selecionar uma categoria, o classificador precisa fornecer dados específicos. Isso limita sua aplicabilidade em sistemas automatizados, que passam a depender da interação humana para fornecer as informações solicitadas. O valor atribuído à categoria derivada fica, portanto, a critério do classificador. Isso significa que ataques semelhantes podem ser classificados de forma diferente, comprometendo a propriedade de *similaridade* e, consequentemente, o requisito de *repetibilidade*. Além disso, o número de classes em taxonomias *não primitivas* pode aumentar significativamente, já que cada valor distinto atribuído a categorias derivadas é considerado uma nova classe.

2.2.4 Abordagem: orientada por característica *versus* orientada por processo

Embora o objetivo das taxonomias de ataque seja sistematizar a classificação dos ataques [87], o escopo de aplicação pode variar amplamente. Dependendo do propósito, os desenvolvedores podem adotar duas abordagens distintas: *orientada por característica* ou *orientada por processo*. Essas abordagens influenciam diretamente a estrutura da taxonomia, pois determinam a escolha das dimensões e das categorias.

As informações consideradas pelo desenvolvedor ao definir os critérios de classificação variam conforme o escopo estabelecido. As taxonomias *orientadas por característica* possuem um escopo mais restrito, concentrando-se no ataque em si. Essa abordagem é mais objetiva e procura caracterizar o ataque de acordo com fatores técnicos, *i.e.* sem considerar elementos como autoria, motivação ou objetivo. Exemplos de taxonomias *orientadas por característica* podem ser encontrados em [72], [102] e [104].

Em contrapartida, as taxonomias *orientadas por processo* adotam uma perspectiva mais ampla, considerando o ciclo completo do ataque [72]. Essa abordagem incorpora informações sobre os autores, suas motivações, objetivos e possíveis contramedidas de defesa. Embora essa visão holística seja valiosa em determinados contextos, ela pode ter aplicabilidade limitada para entidades como o CERT (Computer Emergency Response Team), que estão mais interessadas no ataque em si do que em sua autoria ou intenções [72].

De modo geral, as taxonomias *orientadas por processo* tendem a ser mais subjetivas e a incluir categorias *derivadas*, o que pode afetar as propriedades de *similaridade* e *repetibilidade* no processo de classificação. Exemplos de taxonomias com essa abordagem podem ser encontrados em [38], [105] e [106], além de outros mencionados na Tabela 2.8.

2.3 Critérios de Avaliação de Taxonomias

Diversos pesquisadores têm explorado as características, propriedades, atributos e requisitos de taxonomias aplicadas à segurança cibernética. No entanto, observa-se uma notável escassez de propostas voltadas à avaliação sistemática dessas taxonomias. Para preencher essa lacuna, esta seção apresenta dois conjuntos de critérios de avaliação: o primeiro voltado à análise dos requisitos da taxonomia, conforme discutido na Seção 2.1, e o segundo à avaliação dos aspectos estruturais, conforme apresentados na Seção 2.2.

Ambos os conjuntos são interdependentes, visto que determinadas propriedades podem influenciar a estrutura da taxonomia, e vice-versa. Nesta pesquisa, tais critérios são utilizados para conduzir uma análise comparativa e objetiva das taxonomias de ataque, apresentada na Seção 2.4. Além disso, os critérios aqui propostos podem servir de referência tanto para desenvolvedores de novas taxonomias, auxiliando na verificação da qualidade de seus projetos, quanto para profissionais de segurança, apoiando a seleção da taxonomia de ataque mais adequada às suas necessidades específicas.

2.3.1 Critérios de avaliação dos requisitos da taxonomia

A Seção 2.1 apresentou uma análise abrangente das principais propriedades de taxonomias de segurança, resultando em uma lista de requisitos (Tabela 2.2) que uma taxonomia de ataques deve satisfazer para ser amplamente reconhecida pela comunidade de segurança cibernética. Nesta subseção, são considerados quatro requisitos mínimos obrigatórios, empregados como critérios fundamentais de avaliação: *completa*, *compreensível*, *repetível* e *mutuamente exclusiva*.

Para reduzir a subjetividade do processo avaliativo e aumentar sua precisão, propõem-se subcritérios baseados no diagrama de relacionamento das propriedades da taxonomia (Figura 2.1). Selecionam-se, assim, as propriedades aplicáveis à maioria dos escopos e alinhadas ao objetivo de decompor critérios amplos em componentes menores e verificáveis. A Tabela 2.3 apresenta os critérios e subcritérios correspondentes.

Para que uma taxonomia seja considerada *completa*, é necessário que abranja todos os ataques dentro do escopo definido, demonstrando-se exaustiva. Embora seja difícil contemplar todas as possibilidades de ataque, a simples identificação de um único ataque não coberto é suficiente para comprovar a incompletude da taxonomia.

Tabela 2.3: Critérios e subcritérios para avaliar os requisitos da taxonomia

Critério	**Valor**	**Subcritérios**
Completa	sim/não	Exaustiva e representatividade humana
Compreensível	sim/não	Termos bem definidos, terminologia em conformidade com a terminologia de segurança estabelecida, apropriada e inequívoca
Repetível	sim/não	Similaridade, primitiva e inequívoca
Mutualmente exclusiva	sim/não	-

Adicionalmente, inclui-se o subcritério de *representatividade humana*, que, embora relacionado ao anterior, orienta a verificação quanto à cobertura de ataques que exploram o fator humano — por exemplo, técnicas de engenharia social. Caso esse subcritério não seja atendido, o requisito de completude também não o será.

Para ser *compreensível*, a taxonomia deve satisfazer quatro subcritérios: *termos bem definidos*, *terminologia em conformidade com a terminologia de segurança estabelecida*, *apropriada* e *inequívoca*. Embora semelhantes em certos aspectos, esses subcritérios diferem quanto ao impacto na clareza e na interpretação. Para auxiliar o avaliador, são apresentados quatro cenários hipotéticos que ilustram casos de falha na conformidade desses subcritérios.

1. Suponha uma categoria chamada "Vetor de ataque", que contém apenas classes pertencentes a vulnerabilidades. Assim, pode-se inferir que a categoria em questão não estava bem definida, pois deveria ter sido claramente chamada de "Vulnerabilidade".
2. Se a mesma categoria, "Vetor de ataque", contiver tanto classes de vulnerabilidade quanto classes de vetor, fica evidente que o desenvolvedor confundiu os conceitos de vulnerabilidade e vetor de ataque, desviando-se assim da definição estabelecida na comunidade de segurança [107]. Portanto, neste caso, a *terminologia não está em conformidade com a terminologia de segurança estabelecida.*
3. Se a categoria "Vulnerabilidade" existir na taxonomia, pode-se concluir que as classes relacionadas a vulnerabilidades foram atribuídas indevidamente à categoria "Vetor de ataque". Assim, neste caso, a taxonomia *não* é *apropriada.*
4. Suponha uma taxonomia com duas classes denominadas "Permissão incorreta" e "Configuração incorreta" situadas na mesma dimensão, mas em ramos distintos da árvore. Neste caso, a taxonomia não deve ser avaliada como *inequívoca*, porque a primeira classe é um caso particular da segunda e, portanto, um ataque poderia ser classificado usando ambas as possibilidades.

O requisito *repetível* implica que diferentes profissionais possam classificar, de forma consistente, ataques semelhantes nas mesmas categorias. Para que isso ocorra, a taxonomia deve empregar termos estáveis e objetivos, satisfazendo os subcritérios *similaridade*, *primitiva* e *inequívoca.*

O critério *mutuamente exclusiva*, por sua vez, não demanda subcritérios, pois seu conceito é autoexplicativo. A avaliação pode ser conduzida pela busca de ataques classificados em mais de uma categoria. Uma vez encontrado um exemplo, o critério é considerado reprovado. Recomenda-se iniciar a verificação pelas categorias mais amplas e consultar a documentação da taxonomia para identificar se a exclusividade mútua é um requisito explícito do modelo.

A Tabela 2.4 apresenta um guia de avaliação dos requisitos e condições correspondentes, enquanto a Tabela 2.5 detalha as orientações para a verificação dos subcritérios. A próxima subseção discute os critérios estruturais.

Tabela 2.4: Guia para avaliar os requisitos de taxonomia

Critério	**Valor**	**Condição**
Completa	sim	Se a taxonomia satisfizer as propriedades: *exaustiva* e *representatividade humana*.
	não	Se a taxonomia não satisfizer pelo menos uma das propriedades acima.
Compreensível	sim	Se a taxonomia satisfizer as propriedades: *termos bem definidos*, *terminologia em conformidade com a terminologia de segurança estabelecida*, *apropriada* e *inequívoca*.
	não	Se a taxonomia não satisfizer pelo menos uma das propriedades acima.
Repetível	sim	Se a taxonomia satisfizer as propriedades: *similaridade*, *primitiva* e *inequívoca*.
	não	Se a taxonomia não satisfizer pelo menos uma das propriedades acima.
Mutuamente exclusiva	sim	Se a taxonomia tiver categorias exclusivas que permitem que os ataques sejam classificados em apenas uma classe em cada dimensão.
	não	Se a taxonomia classificar qualquer ataque com mais de uma classe na mesma dimensão.

Tabela 2.5: Orientação sobre como avaliar os subcritérios

Subcritério	**Processo de avaliação**
Exaustiva	É reprovado se houver um ataque que não se encaixe em nenhuma categoria.
Representatividade humana	Falha se não houver categorias para cobrir ataques que exploram o elemento humano. (Ignorar se estiver fora do escopo da taxonomia.)
Termos bem definidos	Falha se a terminologia de qualquer rótulo não representar corretamente as categorias ou classes subjacentes.
Terminologia em conformidade com a terminologia de segurança estabelecida	É reprovado se qualquer termo usado nos rótulos não for o mais apropriado para representar as categorias ou classes subjacentes.
Apropriada	É reprovado se qualquer categoria tiver uma posição mais apropriada dentro da árvore taxonômica.
Inequívoca	É reprovado se houver rótulos com significados duvidosos no mesmo ramo ou cobertura sobreposta entre categorias em diferentes ramos da árvore, levando a múltiplas possibilidades de classificação.
Primitiva	Falha se houver pelo menos uma categoria que exija o fornecimento de algumas informações, como nome, versão ou outros dados relevantes.
Similaridade	Falha se houver a possibilidade de ataques semelhantes serem classificados de forma diferente.

2.3.2 Critérios de avaliação estrutural da taxonomia

Nesta subseção, os aspectos estruturais discutidos na Seção 2.2 são empregados como critérios qualitativos e quantitativos para avaliar as taxonomias de ataque. Os critérios qualitativos e respectivos valores são:

- Esquema: [*plano* / *multidimensional*]
- Organização: [*linear* / *hierárquica*]
- Rotulagem: [*primitiva* / *derivada*]
- Abordagem: [*orientada por característica* / *orientada por processo*]

A avaliação do *esquema* é direta, baseando-se na quantidade de dimensões da taxonomia. Cada dimensão representa um aspecto ou atributo usado na classificação dos ataques. Assim, se apenas uma dimensão for identificada, o esquema é *plano*; caso contrário, é *multidimensional*.

A *organização* é mais complexa de avaliar, pois uma taxonomia multidimensional pode apresentar dimensões de ambos os tipos (*linear* e *hierárquica*). Nesses casos, considera-se a organização como hierárquica, por representar o nível mais elevado de complexidade estrutural.

O critério de *rotulagem* também pode gerar conflitos. Embora a maioria das taxonomias de ataque utilize categorias *primitivas*, algumas apresentam ambas. Se houver ao menos uma categoria *derivada*, a taxonomia deve ser classificada como *derivada*, pois essa característica afeta diretamente propriedades como a *similaridade*, conforme discutido na Subseção 2.2.3.

O último critério qualitativo é a *abordagem*, que pode ser *orientada por característica* ou *orientada por processo*. Taxonomias que classificam ataques apenas com base em suas propriedades técnicas são consideradas *orientadas por característica*. Já aquelas que incorporam informações adicionais, como autoria, motivação, objetivo, defesa, ou outros atributos que ampliam o escopo da classificação, são classificadas como *orientadas por processo*, conforme detalhado na Subseção 2.2.4.

Além dos critérios qualitativos, propõem-se três critérios quantitativos que complementam a avaliação estrutural, fornecendo parâmetros de comparação entre diferentes estruturas:

- Número de dimensões ($\#D$): $[\, d > 0 \,]$
- Número de níveis ($\#L$): $[\, l > 0 \,]$
- Número de classes ($\#C$): $[\, c > 0 \,]$

O *número de dimensões* ($\#D$) corresponde ao número de características ou atributos considerados na classificação, conforme discutido na Subseção 2.2.2. Em uma taxonomia n-dimensional T^n, tem-se $\#D(T^n) = n$.

O *número de níveis* ($\#L$) reflete a profundidade hierárquica da taxonomia e deve considerar o maior nível entre as dimensões existentes. Se m representa o número de passos para percorrer da raiz até um nó folha da árvore em qualquer dimensão i, então $\#L(T^n) = \max(m)$.

O *número de classes* ($\#C$) expressa a capacidade de classificação da taxonomia, *i.e.*, o total de possibilidades de classificação. Considerando a exclusividade mútua, o número de classes em uma taxonomia plana T é dado por $\#C(T) = |T| = k$, onde k é a cardinalidade do conjunto de classes $\{c_1, c_2, \ldots, c_k\}$. Para uma taxonomia multidimensional, o número total é $\#C(T^n) = \prod_{i=1}^{n} |D_i|$, sendo D_i o conjunto de classes da dimensão i. Taxonomias derivadas podem permitir a criação de novas classes, tornando o número de possibilidades indeterminado. Entretanto, ao considerar cada categoria como primitiva, pode-se estimar um número mínimo de classificações possíveis.

Por fim, a Tabela 2.6 apresenta o guia consolidado para avaliação estrutural, reunindo critérios, valores e condições correspondentes. Na seção seguinte, esses critérios são aplicados na avaliação comparativa das taxonomias de ataque.

Tabela 2.6: Guia para avaliar a estrutura da taxonomia

	Critério	**Valor**	**Condição**
Qualitativo	Esquema	Plano	Se a taxonomia tiver apenas uma dimensão.
		Multidimensional	Se a taxonomia tiver duas ou mais dimensões.
	Organização	Linear	Se a taxonomia tiver todas as categorias no mesmo nível hierárquico.
		Hierárquico	Se a taxonomia tiver categorias com diferentes níveis hierárquicos.
	Rotulagem	Primitiva	Se a taxonomia tiver todas as categorias com rótulos que correspondem a características a serem verificadas durante o processo de classificação por meio de perguntas sim-não.
		Derivada	Se a taxonomia tiver pelo menos uma categoria com um rótulo que indique uma solicitação de informações que o operador precisa fornecer no momento da classificação.
	Abordagem	Orientada por característica	Se a taxonomia tiver todas as dimensões e categorias focadas nas características técnicas do próprio ataque.
		Orientada por processo	Se a taxonomia tiver pelo menos uma dimensão ou categoria que considere informações adicionais relacionadas ao processo de ataque, como autoria, motivação, objetivo, estratégia de defesa, etc.
Quantitativo	Número de dimensões	$\#D = n$	Onde n é o número de aspectos ou atributos exigidos pela taxonomia T^n para classificar ataques.
	Número de níveis	$\#L = max(m)$	Onde m é o m-ésimo nó na hierarquia, começando pelo nó do primeiro nível $N^i a_1$ até o nó folha $N^i a_1.b_2.\cdots.z_m$ para qualquer dimensão i da taxonomia T^n, tal que $1 \leq i \leq n$.
	Número de classes	$\#C = \prod_{i=1}^{n} \lvert D_i \rvert$	Onde $\lvert D_i \rvert$ é o número de classes (nós folha) da dimensão i na taxonomia T^n, tal que $1 \leq i \leq n$.

2.4 Revisões de Taxonomias

Ao longo dos anos, a comunidade de segurança da informação propôs inúmeras taxonomias de ataques, desenvolvidas com diferentes objetivos e aplicações. Cada uma delas é aplicável apenas a um contexto ou campo de interesse [35]. A literatura apresenta um número expressivo de taxonomias, que podem ser agrupadas em duas grandes categorias: gerais e específicas.

As *taxonomias gerais* abrangem de forma ampla os ataques contra redes e sistemas computacionais, enquanto as *taxonomias específicas* concentram-se em tipos particulares de ataque. Estas últimas podem estar relacionadas ao alvo — como sistemas em nuvem, aplicações Web, aplicações Peer-to-Peer (P2P) ou infraestrutura da Internet —, à metodologia de ataque — como negação de serviço (DoS), negação de serviço distribuída (DDoS) e golpes de *phishing* —, ou ainda a outros contextos definidos conforme o interesse dos autores. Em geral, taxonomias desenvolvidas para um sistema ou aplicação específica não são úteis em outros contextos [49]. Por outro lado, tendem a ser mais precisas para análise de ataques dentro do escopo em que foram concebidas, devido ao maior nível de detalhamento.

Nesta seção, são revisadas 20 taxonomias de ataques cibernéticos publicadas entre 2011 e 2022, considerando seus objetivos, estrutura, finalidade e requisitos. Para fins de organização, as taxonomias foram classificadas conforme o contexto de aplicação, na seguinte ordem: Rede e Computadores, Ataques DoS/DDoS, Ataques de *Phishing*, Sistemas em Nuvem, Aplicações Web, Aplicações P2P e Infraestrutura da Internet. Em seguida, são apresentadas duas tabelas comparativas — uma relativa à avaliação dos requisitos e outra à estrutura —, seguidas de suas análises. Por fim, são destacadas questões em aberto e desafios identificados a partir das taxonomias estudadas, bem como direções futuras para essa área.

2.4.1 Rede e computadores

Taxonomia de Chapman

Chapman *et al.* (2011) [104] apresentam uma taxonomia de ataques cibernéticos com base no nível de acesso necessário ao invasor para lançar o ataque. Sua taxonomia concentra-se em ataques amplos comumente observados em grandes redes de computadores, como a Internet. A intenção dessa taxonomia é facilitar a compreensão das ameaças cibernéticas para ajudar os defensores da rede a estruturar seus planos de defesa. Além disso, ela desempenha um papel didático, tornando os usuários da rede mais familiarizados com o jargão dos ataques cibernéticos e os níveis de penetração que diferentes ataques envolvem. Eles propõem uma taxonomia plana de dois níveis com três categorias na primeira ca-

mada: 'No Network or Computer Access', 'User Access with Limited Privileges', e 'Root Access/Administrative Privileges'.

De acordo com o artigo, essa taxonomia cobre uma parte dos possíveis ataques cibernéticos, o que significa que não é exaustiva e, portanto, não é completa. Embora abranja ataques de phishing, ela não possui uma categoria para engenharia social, portanto, também não satisfaz o critério de *representatividade humana*. As categorias 'Rootkit' e 'Kernel-level Rootkit' criam ambiguidade na classificação, pois os termos sugerem que o primeiro abrange o segundo. Categorias ambíguas tornam a taxonomia *não compreensível* e *não repetível*. Além disso, essa sobreposição pode levar à *não similaridade* na classificação e também torna a taxonomia *não mutuamente exclusiva*.

AVOIDIT

Simmons *et al.* (2014) [1] propõem uma taxonomia de ataques cibernéticos chamada AVOIDIT. Esta sigla é um acrônimo de 'Attack Vector', 'Operational Impact', 'Defense', 'Information Impact', e 'Target', que são as 5 dimensões da taxonomia. Ao incluir a classificação dos mecanismos de defesa na taxonomia de ataques, eles pretendiam ajudar os defensores não apenas a identificar os ataques cibernéticos, mas também a compreender a estratégia necessária para mitigar e/ou remediar a possível exploração. Em relação à organização da taxonomia, a dimensão 'Information Impact' é linear, mas as outras são hierárquicas, com até cinco níveis.

Apesar de cobrir a maioria dos ataques, os autores afirmam na documentação que o AVOIDIT não é *exaustivo* e, consequentemente, não é *completo*. Eles afirmam que são necessárias mais pesquisas para fornecer uma lista *exaustiva* de possíveis estratégias de defesa para cada vulnerabilidade explorada. O AVOIDIT também não é *inequívoco*, pois há sobreposição de algumas categorias, como 'Incorrect Permission' e 'Misconfiguration', em que a primeira pode ser considerada um caso particular da segunda. Além disso, o termo usado na primeira dimensão não está em conformidade com a terminologia estabelecida na comunidade, porque essa dimensão mistura vulnerabilidades e métodos de ataque, divergindo da definição de 'Attack Vector' [107], que deve conter apenas os métodos. Consequentemente, o AVOIDIT não é considerado *compreensível*. Algumas categorias, como 'Name' e 'Version' nos ramos 'OS', 'Server' e 'Client' da dimensão 'Attack Target', não são *primitivas* porque exigem que valores adicionais sejam atribuídos por quem está classificando. Atribuições arbitrárias dessas categorias *derivadas* podem levar a classificações diferentes para ataques semelhantes, afetando assim o requisito *repetível*. Também não é *mutuamente exclusiva* porque a dimensão 'Defense' permite a classificação simultânea nas duas categorias 'Mitigation' e 'Remediation'.

ADMIT

Joshi e Singh (2014) [105] propõem uma taxonomia chamada ADMIT. Esta sigla significa 'Attack vector', 'Defense', 'Method', 'Impact', e 'attack Target', que são as 5 dimensões da taxonomia. O esquema de classificação desta taxonomia foi projetado para descrever todo o processo de ataque, respondendo às seguintes perguntas: (i) "quem é o atacante?"; (ii) "como enfrentar o ataque?"; (iii) "como é o ataque?"; (iv) "quais são os resultados?" e (v) "qual é o alvo?" O objetivo do ADMIT é fornecer informações sobre a natureza do ataque para ajudar os administradores de rede a localizar as estratégias mais adequadas para proteger seu sistema contra vulnerabilidades que podem ser exploradas. Em relação à organização da taxonomia, apenas a dimensão 'Attack Target' é linear, as outras são hierárquicas com dois níveis.

A taxonomia ADMIT não é *completa*, pois não abrange todas as categorias de ataques, como engenharia social, portanto, não possui *representatividade humana* e nem é *exaustiva*. A dimensão 'Attack Vector', em vez de se referir apenas a caminhos potenciais, contém uma categoria 'Attacker', indicando que o termo da dimensão não está em conformidade com a terminologia de segurança estabelecida. Para cobrir o que a dimensão anterior deveria cobrir, existe uma dimensão chamada 'Method'. Isso é uma indicação de que os termos das dimensões não são *bem definidos* e, como resultado, o ADMIT não atende o requisito *compreensível*. Felizmente, essa ambiguidade nos termos da dimensão não afeta as categorias subjacentes, que permanecem *inequívocas*. Além disso, o processo não é *repetível* devido à presença de categorias *derivadas*, como 'Technology' e 'Process' na dimensão 'Attack Vector' e 'Operational System' na dimensão 'Target'. Por fim, o artigo ADMIT oferece exemplos de classificação em várias classes dentro das dimensões 'Defense' ou 'Impact', demonstrando que não é *mutuamente exclusiva*.

Taxonomia de Hindy

Hindy *et al.* (2020) [108] propõem uma taxonomia genérica e modular para ameaças à rede. Sua taxonomia categoriza os ataques à rede com base na origem do ataque, na principal camada afetada do modelo OSI (Open System Interconnection) [109] e se a ameaça é ativa ou passiva. De acordo com os autores, os ataques que afetam qualquer aspecto da mídia em que estão sendo executados, como informações ou desempenho, são considerados ativos; os ataques que coletam informações ou monitoram a rede são considerados ameaças passivas. O objetivo dessa taxonomia é ajudar os pesquisadores a construir IDSs (Sistemas de Detecção de Intrusão) e ferramentas capazes de detectar vários ataques. Além disso, ao associar ameaças ao modelo OSI, a taxonomia pode ser usada para construir conjuntos de dados melhores e, assim, melhorar a detecção de ataques,

reduzindo o número de falsos positivos dos IDSs atuais. A taxonomia tem três dimensões: 'Threat Sources', 'OSI Layer', e 'Mode'. O diagrama proposto organiza os ataques em uma árvore de acordo com as fontes de ameaça (Network, Host, Software, Physical, Human) e, simultaneamente, indica qual a camada OSI é o principal alvo. A dimensão 'Mode' é diferenciada graficamente por um quadro específico que indica quais ataques são ativos, passivos ou ambos.

Apesar da falta de profundidade em algumas categorias, como '1.1.2 Amplification', a taxonomia é *exaustiva* e, portanto, *completa*. A classe '1.1.1.1 Smurf' é atribuída à camada de transporte do modelo OSI, mas deveria ser atribuída à camada de rede, pois os ataques Smurf[2] usam o protocolo ICMP [110, 111]. Esta questão demonstra que a taxonomia não está em conformidade com o conhecimento prévio sobre ataques Smurf, impedindo uma classificação *apropriada*. Consequentemente, a taxonomia não é *compreensível*. Apesar desta limitação, é *repetível* e *mutuamente exclusiva*.

2.4.2 Ataques DoS/DDoS

Taxonomia de Bhardwaj

Bhardwaj et al. (2016) [112] introduzem uma taxonomia para classificar ataques DDoS por'Degree of automation', 'Exploitation of vulnerabilities', 'Attack rate dynamics', e 'Attack impact', que são as quatro dimensões dessa taxonomia. Eles a desenvolveram no contexto de ataques DDoS contra computação em nuvem com o objetivo de classificar de forma clara e simples os ataques DDoS com base em critérios técnicos que podem ser facilmente obtidos e medidos. Essa taxonomia tem como objetivo ajudar os profissionais de segurança a tomar decisões corretas e informadas para mitigar ataques DDoS e, assim, ajudá-los a reduzir o tempo de inatividade do serviço.

A taxonomia tem uma organização hierárquica de dois níveis e atende aos requisitos de ser textitcompleta, textitcompreensível e textitrepetível. No entanto, ela não é textitmutuamente exclusiva na dimensão 'Exploitation of vulnerabilities', pois há ataques que se enquadram em mais de uma classe simultaneamente.

Taxonomia de Masdari

Masdari e Jalali (2016) [113] propõem uma taxonomia para ataques DoS em ambiente de computação em nuvem. O objetivo dessa taxonomia é proporcionar uma melhor compreensão desses ataques, suas propriedades e capacidades, para permitir que os profissionais de segurança investiguem melhores soluções para prevenir, detectar ou lidar com cada

[2]Visão geral do ataque Smurf em `https://www.cloudflare.com/learning/ddos/smurf-ddos-attack/`

tipo de ataque DoS na nuvem. A proposta tem uma estrutura plana e hierárquica e visa classificar os ataques DoS que têm como alvo a infraestrutura de rede ou componentes da nuvem. Esta taxonomia atende aos quatro requisitos obrigatórios, considerando o escopo proposto. No entanto, como o campo da segurança é vasto e dinâmico, são necessárias avaliações periódicas, especialmente no que diz respeito à completude.

Taxonomia de De Donno

De Donno *et al.* (2017) [114] propõem uma taxonomia abrangente de ataques DDoS com base em taxonomias anteriores e trabalhos recentes sobre ataques DDoS, especialmente aqueles relacionados a dispositivos IoT (Internet das Coisas). Na verdade, devido à alta disponibilidade de dispositivos IoT facilmente "hackeáveis", os ataques DDoS se tornaram mais populares entre os cibercriminosos, bem como mais poderosos. A taxonomia proposta tem 14 dimensões organizadas hierarquicamente e pretende ser uma referência completa e atualizada para os profissionais de segurança compreenderem todos os tipos de ataques DDoS.

Esta taxonomia é *exaustiva* e, portanto, *completa*. De acordo com os autores, a categoria 'Network Level' da dimensão 'Protocol Level' inclui protocolos da camada de transporte, mas seu rótulo não indica isso. Isso significa que o termo não foi *bem definido* e pode levar a uma classificação incorreta. Portanto, a taxonomia não pode ser considerada *compreensível*. No entanto, os requisitos restantes, *repetível* e *mutuamente exclusivo*, são cumpridos.

Taxonomia de Sharafaldin

Sharafaldin *et al.* (2019) [115] propõem uma taxonomia plana de três níveis para classificar ataques DDoS baseados em reflexão e exploração. Eles se concentram em ataques da camada de transporte e aplicação, que são realizados usando protocolos TCP/UDP. Ao analisar e categorizar os ataques DDoS, os autores conseguiram gerar um novo conjunto de dados, denominado CICDDoS2019, para avaliar algoritmos e sistemas IDS em ataques DDoS. Este conjunto de dados tem perfis de ataque baseados na taxonomia proposta.

As classes 'MSSQL[3] e 'SSDP[4] são atribuídas à categoria 'TCP based attacks' em 'Reflection attacks', mas deveriam ser atribuídas à categoria 'UDP based attacks' para estar em conformidade com o conhecimento prévio sobre esses ataques [116, 117]. Isso significa que a taxonomia não é capaz de classificar adequadamente esses ataques, o que a torna

[3]Visão geral do ataque Microsoft SQL (MS SQL) em `https://ddos-guard.net/en/terms/ddos-attack-types/ms-sql-attack`

[4]Visão geral do ataque Simple Service Discovery Protocol (SSDP) em `https://www.cloudflare.com/learning/ddos/ssdp-ddos-attack/`

pouco *compreensível*. A ausência das classes 'MSSQL' e 'SSDP em 'UDP based attacks' poderia nos levar à conclusão de que a taxonomia não é *exaustiva*, mas seria um erro de avaliação, pois, na verdade, essas classes pertencem à taxonomia, mas foram atribuídas à categoria errada. Portanto, ainda podemos considerá-la *completa*. Além disso, o rótulo 'TCP/UDP based attacks' abrange os termos de outras categorias no mesmo ramo, o que pode causar confusão durante o processo de classificação. Essa ambiguidade reforça que a taxonomia não é *compreensível* e mostra que também não é *repetível*. Apesar dessas questões, as classes são *mutuamente exclusivas*.

Taxonomia de Thorat

Thorat *et al.* (2021) [118] apresentam uma versão refinada da taxonomia de Sharafaldin [115], mantendo sua estrutura hierárquica e preservando 11 das 13 classes originais. Eles reposicionaram a classe 'MSSQL' sob 'UDP-based attacks' no ramo 'Reflection Attacks', já que os servidores MSSQL lidam principalmente com solicitações UDP [118]. Usando essa taxonomia, eles propõem o TaxoDaCML (Taxonomy-based Divide and Conquer using Machine Learning), que decompõe o problema de classificação em subproblemas menores antes de aplicar técnicas de Aprendizado de Máquina para detecção de ataques DDoS com alta precisão.

Essa taxonomia não é *exaustiva* porque, ao contrário da taxonomia de Sharafaldin [115], ela não cobre o ataque SSDP. A ausência da classe 'SSDP em 'UDP' no ramo 'Reflection' a torna incompleta. As categorias 'TCP/UDP' e 'UDP' têm rótulos ambíguos, uma vez que a primeira inclui o termo da segunda. Isso pode levar a confusão durante o processo de classificação, portanto, não podemos considerar a taxonomia *compreensível* ou *repetível*. Apesar dessas questões, a taxonomia é *mutuamente exclusiva*.

2.4.3 Ataques de phishing

Taxonomia de Aleroud

Aleroud e Zhou (2017) [119] propõem uma taxonomia de ataques de phishing que aborda 'Attack Techniques', 'Target Devices', 'Countermeasures', e 'Communication Media'. Com essa taxonomia, os autores pretendem facilitar aos profissionais de segurança a escolha e a avaliação de métodos e ferramentas para lidar com ataques de phishing específicos, bem como ajudar os profissionais a desenvolver técnicas eficazes para a prevenção e detecção de phishing. Ao identificar vetores de ataque emergentes, eles levantam questões em aberto para futuras pesquisas e desenvolvimentos na detecção e prevenção de ataques de phishing. A taxonomia proposta tem um esquema de 4 dimensões, baseado em ati-

vidades do processo, para fornecer uma visão integrada dos ataques de phishing e das contramedidas.

A taxonomia não é *exaustiva*, uma vez que não cobre todas as possibilidades de ataque dentro do escopo proposto, por exemplo, a dimensão 'Attack Techniques' não tem uma classe que o ataque QRLJacking[5] [120] se enquadre. Consequentemente, a taxonomia não é *completa*. Todos os outros requisitos são satisfatoriamente atendidos.

Taxonomia de Gupta

Gupta *et al.* (2018) [121] propõem uma taxonomia plana organizada hierarquicamente para classificar vários tipos de ataques de phishing com base nos mecanismos usados pelos invasores para obter informações pessoais sobre a vítima. Eles também usam os ataques identificados nesta taxonomia como base para projetar uma taxonomia de técnicas de detecção e proteção contra phishing. Com essas taxonomias, os autores pretendem ajudar novos pesquisadores e profissionais de segurança a compreender os tipos de ataques de phishing e as soluções disponíveis para proteger os usuários contra esses ataques.

A categoria 'Technical Subterfuge' não tem nenhuma classe adequada para enquadrar ataques do tipo Cross Site Request Forgery (CSRF)[6] [122, 123] por exemplo, provando que a taxonomia não é *exaustiva* e, portanto, não é *completa*. No entanto, a taxonomia é *compreensível*, *repetível* e *mutuamente exclusiva*.

Taxonomia de Rastenis

Rastenis *et al.* (2020) [124] propõem uma taxonomia de ataques de phishing por e-mail, organizada por fases de ataque. Cada uma das seis fases do processo tem pelo menos um critério (dimensão) para caracterizar o ataque. A proposta tem uma organização hierárquica com um total de 12 dimensões, a fim de classificar toda a gama de ataques de phishing baseados em e-mail. Além de destacar a variedade desses ataques, essa taxonomia visa permitir uma compreensão inequívoca do panorama dos ataques, contribuindo assim para a segurança pessoal e organizacional. Uma vantagem dessa taxonomia é seu uso como notação de ataque de phishing para sistemas de gerenciamento de incidentes, pois aumenta o nível de descrição do ataque em comparação com a notação de forma livre.

Essa taxonomia é *completa*, *compreensível* e *repetível*. Ela não é *mutuamente exclusiva* porque, de acordo com os autores, os ataques sistêmicos podem realizar mais de uma

[5]Visão geral do ataque Quick Response Code Login Jacking (QRLJacking) em `https://www.infosecinstitute.com/resources/hacking/qrl-jacking/`

[6]Visão geral dos ataques Cross Site Request Forgery (CSRF) em `https://owasp.org/www-community/attacks/csrf`

ação adicional entre as listadas, se encaixando simultaneamente em mais de uma classe na dimensão 'Systematic Strategy'.

2.4.4 Sistemas em nuvem

CATRA

Juliadotter e Choo (2015) [96] propõem uma taxonomia abrangente e extensível de ataques a serviços de computação em nuvem chamada CATRA (Cloud Attack Taxonomy and Risk Assessment). Além disso, eles a utilizam como uma estrutura de avaliação de riscos, atribuindo parâmetros de classificação de risco a algumas classes. Seguindo a taxonomia da esquerda para a direita, os autores pretendem criar cenários para compreender e avaliar os riscos de um ataque a serviços em nuvem, a fim de identificar áreas de preocupação a serem tratadas com técnicas de gerenciamento de riscos. Essa taxonomia utiliza seis critérios básicos como dimensões: 'Source', 'Vector', 'Vulnerability', 'Target', 'Impact' and 'Defense'. No entanto, alguns deles são divididos em critérios secundários que atuam como subdimensões. A estrutura resultante corresponde a uma taxonomia com 13 dimensões. Ela também possui uma organização hierárquica.

Essa taxonomia não é *exaustiva* e, portanto, não é *completa*, porque não há uma classe na dimensão 'Vetor' adequada para ataques de *buffer overflow*, por exemplo. Ela também não é *repetível* porque possui algumas classes *não primitivas*, cujos valores são definidos pelo operador durante a classificação. Apesar de tornar a taxonomia mais flexível, essa propriedade leva a uma perda de *similaridade* na classificação de ataques semelhantes por diferentes profissionais. No entanto, a taxonomia permanece *compreensível*. Ela não é *mutuamente exclusiva*, porque, de acordo com a documentação, os ataques devem ser classificados em mais de uma classe nas dimensões 'Vulnerability' e 'Defense'.

Taxonomia de Juliadotter

Juliadotter e Choo (2015) [106] apresentam uma taxonomia simplificada baseada em seu trabalho anterior [96]. Trata-se de uma taxonomia conceitual de ataques à nuvem com 5 dimensões que seguem o fluxo natural de um ataque a um serviço em nuvem. A organização é linear porque há apenas um nível de classes em cada dimensão. Como no trabalho anterior, eles usam a taxonomia proposta para avaliação de risco, atribuindo uma classificação de risco entre 1 (baixo risco) e 9 (alto risco) para quantificar a probabilidade geral estimada e o impacto geral de um determinado cenário de ataque. O cenário de ataque é o conjunto de informações fornecidas pelo profissional de segurança quando orientado pela taxonomia apresentada.

Essa taxonomia é *completa* e *compreensível*, mas não é *repetível* devido à inclusão de classes *derivadas*, como 'Ease of exploit' e 'Ease of discovery' na dimensão 'Vector', que exigem a inserção de valores. Também não é *mutuamente exclusiva*, pois requer mais de uma classe para classificar os ataques na dimensão 'Source', por exemplo.

Taxonomia de Iqbal

Iqbal *et al.* (2016) [125] propõem uma taxonomia de ataques à segurança na nuvem com base nos modelos de prestação de serviços de computação em nuvem (IaaS, PaaS, SaaS) [126, 127]. Com essa taxonomia, Iqbal *et al.* visa identificar e compreender os ataques potenciais ao ambiente de computação em nuvem. Eles apresentam uma taxonomia plana com uma organização hierárquica de 2 níveis, que não é *exaustiva* porque não cobre todas as possibilidades de ataques, considerando o escopo proposto. Por exemplo, a dimensão 'Security attacks on SaaS cloud layer' não possui classes adequadas para enquadrar ataques de *Cookie Poisoning*[7]. Embora não seja *completa*, a taxonomia é *compreensível*, *repetível* e *mutuamente exclusiva*.

Taxonomia de Mishra

Mishra *et al.* (2017) [87] propõem uma taxonomia de ataques em ambiente de nuvem com base no componente alvo. Com essa taxonomia, os autores pretendem elucidar as vulnerabilidades da nuvem, especialmente aquelas no ambiente virtualizado, e, em seguida, realizar um estudo sobre a capacidade das técnicas de detecção de intrusão no ambiente de nuvem. Eles também apresentam um modelo de ameaça, que foi usado para identificar alvos potenciais no ambiente. Trata-se de uma taxonomia unidimensional de dois níveis, em que o primeiro nível categoriza os ataques de acordo com a camada do ambiente de nuvem que eles exploram e o segundo é o próprio ataque. Essa taxonomia atende a todos os quatro requisitos obrigatórios, sendo *completa*, *compreensível*, *repetível* e *mutuamente exclusiva*.

Taxonomia de Akshaya

Swathy Akshaya e Padmavathi (2019) [128] apresentam uma taxonomia de ataques à segurança na nuvem com base nos modelos de prestação de serviços de computação em nuvem (IaaS, PaaS, SaaS) [126, 127]. Os autores pretendem fornecer uma compreensão aprofundada dos requisitos de segurança na nuvem. Eles também abordam a avaliação de riscos usando o fluxo da taxonomia de 5 dimensões proposta por Juliadotter e Choo

[7]Visão geral do ataque *Cookie Poisoning* em `https://www.gcst.ae/understanding-cookie-poisoning-attacks/`

[106]. A taxonomia de Akshaya tem a mesma estrutura que a taxonomia de Iqbal [125], com a mesma divisão no primeiro nível, mas apresenta diferenças em algumas classes.

Essa taxonomia não é *exaustiva* porque, ao contrário da taxonomia de Iqbal [125], ela não cobre ataques do tipo *VM Escape*[8] [129, 130] dentro da categoria 'Infrastructure as a Service'. Portanto, a taxonomia não é *completa*. Também não é *compreensível*, pois a definição de alguns termos de classe não é clara, *e.g.* 'Programming Attack' e 'Return Oriented'. Esses termos não são *bem definidos* e não *estão em conformidade com a terminologia de segurança estabelecida*. Além disso, os ataques ROP[9] [131] podem ser classificados em ambas as classes, indicando ambiguidade em vez de exclusividade mútua. A ambiguidade, neste caso, pode levar a ataques semelhantes serem classificados de forma diferente, dependendo de quem está realizando o processo de classificação. Outro exemplo de ambiguidade nesta taxonomia são as classes 'Web or Browser Security Attack' e 'XML signature Attack/Cross-site Scripting' da categoria 'Software as a Service', porque ambas as classes abrangem Cross Site Scripting (XSS)[10] [132, 133]. Consequentemente, a taxonomia não é *repetível* e *mutuamente exclusiva*.

2.4.5 Aplicações Web

Taxonomia de Singh

Singh *et al.* (2019) [134] propõem uma taxonomia de ataques em aplicações baseadas na web. Eles a utilizam para discutir os ataques mais comuns a aplicações web e seus impactos, com o objetivo de compreender as principais ameaças e a importância de implementar medidas preventivas. A proposta é plana, hierárquica e categoriza os ataques de acordo com o que eles exploram, como validação de entrada, autenticação, sessão ou dados confidenciais.

Essa taxonomia não é *completa*, pois não abrange ataques de XML malformados[11] [135]. No entanto, a taxonomia é *compreensível* e *repetível*. Como existem ataques que podem ser classificados em duas classes simultaneamente, por exemplo, 'SQL Injection' e 'Accessing sensitive data', ou 'Cross Site Scripting (XSS)' e 'Data Tempering', a taxonomia não pode ser considerada *mutuamente exclusiva*.

[8]Definição de *VM Escape* em `https://www.techtarget.com/whatis/definition/virtual-machine-escape`

[9]Definição de Return-Oriented Programming (ROP) em `https://help.eset.com/glossary/en-US/rop_attack.html`

[10]Visão geral do ataque XSS em `https://owasp.org/www-community/attacks/xss/`

[11]Visão geral do ataque de XML malformado (Malformed XML) em `https://www.soapui.org/docs/security-testing/security-scans/malformed-xml/`

2.4.6 Aplicações P2P

Taxonomia de Koutrouli

Koutrouli e Tsalgatidou (2012) [136] propõem uma taxonomia para classificar adequadamente os vários ataques contra sistemas de confiança baseados em reputação para aplicações P2P. Por meio dessa taxonomia, os autores pretendem fornecer uma visão geral abrangente das ameaças de credibilidade existentes contra um sistema de reputação descentralizado, a fim de compreender melhor suas fraquezas e ajudar os desenvolvedores a se concentrarem em como se defender contra esses ataques específicos. Eles também mapeiam as classes de ataque da taxonomia proposta para os mecanismos de defesa disponíveis na literatura, a fim de revelar possíveis lacunas que precisam de mais pesquisas para desenvolver novas proteções contra ataques à reputação. A proposta tem uma estrutura plana e hierárquica.

As categorias 'Bad mouthing' e 'Collusive deceit' no ramo 'Unfair Recommendations' têm apenas uma classe filha cada. Isso pode indicar que as classes não são *exaustivas*, porque geralmente as categorias são divididas em mais de uma classe, formando uma árvore de decisão que facilita o processo de classificação. Na documentação, os autores afirmam que essas classes representam casos específicos do ataque representado pela categoria pai, o que implica que pode haver outros ataques dentro da mesma categoria. Portanto, consideramos essa taxonomia não *completa*. No entanto, ela é *compreensível*, *repetível* e *mutuamente exclusiva*.

2.4.7 Infraestrutura da internet

Taxonomia de Hollick

Hollick *et al.* (2017) [137] propõem uma taxonomia de ataques para classificar possíveis ataques a protocolos de roteamento seguro. Ao decompor o sistema de roteamento, eles fornecem uma taxonomia centrada em suas funções/serviços fundamentais, como roteamento, topologia, transporte e resolução de identidade. O objetivo dessa taxonomia plana de dois níveis é compreender melhor os protocolos de roteamento seguros existentes e fornecer uma estrutura para projetar novos protocolos. Os autores acreditam que uma visão estruturada pode ajudar a responder a importantes questões de segurança sobre sistemas de roteamento. Essa taxonomia é *completa*, *compreensível*, *repetível* e *mutuamente exclusiva*, satisfazendo os quatro requisitos obrigatórios.

2.4.8 Resumo comparativo

Nesta subseção, apresenta-se o resumo dos resultados das revisões das taxonomias de ataques, com base nos critérios definidos na Seção 2.3. As taxonomias são construídas sobre um domínio bem delimitado e, portanto, seus atributos são avaliados levando em consideração o contexto e os objetos desse domínio. Os resultados são organizados em tabelas comparativas, de modo a facilitar a análise e apoiar profissionais de segurança na escolha da taxonomia mais adequada às suas necessidades. Assim, as taxonomias foram agrupadas conforme o contexto de cobertura.

A Tabela 2.7 compara as taxonomias com base nos critérios de avaliação de requisitos. O resultado é expresso em termos binários ("*sim*" ou "*não*") para cada um dos quatro requisitos obrigatórios: *Completa*, *Compreensível*, *Repetível* e *Mutualmente Exclusiva*. Para fundamentar os resultados, a tabela também apresenta os subcritérios considerados na avaliação. Um requisito recebe a marcação "*sim*" apenas quando todos os seus subcritérios são atendidos (✓). Subcritérios não aplicáveis a determinada taxonomia ou contexto são representados por um traço (–) e não influenciam o resultado final.

Tabela 2.7: Comparação de taxonomias de ataque pela avaliação de requisitos

Taxonomia	**Ano**	**Contexto**	Exaustiva	Representatividade humana	**Completa**	Termos bem definidos	Terminologia em conformidade	Apropriada	Inequívoca	**Compreensível**	Similaridade	Primitiva	Inequívoca	**Repetível**	**Mutualmente Exclusivo**
Taxonomia de Chapman [104]	2011	Rede e Computadores	✗	✓	**não**	✓	✓	✓	✗	**não**	✗	✓	✗	**não**	**não**
AVOIDIT [1]	2014		✗	✓	**não**	✓	✗	✓	✗	**não**	✗	✗	✗	**não**	**não**
ADMIT [105]	2014		✗	✗	**não**	✗	✗	✓	✓	**não**	✗	✗	✓	**não**	**não**
Taxonomia de Hindy [108]	2020		✓	✓	**sim**	✓	✗	✗	✓	**não**	✓	✓	✓	**sim**	**sim**
Taxonomia de Bhardwaj [112]	2016	Ataques DoS/DDoS	✓	–	**sim**	✓	✓	✓	✓	**sim**	✓	✓	✓	**sim**	**não**
Taxonomia de Masdari [113]	2016		✓	–	**sim**	✓	✓	✓	✓	**sim**	✓	✓	✓	**sim**	**sim**
Taxonomia de De Donno [114]	2017		✓	–	**sim**	✗	✓	✓	✓	**não**	✓	✓	✓	**sim**	**sim**
Taxonomia de Sharafaldin [115]	2019		✓	–	**sim**	✓	✗	✗	✗	**não**	✓	✓	✗	**não**	**sim**
Taxonomia de Thorat [118]	2021		✗	–	**não**	✓	✓	✓	✗	**não**	✓	✓	✗	**não**	**sim**
Taxonomia de Aleroud [119]	2017	Ataques de phishing	✗	✓	**não**	✓	✓	✓	✓	**sim**	✓	✓	✓	**sim**	**não**
Taxonomia de Gupta [121]	2018		✗	✓	**não**	✓	✓	✓	✓	**sim**	✓	✓	✓	**sim**	**sim**
Taxonomia de Rastenis [124]	2020		✓	✓	**sim**	✓	✓	✓	✓	**sim**	✓	✓	✓	**sim**	**não**
CATRA [96]	2015	Sistemas em nuvem	✗	✓	**não**	✓	✓	✓	✓	**sim**	✗	✗	✓	**não**	**não**
Taxonomia de Juliadotter [106]	2015		✓	✓	**sim**	✓	✓	✓	✓	**sim**	✗	✗	✓	**não**	**não**
Taxonomia de Iqbal [125]	2016		✗	✓	**não**	✓	✓	✓	✓	**sim**	✓	✓	✓	**sim**	**sim**
Taxonomia de Mishra [87]	2017		✓	✓	**sim**	✓	✓	✓	✓	**sim**	✓	✓	✓	**sim**	**sim**
Taxonomia de Akshaya [128]	2019		✗	✓	**não**	✗	✗	✓	✗	**não**	✗	✓	✗	**não**	**não**
Taxonomia de Singh [134]	2019	Aplicações web	✗	✓	**não**	✓	✓	✓	✓	**sim**	✓	✓	✓	**sim**	**não**
Taxonomia de Koutrouli [136]	2012	Aplicações P2P	✗	✓	**não**	✓	✓	✓	✓	**sim**	✓	✓	✓	**sim**	**sim**
Taxonomia de Hollick [137]	2017	Infraestrutura da Internet	✓	–	**sim**	✓	✓	✓	✓	**sim**	✓	✓	✓	**sim**	**sim**

O campo da segurança cibernética é amplo em escopo e intrinsecamente complexo, abrangendo múltiplos contextos e perspectivas, conforme discutido no início deste capítulo. Essa diversidade impõe desafios significativos à construção de taxonomias verda-

deiramente abrangentes. A principal limitação observada é a dificuldade em satisfazer simultaneamente todos os requisitos definidos na Seção 2.1, conforme evidenciado na Tabela 2.7. As taxonomias voltadas para redes e computadores — que representam as abordagens mais gerais — não conseguiram atender plenamente a todos os requisitos. Nesse conjunto (primeiras quatro linhas da Tabela 2.7), as propriedades '*exaustiva*', '*terminologia em conformidade*' e '*similaridade*' foram as mais frequentemente reprovadas.

De acordo com a Tabela 2.7, apenas 3 das 20 taxonomias satisfazem todos os requisitos: as taxonomias de Masdari [113], Mishra [87] e Hollick [137]. Além disso, 7 taxonomias atendem a 3 requisitos, 4 a 2 requisitos, 2 a 1 requisito e 4 não atendem a nenhum deles. Assim, observa-se que 50% das taxonomias analisadas cumprem pelo menos 3 dos 4 requisitos avaliados. Em uma análise individual, 45% das taxonomias são *completas*, 60% são *compreensíveis*, 60% são *repetíveis* e 50% são *mutuamente exclusivas*. Esses resultados indicam que a completude é o requisito mais difícil de alcançar, o que pode ser atribuído à grande variedade de ataques existentes e ao surgimento contínuo de novas ameaças. O segundo requisito menos atendido é a exclusividade mútua, embora, em alguns casos, essa característica seja deliberadamente flexibilizada pelos próprios autores da taxonomia.

A Tabela 2.8 apresenta a comparação das taxonomias com base nos critérios de avaliação estrutural. Nessa análise, os critérios *Esquema*, *Organização*, *Rotulagem* e *Abordagem* possuem natureza qualitativa, enquanto $\#D$, $\#L$ e $\#C$ representam, respectivamente, o número de dimensões, níveis e classes da taxonomia. O valor de $\#C$ é exato apenas em taxonomias *mutuamente exclusivas* com categorias *primitivas*; nos demais casos, ele deve ser interpretado como uma estimativa comparativa, pois o número efetivo pode variar.

A avaliação estrutural (Tabela 2.8) revela que a maioria das taxonomias analisadas é *plana* (55%), *hierárquica* (95%), *primitiva* (80%) e *orientada por característica* (70%). Embora a proporção entre os tipos de esquema apresente relativa homogeneidade, observa-se que as taxonomias de escopo mais amplo tendem a adotar esquemas *multidimensionais*, enquanto as mais específicas optam por esquemas *planos*. Entre as taxonomias multidimensionais, o número médio de dimensões é 7. Quanto à organização, a média é de três níveis, independentemente do esquema adotado. Por fim, nas taxonomias planas, o número de classes varia de 8 a 47, com média de 17.

2.4.9 Questões em aberto, desafios e possíveis direções futuras

As taxonomias de ataques desempenham um papel fundamental na segurança cibernética, ao fornecer uma estrutura sistemática para categorizar e compreender diferentes tipos de ameaças. Ao organizar os ataques em categorias distintas, essas taxonomias permitem

Tabela 2.8: Comparação de taxonomias de ataque pela avaliação estrutural

Taxonomia	Ano	Contexto	Esquema	#D	Organização	#L	Rotulagem	#C	Abordagem
Taxonomia de Chapman [104]	2011	Rede e Computadores	plana	1	hierárquica	2	primitiva	13	OC
AVOIDIT [1]	2014		multi.	5	hierárquica	5	derivada	56320	OP
ADMIT [105]	2014		multi.	5	hierárquica	2	derivada	13552	OP
Taxonomia de Hindy [108]	2020		multi.	3	hierárquica	4	primitiva	85	OC
Taxonomia de Bhardwaj [112]	2016	Ataques DoS/DDoS	multi.	4	hierárquica	2	primitiva	192	OC
Taxonomia de Masdari [113]	2016		plana	1	hierárquica	4	primitiva	47	OC
Taxonomia de De Donno [114]	2017		multi.	14	hierárquica	3	primitiva	152000	OC
Taxonomia de Sharafaldin [115]	2019		plana	1	hierárquica	3	primitiva	13	OC
Taxonomia de Thorat [118]	2021		plana	1	hierárquica	3	primitiva	11	OC
Taxonomia de Aleroud [119]	2017	Ataques de phishing	multi.	4	hierárquica	3	primitiva	11400	OP
Taxonomia de Gupta [121]	2018		plana	1	hierárquica	3	primitiva	8	OC
Taxonomia de Rastenis [124]	2020		multi.	12	hierárquica	2	primitiva	1512000	OP
CATRA [96]	2015	Sistemas em nuvem	multi.	13	hierárquica	3	derivada	13,9 bi	OP
Taxonomia de Juliadotter [106]	2015		multi.	5	linear	1	derivada	864	OP
Taxonomia de Iqbal [125]	2016		plana	1	hierárquica	2	primitiva	15	OC
Taxonomia de Mishra [87]	2017		plana	1	hierárquica	2	primitiva	22	OC
Taxonomia de Akshaya [128]	2019		plana	1	hierárquica	2	primitiva	17	OC
Taxonomia de Singh [134]	2019	Aplicações web	plana	1	hierárquica	3	primitiva	14	OC
Taxonomia de Koutrouli [136]	2012	Aplicações P2P	plana	1	hierárquica	4	primitiva	16	OC
Taxonomia de Hollick [137]	2017	Infraestrutura da Internet	plana	1	hierárquica	2	primitiva	12	OC

OC: orientada por característica; OP: orientada por processo.

que profissionais e pesquisadores analisem, identifiquem padrões e respondam às ameaças de maneira mais eficaz.

Apesar de sua relevância conceitual e prática, as taxonomias de ataques ainda enfrentam diversas limitações e lacunas, tanto no que diz respeito à sua padronização quanto à sua capacidade de acompanhar a rápida evolução do cenário de ameaças. Nesse contexto, esta subseção discute questões em aberto, desafios persistentes e possíveis direções futuras para o desenvolvimento de taxonomias mais abrangentes, dinâmicas e interoperáveis.

Questões em aberto

1. *Como as taxonomias de ataques podem ser padronizadas e harmonizadas entre organizações e setores?* Atualmente, observa-se uma falta de consistência na forma como as taxonomias de ataques são definidas e utilizadas, o que dificulta a comparação e o compartilhamento de informações entre entidades. O desenvolvimento de uma taxonomia padronizada e amplamente aceita poderia aprimorar a colaboração e o intercâmbio de conhecimento na comunidade de segurança cibernética.

2. *Como as taxonomias de ataques podem ser adaptadas para acompanhar o cenário de ameaças em constante evolução?* As ameaças cibernéticas evoluem continuamente, à medida que adversários adotam novas técnicas e estratégias. Assim, as taxonomias de ataques devem ser flexíveis e versáteis, de modo a incorporar vetores e táticas emergentes. É necessário promover pesquisas que identifiquem novos pa-

drões de ataque e viabilizem sua integração nas taxonomias existentes, mantendo-as atualizadas e relevantes.

3. *Como as taxonomias de ataques podem se tornar mais abrangentes e inclusivas?* As taxonomias atualmente disponíveis nem sempre cobrem todos os cenários de ataque possíveis e, em muitos casos, concentram-se em domínios ou tecnologias específicas. Ampliar a abrangência e a inclusividade dessas taxonomias permitiria uma compreensão mais holística do ecossistema de ameaças. Essa ampliação deve contemplar, por exemplo, ataques direcionados a tecnologias emergentes, como Internet das Coisas (IoT), computação em nuvem e inteligência artificial (IA).

Desafios

1. *Falta de consenso:* Não existe, atualmente, uma taxonomia de ataques amplamente aceita pela comunidade de segurança cibernética. Embora existam iniciativas abertas, como a CAPEC[12] [138] e a WASC Threat Classification[13], diversas organizações desenvolvem e mantêm taxonomias próprias, buscando atender a necessidades específicas ou preservar controle e flexibilidade. O desenvolvimento de uma taxonomia baseada em consenso exigiria colaboração interinstitucional e alinhamento entre múltiplas partes interessadas.
2. *Evolução acelerada das técnicas de ataque:* Os adversários estão em constante inovação, desenvolvendo novas técnicas e estratégias para contornar mecanismos de defesa. Esse dinamismo impõe desafios às taxonomias, que precisam acompanhar tais avanços para permanecerem eficazes. Quanto mais abrangente for uma taxonomia, mais complexa será sua manutenção. Assim, são necessárias atualizações e pesquisas contínuas para assegurar que as taxonomias reflitam os vetores de ataque mais recentes.
3. *Equilíbrio entre granularidade e usabilidade:* O desenvolvimento de taxonomias de ataque enfrenta o desafio de equilibrar a riqueza de detalhes (granularidade) com a facilidade de uso em contextos práticos. Esse dilema é particularmente evidente em taxonomias de escopo amplo, nas quais a necessidade de representar uma grande diversidade de ameaças pode comprometer a simplicidade e a aplicabilidade do modelo. A granularidade excessiva — associada a valores elevados de $\#D$, $\#L$ ou $\#C$ — tende a gerar estruturas complexas e de difícil aplicação. Em contrapartida, uma simplificação excessiva pode resultar na omissão de vetores de ataque relevantes, reduzindo a eficácia e a utilidade da taxonomia.

[12]Disponível em `https://capec.mitre.org`
[13]Disponível em `http://projects.webappsec.org/Threat-Classification`

4. *Amplitude do domínio:* A vastidão e a complexidade inerentes ao campo da segurança cibernética impõem desafios significativos ao desenvolvimento de taxonomias abrangentes. À medida que o escopo de uma taxonomia se amplia para abranger diferentes contextos, torna-se progressivamente mais difícil satisfazer simultaneamente todos os requisitos — completa, compreensível e repetível. O aumento do número de tipos de ataque, da terminologia envolvida e das possíveis ambiguidades intensifica a complexidade do processo de modelagem e validação.

Possíveis direções futuras

1. *Aprendizado de máquina e abordagens baseadas em dados:* O uso de técnicas de aprendizado de máquina (ML) pode contribuir para a identificação e categorização automática de novos padrões de ataque. A análise de grandes datasets de incidentes cibernéticos possibilita o desenvolvimento de taxonomias de ataque mais precisas e atualizadas.
2. *Esforços de colaboração e padronização:* A promoção da colaboração entre pesquisadores, organizações e setores pode conduzir ao desenvolvimento de taxonomias de ataque padronizadas e interoperáveis. Essa padronização facilitaria o compartilhamento de informações, a comparação de resultados e a formulação de boas práticas em segurança cibernética.
3. *Integração com inteligência de ameaças:* A incorporação de *feeds* de inteligência de ameaças às taxonomias de ataque pode aumentar sua relevância e eficácia. Ao integrar informações em tempo real sobre ameaças emergentes, as taxonomias podem ser atualizadas de forma contínua, refletindo com maior precisão o cenário dinâmico das ameaças cibernéticas.

Capítulo 3

Inteligência de Ameaça Cibernética

O conceito de inteligência é tão antigo quanto a própria guerra. A história da humanidade está repleta de relatos de espionagem e coleta de informações estratégicas, motivadas pela necessidade de obter vantagem sobre o inimigo. "Por *inteligência* entendemos ser todo o tipo de informação sobre o inimigo e seu país — a base, em suma, dos nossos próprios planos e operações", afirmou o general prussiano Carl von Clausewitz em seu livro *On War* de 1832 [139]. Essa necessidade militar, especialmente durante as duas guerras mundiais, impulsionou o crescimento e a evolução do campo da inteligência como o conhecemos atualmente [3].

A inteligência distingue-se da informação por duas características essenciais:

- A capacidade de antecipar ou prever situações e circunstâncias futuras; e
- O papel de apoiar a tomada de decisões, esclarecendo as diferenças entre os possíveis cursos de ação (Course of Action – COA) [140].

Os dados brutos, por si só, possuem utilidade limitada. Quando processados em uma forma inteligível, tornam-se informações. Contudo, somente quando essas informações são correlacionadas com o ambiente operacional e interpretadas à luz da experiência, elas geram um novo entendimento — a *inteligência* [140]. O processo de produção de inteligência, representado na Figura 3.1, é tradicionalmente composto por três etapas principais: coleta, processamento e análise.

A coleta de dados é realizada a partir de múltiplas fontes do ambiente operacional, podendo envolver meios técnicos ou humanos. Esses dados são então processados e transformados em informações, que, após análise e contextualização, originam a inteligência propriamente dita.

O conceito de inteligência aplicado à cibersegurança ganhou relevância no final da década de 1990 e início dos anos 2000 [141]. Desde então, diversas entidades de segurança cibernética têm proposto definições próprias, refletindo diferentes perspectivas procedimentais e, em alguns casos, motivações competitivas [142].

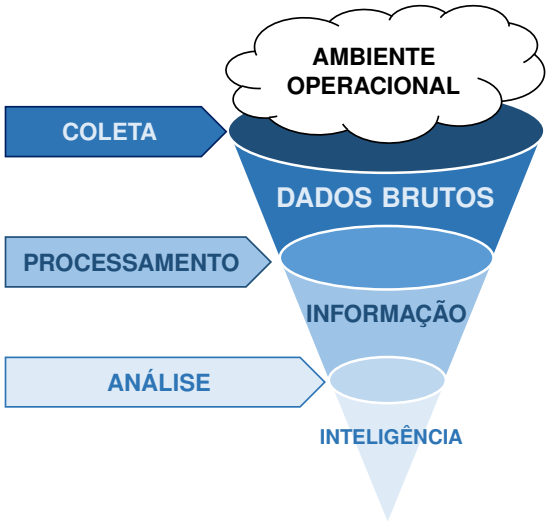


Figura 3.1: Processo de produção de inteligência

A Inteligência de Ameaça Cibernética (Cyber Threat Intelligence – CTI) pode ser compreendida, de forma simplificada, como um conjunto de informações detalhadas e acionáveis sobre ameaças de cibersegurança [143]. Uma definição mais abrangente descreve a CTI como "o conhecimento baseado em evidências — incluindo contexto, mecanismos, indicadores, implicações e recomendações acionáveis — sobre uma ameaça existente ou emergente para os ativos, que pode ser utilizado para embasar decisões relacionadas à resposta a essa ameaça" [144, 142].

O termo "*acionável*" indica que a inteligência deve fornecer informações úteis, com contexto suficiente para permitir que os destinatários tomem decisões e desenvolvam respostas adequadas [145, 12]. Assim, para ser considerada acionável, a CTI deve ser oportuna, precisa e relevante [11].

No cenário dinâmico da segurança cibernética, a CTI desempenha papel fundamental na manutenção da postura defensiva das organizações. Ela auxilia os defensores a compreender os atacantes, responder com maior rapidez a incidentes e antecipar ameaças futuras [10].

3.1 Classificação e escopo da CTI

O escopo da CTI é amplo e abrange múltiplas fontes e finalidades de uso. Por essa razão, a literatura classifica a CTI em subdomínios, de acordo com o público-alvo, os objetivos e o horizonte temporal de aplicação (curto ou longo prazo). Entre os diversos modelos existentes, este trabalho adota a classificação mais recente na literatura, apresentada na Figura 3.2, que divide a CTI em quatro tipos: estratégica, tática, operacional e técnica [14, 16, 146].

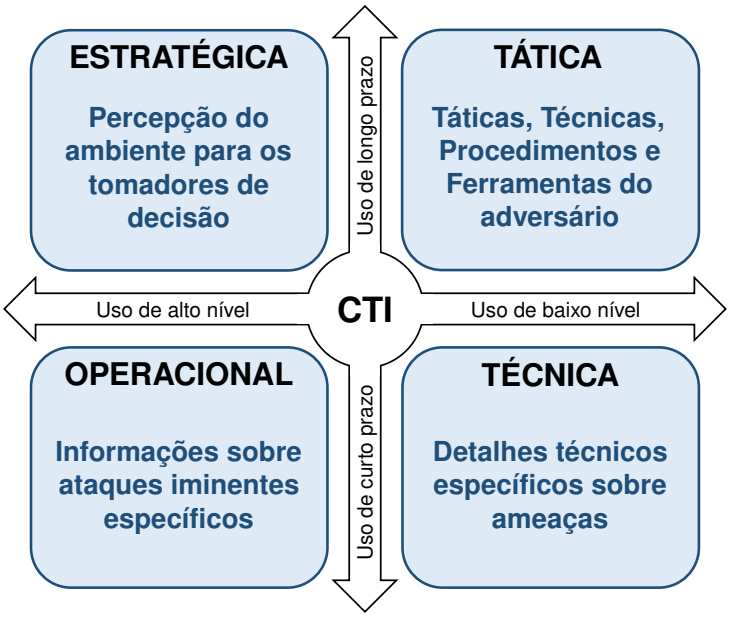


Figura 3.2: Tipos de inteligência de ameaça

CTI Estratégica: Fornece informações de alto nível sobre o panorama de ameaças, como tendências de ataque, impactos econômicos e implicações geopolíticas. É voltada a executivos e tomadores de decisão, auxiliando na definição de políticas e investimentos em segurança.

CTI Tática: Descreve como os agentes de ameaça conduzem suas operações, incluindo ferramentas, técnicas e metodologias empregadas — isto é, suas TTPs. É consumida por analistas de defesa e equipes de resposta a incidentes, permitindo ajustar defesas, alertas e investigações de acordo com as táticas adversárias em uso.

CTI Operacional: Foca em ataques específicos ou iminentes contra a organização. É voltada a equipes de segurança de nível superior, como gerentes de segurança e líderes de resposta a incidentes. Informações dessa natureza — sobre quem, quando e como atacará — são raramente acessíveis a entidades privadas, sendo mais comuns em agências governamentais com acesso privilegiado a fontes sigilosas.

CTI Técnica: Contém detalhes técnicos diretamente observáveis sobre ameaças, como dados de infraestrutura, ferramentas, canais de C2 e outros indicadores técnicos. Normalmente é compartilhada na forma de IoCs, como *hashes* de arquivos maliciosos, endereços IP comprometidos ou domínios suspeitos.

3.2 Limitações da CTI Técnica e relevância da CTI Tática

A CTI Técnica é o tipo mais amplamente disseminado por fornecedores de inteligência [15]. Isso ocorre porque os Indicadores de Comprometimento (IoCs) são diretamente acionáveis e fáceis de integrar a sistemas de segurança. Geralmente, são consumidos via *feeds* automatizados, em formatos padronizados e legíveis por máquina (*machine-readable*), permitindo sua rápida incorporação em mecanismos de detecção e bloqueio.

No entanto, essa vantagem operacional traz limitações significativas. Como os indicadores são amplamente distribuídos em *feeds* públicos, os adversários também têm acesso a eles e podem alterar rapidamente seus artefatos e sua infraestrutura de ataque — modificando *hashes*, endereços IP ou domínios — para evitar detecção. Por isso, a CTI Técnica possui vida útil curta e deve ser consumida de forma imediata. Além disso, *feeds* imprecisos podem gerar falsos positivos e até bloquear tráfego legítimo [14, 147].

Outra limitação é o baixo grau de correlação entre os IoCs, que representam artefatos isolados e não descrevem o comportamento do atacante. Como resultado, as detecções baseadas exclusivamente em indicadores técnicos tendem a ser fragmentadas e pouco eficazes na identificação da cadeia completa de ataque [148].

Por essa razão, diversas pesquisas defendem a transição do foco em artefatos para o foco em comportamentos dos adversários [149]. Operar no nível da CTI Tática, buscando identificar as táticas, técnicas e procedimentos (TTPs) utilizados pelo adversário, permite compreender e detectar o *modus operandi* da ameaça de forma mais robusta e duradoura do que com indicadores técnicos.

3.3 A Pirâmide da Dor

A relação hierárquica entre os diferentes tipos de indicadores é representada pela Pirâmide da Dor (Pyramid of Pain – PoP), ilustrada na Figura 3.3. O modelo expressa o grau de dificuldade imposto ao adversário quando determinado tipo de indicador é identificado e neutralizado pelos mecanismos defensivos [2].

Os níveis inferiores da pirâmide representam os indicadores técnicos — como *hashes*, endereços IP, domínios e artefatos de rede ou *host* — que são facilmente modificáveis pelos atacantes. Nos níveis superiores encontram-se os indicadores táticos, como ferramentas e TTPs. Estes são mais estáveis e difíceis de alterar, pois refletem o comportamento e o conhecimento operacional do adversário, fortemente condicionado às limitações tecnológicas do alvo [149].

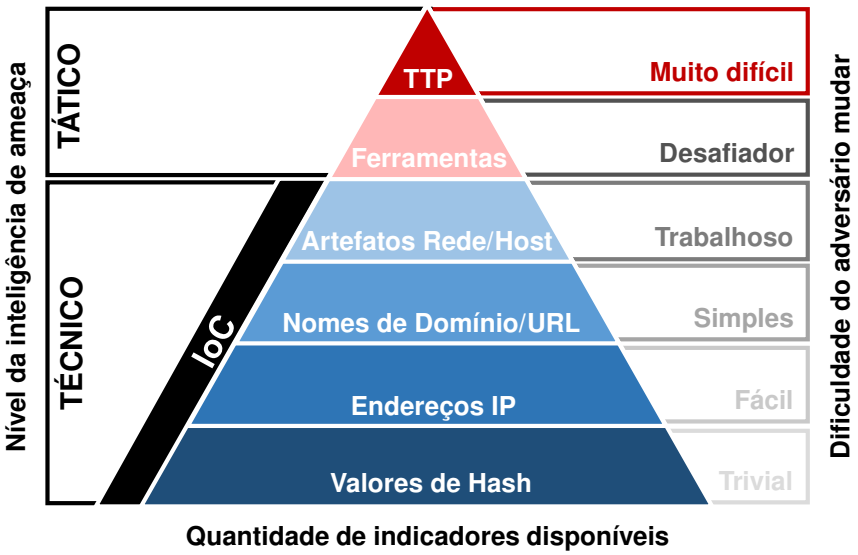


Figura 3.3: Pirâmide da Dor (PoP) [2]

Assim, identificar e responder de forma oportuna às TTPs de um adversário o força, para evitar a detecção, a realizar a mudança mais onerosa possível: aprender novas técnicas e alterar seu *modus operandi*. Essa característica torna a detecção baseada em TTP um componente essencial da defesa cibernética moderna.

Capítulo 4

Threat Hunting

Em termos quantitativos, a maioria das ameaças à segurança da informação é conhecida e pode ser contida por mecanismos tradicionais de defesa, como antivírus e *firewalls* [150]. No entanto, existe uma minoria de ameaças pouco conhecidas ou até desconhecidas que representam riscos significativamente maiores, pois são difíceis de detectar e de neutralizar. As consequências dessas ameaças tendem a ser mais severas, uma vez que elas operam fora da cobertura dos mecanismos de defesa convencionais.

No domínio da segurança cibernética, há um princípio amplamente aceito: "não existe 100% de segurança". Em outras palavras, nenhuma organização é capaz de impedir todos os ataques o tempo todo [151]. Isso implica que, cedo ou tarde, adversários conseguirão violar alguma camada de defesa. Reconhecendo essa inevitabilidade, o foco da segurança cibernética foi estendido para não apenas impedir o ataque, mas também reduzir e controlar o impacto de uma violação quando ela ocorre.

Essa mudança de paradigma levou ao desenvolvimento de mecanismos proativos voltadas à detecção e neutralização de ameaças que conseguiram transpor as barreiras de proteção existentes — o *Threat Hunting*.

O Threat Hunting pode ser definido como "o processo proativo e iterativo de busca em redes para detectar e isolar ameaças avançadas que evadem as soluções de segurança existentes" [152]. Seu objetivo principal é reduzir o tempo necessário para identificar invasores que já comprometeram o ambiente de TI, minimizando o impacto da violação. Atualmente, essa atividade depende fortemente da capacidade analítica humana para identificar indícios sutis de ameaça que passaram despercebidos pelos mecanismos automatizados. Ao identificar precocemente a atuação adversária, o impacto operacional e financeiro pode ser substancialmente reduzido [150].

Em síntese, o papel do caçador de ameaças é diminuir o chamado '*dwell time*' — o intervalo entre o momento em que o adversário obtém acesso ao ambiente e o momento em que a violação é detectada [3]. Essa dinâmica é ilustrada na Figura 4.1.

Figura 4.1: Linha do tempo do Threat Hunting [3]

Sempre haverá uma lacuna entre a capacidade de detecção de uma organização e a habilidade de um adversário sofisticado em evitá-la. Os recursos disponíveis para o atacante e para o defensor variam amplamente, o que torna inevitável a existência de brechas de detecção. Uma abordagem eficaz de Threat Hunting busca justamente reduzir essa lacuna, rastreando o comportamento adversário e aprimorando continuamente os mecanismos de detecção [150].

4.1 Abordagens de detecção de Threat Hunting

A atividade de Threat Hunting engloba múltiplas abordagens de detecção, que podem ser implementadas por diferentes soluções técnicas. As principais abordagens são descritas a seguir:

Detecção por Indicadores de Comprometimento (IoC): Foca na busca de indicadores técnicos específicos — como endereços IP, *hashes* de arquivos, URLs suspeitas ou padrões de tráfego — associados a ameaças conhecidas. Embora de fácil implementação, essa abordagem exige atualização constante das assinaturas e é afetada pela fragilidade e volatilidade dos IoCs, o que pode resultar em altas taxas de falsos positivos.

Detecção baseada em anomalias: Utiliza análise estatística, ML e outras técnicas de análise de grandes volumes de dados para identificar comportamentos atípicos. Essa abordagem exige investimentos expressivos em coleta e processamento de dados, e frequentemente carece de contexto suficiente para justificar por que uma atividade foi classificada como suspeita. Além disso, definir o comportamento "normal" de uma rede corporativa é um desafio, dada a variabilidade natural das atividades legítimas ao longo do tempo.

Detecção baseada em TTPs: Em vez de procurar ferramentas ou artefatos específicos, essa abordagem busca identificar as técnicas que os adversários utilizam para atingir seus objetivos. As TTPs são menos suscetíveis a mudanças e mais representativas do comportamento adversário, uma vez que estão condicionadas às restrições tecnológicas do alvo. Modelos orientados ao comportamento, como o MITRE ATT&CK

[18], permitem que os defensores estruturem hipóteses de detecção baseadas nas ações e fases operacionais do adversário.

4.2 Classificação das soluções técnicas

Quanto à solução técnica empregada, o Threat Hunting pode ser classificado em quatro categorias principais [153]:

Baseada em estatística: Fundamenta-se em métodos estatísticos para detectar desvios em relação a padrões de comportamento normais. Ao analisar séries históricas e estabelecer *baselines*, essa abordagem identifica entidades ou atividades potencialmente maliciosas. É particularmente útil para detectar ataques que se disfarçam em volumes legítimos de tráfego. Entretanto, é altamente suscetível a falsos positivos, especialmente em ambientes dinâmicos. Além disso, requer grande investimento em coleta, representação e processamento de dados em larga escala [153].

Baseada em ML/IA: Utiliza algoritmos de aprendizado de máquina e técnicas de IA para identificar padrões anômalos e comportamentos maliciosos. O aprendizado pode ser supervisionado — quando o modelo é treinado com dados rotulados — ou não supervisionado — quando identifica padrões ocultos sem orientação prévia. Apesar do potencial para detectar ameaças emergentes e desconhecidas, essas técnicas enfrentam desafios consideráveis, como a necessidade de grandes volumes de dados rotulados para treinamento, a complexidade dos modelos e a dificuldade de interpretação dos resultados.

Baseada em grafos: Representa a sequência e a semântica dos eventos por meio de grafos, permitindo analisar relações complexas entre entidades. Essa abordagem é eficaz para correlacionar eventos dispersos, compreender comportamentos adversários e identificar causas raiz de incidentes [154, 153]. Contudo, a construção e manutenção dos grafos são operações computacionalmente custosas, e a análise dos resultados requer conhecimento especializado.

Baseada em regras: Fundamenta-se em regras pré-definidas para identificar comportamentos suspeitos ou artefatos maliciosos em dados coletados. Tais regras são derivadas de conhecimento especializado e de IoCs. Embora eficaz na detecção de ameaças conhecidas, essa abordagem é limitada contra ataques novos e sofisticados. A necessidade de manutenção contínua das regras e a ausência de metodologias estruturadas para sua criação agravam esse desafio.

A metodologia proposta nesta pesquisa busca operacionalizar a abordagem de detecção baseada em TTPs por meio da criação de Runbooks acionáveis que contenham regras de detecção estruturadas. Esses Runbooks não apenas consolidam as regras baseadas em TTPs, mas também integram diversas informações táticas relevantes, servindo como instrumentos de inteligência acionável. Dessa forma, eles fornecem aos analistas os meios necessários para implementar *workflows* de detecção em suas plataformas de Threat Hunting, aumentando a precisão, a eficiência e a capacidade de resposta das operações defensivas.

Capítulo 5

MITRE ATT&CK

O ATT&CK, acrônimo de *Adversarial Tactics, Techniques, and Common Knowledge*, é um *framework* desenvolvido pela MITRE Corporation[1] para descrever as táticas e técnicas empregadas por adversários cibernéticos durante a execução de um ataque [18]. Esse modelo fornece uma taxonomia comum para a comunidade de segurança cibernética, sendo amplamente utilizado para estudar, classificar e rotular as atividades que um agente de ameaça é capaz de realizar ao se estabelecer e operar em um ambiente cibernético [3].

Com o objetivo de padronizar a descrição e a categorização das ações adversárias, o framework ATT&CK define os seguintes conceitos [155]:

Táticas (*Tactics*): Representam as fases ou objetivos de alto nível de um ataque cibernético. Cada tática descreve o propósito que o adversário busca atingir em determinado estágio da operação, sendo identificada por um código no formato `TAxxxx`. Exemplo: 'TA0006 – Credential Access'.

Técnicas (*Techniques*): Descrevem as ações específicas empregadas pelos adversários para atingir os objetivos definidos em cada tática. Cada técnica possui um identificador exclusivo no formato `Txxxx`. Exemplo: 'T1003 – OS Credential Dumping'.

Subtécnicas (*Subtechniques*): Detalham variações de uma técnica principal, permitindo granularidade adicional na descrição das ações adversárias. São identificadas acrescentando-se um número no formato `.xxx` ao identificador da técnica. Exemplo: 'T1003.001 – OS Credential Dumping: LSASS Memory'.

Procedimentos (*Procedures*): Representam a implementação prática das técnicas ou subtécnicas, descrevendo como os adversários executam as ações em um ambiente real. Os procedimentos constituem o nível mais detalhado da hierarquia, especificando comandos, ferramentas e comportamentos observáveis. Embora o framework

[1] `https://www.mitre.org`

ATT&CK não atribua identificadores formais aos procedimentos, ele fornece exemplos extraídos de ameaças conhecidas.

No contexto da inteligência de ameaça, as Táticas, Técnicas e Procedimentos (TTPs) formam uma estrutura hierárquica que descreve o *modus operandi* dos adversários. As táticas respondem ao "porquê", ou seja, o objetivo pretendido em cada etapa do ataque; as técnicas indicam "como" esse objetivo pode ser alcançado; e os procedimentos mostram formas de implementação das técnicas, descrevendo "o que" exatamente o atacante faz para para atingir o objetivo pretendido, incluindo os meios específicos utilizados [156, 157].

Essa representação fornece uma visão comportamental da ameaça, permitindo compreender não apenas o que foi feito, mas também como e por que foi feito. Essa perspectiva é essencial para a detecção baseada em comportamento, o desenvolvimento de defesas proativas e a predição de ações adversárias com base em padrões conhecidos.

5.1 Fontes e componentes de Dados

O framework ATT&CK também define dois conceitos complementares — *Data Sources* e *Data Components* — que são fundamentais para o desenvolvimento de mecanismos de detecção.

Fontes de Dados (*Data Sources*): São categorias de dados de segurança de alto nível (tópicos ou sujeitos) coletados de sistemas, aplicações e redes que podem ser utilizadas para detectar atividades adversárias. Elas representam o "onde" os dados são gerados (*e.g.* processo, registro do Windows, tráfego de rede, arquivo). Cada *Data Source* possui um identificador exclusivo no formato `DSxxxx` [158]. Exemplo: `DS0024 - Windows Registry`.

Componentes de Dados (*Data Components*): Representam um nível de granularidade mais aprofundado que define propriedades ou valores específicos dentro de um *Data Source* que são relevantes para a detecção de uma técnica. Eles definem "o que" monitorar nos *logs* (*e.g.* criação de processo, deleção de arquivo, modificação de chave de registro). Os *Data Components* recebem identificadores exclusivos no formato `DCxxxx` [159]. Exemplo: `DC0056 - Windows Registry Key Creation`.

A Tabela 5.1 apresenta exemplos de *Data Sources* e *Data Components* definidos no framework ATT&CK [158, 159].

5.2 Sub-frameworks do MITRE ATT&CK

O MITRE ATT&CK é atualmente dividido em três *sub-frameworks*, desenvolvidos para cobrir diferentes contextos de ameaças:

Tabela 5.1: Exemplos de *Data Sources* e *Data Components* no framework ATT&CK

DS ID	Data Source	Data Component	DC ID
DS0002	User Account	User Account Authentication	DC0002
		User Account Creation	DC0014
		User Account Deletion	DC0009
		User Account Metadata	DC0013
		User Account Modification	DC0010
DS0022	File	File Access	DC0055
		File Creation	DC0039
		File Deletion	DC0040
		File Metadata	DC0059
		File Modification	DC0061
DS0009	Process	OS API Execution	DC0021
		Process Access	DC0035
		Process Creation	DC0032
		Process Metadata	DC0034
		Process Modification	DC0020
		Process Termination	DC0033
DS0024	Windows Registry	Windows Registry Key Access	DC0050
		Windows Registry Key Creation	DC0056
		Windows Registry Key Deletion	DC0045
		Windows Registry Key Modification	DC0063

Enterprise: *Sub-framework* principal, voltado a ataques que visam redes corporativas, sistemas operacionais e aplicativos de uso geral

Mobile: Focado em ameaças que exploram dispositivos móveis, como *smartphones* e *tablets*, abrangendo tanto os sistemas operacionais quanto as comunicações móveis.

ICS: Direcionado a ambientes de controle industrial, abordando técnicas e ameaças específicas de sistemas de automação e infraestrutura crítica.

Cada *sub-framework* possui uma matriz bidimensional que representa, de forma visual, a relação entre táticas e técnicas utilizadas por adversários durante suas operações. Essas matrizes são dinâmicas e estão em constante evolução, sendo atualizadas com a inclusão de novas técnicas e táticas.

Atualmente, a matriz do *sub-framework Enterprise* contém 14 táticas que descrevem as fases e os objetivos de um ataque em ambiente corporativo:

1. *Reconnaissance* (Reconhecimento) - TA0042: Engloba técnicas para coletar informações sobre o ambiente alvo, como identificação de ativos, mapeamento de rede, identificação de serviços em execução e descoberta de vulnerabilidades. Essa tática

visa obter um conhecimento mais profundo do ambiente antes de prosseguir com o ataque.

2. *Resource Development* (Desenvolvimento de Recursos) - TA0043: Engloba técnicas para identificar e adquirir recursos adicionais, como contas de usuário adicionais, credenciais, ferramentas ou acesso a serviços externos. Essa tática busca expandir as capacidades do adversário e garantir recursos para avançar em outras fases do ataque.
3. *Initial Access* (Acesso Inicial) - TA0001: Engloba técnicas para obter acesso inicial a um sistema ou rede.
4. *Execution* (Execução) - TA0002: Engloba técnicas para executar código malicioso em um sistema-alvo.
5. *Persistence* (Persistência) - TA0003: Engloba técnicas para manter acesso persistente após reinicialização ou interrupção do sistema.
6. *Privilege Escalation* (Escalação de Privilégios) - TA0004: Engloba técnicas para obter privilégios elevados em um sistema ou rede.
7. *Defense Evasion* (Defesa e Evasão) - TA0005: Engloba técnicas para evitar detecção por ferramentas de segurança e contornar mecanismos de defesa.
8. *Credential Access* (Acesso a Credenciais) - TA0006: Engloba técnicas para obter credenciais válidas para acesso não autorizado.
9. *Discovery* (Descoberta) - TA0007: Engloba técnicas para identificar informações sobre sistemas, redes e ambientes alvo.
10. *Lateral Movement* (Movimento Lateral) - TA0008: Engloba técnicas para se mover lateralmente dentro de uma rede comprometida.
11. *Collection* (Coleta de Informações) - TA0009: Engloba técnicas para coletar dados e informações de interesse.
12. *Command and Control* (Comando e Controle) - TA0011: Engloba técnicas para estabelecer e manter comunicações entre os adversários e sistemas comprometidos.
13. *Exfiltration* (Exfiltração) - TA0010: Engloba técnicas para extrair dados roubados do ambiente comprometido.
14. *Impact* (Impacto) - TA0040: Engloba técnicas para interromper ou causar danos aos sistemas ou à infraestrutura alvo.

As duas primeiras táticas — TA0042 e TA0043 — foram incorporadas posteriormente, quando a matriz *Pre-ATT&CK* foi fundida à matriz *Enterprise*. Essas táticas descrevem

etapas de preparação e desenvolvimento de recursos antes da execução do ataque propriamente dito.

5.3 Relevância na pesquisa

O framework ATT&CK [18] representa um marco fundamental na segurança cibernética, fornecendo uma linguagem comum para descrever e compartilhar conhecimento sobre o comportamento adversário. Ele permite não apenas identificar Táticas, Técnicas e Subtécnicas, mas também estabelecer correlações entre ações adversárias e evidências observáveis em dados operacionais. Contudo, o framework apresenta limitações que desafiam sua aplicação prática.

A classificação de táticas e técnicas envolve certo grau de subjetividade e depende da disponibilidade de exemplos empíricos, o que pode gerar inconsistências no mapeamento de TTPs. Além disso, a cobertura de procedimentos ainda é restrita, e desafios relacionados à interoperabilidade e à automatização limitam a aplicação direta do framework em ambientes operacionais [160].

Nesta pesquisa, o MITRE ATT&CK desempenha papel central como base taxonômica para a classificação das Táticas e Técnicas adversárias ao longo da metodologia proposta. Adicionalmente, suas estruturas de *Data Sources* e *Data Components* são utilizadas como referência para o mapeamento de dados e o desenvolvimento das Regras de Detecção, consolidando o framework como uma das principais fontes de inteligência tática de ameaça na construção dos *Runbooks*.

Capítulo 6

Trabalhos Relacionados

A inteligência de ameaça cibernética (Cyber Threat Intelligence – CTI) evoluiu de uma prática emergente para uma disciplina essencial na defesa cibernética, amplamente estudada e incorporada em soluções de segurança. Um marco importante nessa evolução foi a publicação do relatório *APT1 – Exposing One of China's Cyber Espionage Units* [161] pela Mandiant, em 2013, que destacou a importância de compreender o elemento humano por trás dos ataques, e não apenas o *malware* em si [162].

Outro avanço relevante foi a introdução em 2013 do conceito de Táticas, Técnicas e Procedimentos (TTPs) na Pyramid of Pain (PoP) [2], Figura 3.3, que enfatizou a importância da detecção nesse nível. Sob essa perspectiva, identificar TTPs é mais eficaz, pois obriga o adversário a modificar seu comportamento para não ser detectado, dificultando as operações de ataque. Em 2015, a MITRE lançou o framework ATT&CK, consolidando o uso das TTPs ao propor uma taxonomia estruturada que classifica procedimentos em táticas e técnicas, além de prover uma base de conhecimento de inteligência tática de ameaça voltada ao desenvolvimento de modelos e metodologias [163, 18].

No contexto acadêmico, Maymí *et al.* (2017) [156] buscaram definir formalmente as TTPs e apresentaram um modelo genérico baseado em grafos direcionados para representar técnicas utilizadas em ataques. Embora forneçam uma estrutura conceitual útil para modelar o comportamento do adversário, os autores não exploraram métodos para extrair e mapear TTPs a partir de relatórios de CTI, o que limita a aplicabilidade prática do modelo.

6.1 Extração de TTPs

A necessidade de tornar as TTPs acionáveis impulsionou diversas pesquisas voltadas à extração dessas informações a partir de relatórios de CTI e outras fontes. Uma das primeiras iniciativas foi o *TTPDrill* [164], que propôs um sistema para identificar ações de

ameaça em textos não estruturados e mapear essas ações às táticas e técnicas do MITRE ATT&CK. Posteriormente, Noor *et al.* (2019) [165] apresentaram um *framework* que relaciona técnicas extraídas de repositórios de inteligência com mecanismos de detecção, permitindo prever a evolução de ataques cibernéticos.

Outros trabalhos avançaram na automação da extração de TTPs por meio de diferentes abordagens. Legoy *et al.* (2020) [166] propuseram a ferramenta *rcATT* para recuperar táticas e técnicas a partir de relatórios de ameaça, enquanto Zongxun *et al.* (2021) [167] desenvolveram um modelo para capturar ações de ataque e estruturar técnicas extraídas de relatórios de APTs. Sauerwein e Pfohl (2022) [168] investigaram a classificação automática de táticas e técnicas do MITRE ATT&CK, avaliando diferentes estratégias para aprimorar a identificação de técnicas adversárias.

Rani *et al.* (2023) [169] apresentaram o *TTPHunter*, uma abordagem para mapear frases extraídas de relatórios de APTs a um conjunto de técnicas do framework ATT&CK. Sua versão aprimorada, o *TTPXHunter* [170], expandiu a cobertura para um número maior de técnicas e introduziu melhorias na precisão da categorização. Outra iniciativa relevante é a ferramenta *open-source* TRAM (Threat Report ATT&CK Mapper) [171], desenvolvida pela MITRE, que auxilia analistas na categorização de técnicas em relatórios de CTI. O TRAM utiliza um modelo LLM (Large Language Model) pré-treinado para identificar automaticamente as 50 técnicas mais comuns do framework ATT&CK.

A extração automatizada de TTPs representa um avanço significativo no mapeamento de comportamentos adversários, facilitando a identificação de técnicas em relatórios de CTI. No entanto, essas abordagens ainda apresentam limitações quanto à precisão e à abrangência. Outra limitação é que as soluções atuais se restringem à identificação de táticas e técnicas, sem capturar os procedimentos empregados pelos adversários para implementá-las. Além disso, elas não consideram a identificação de múltiplas técnicas por procedimento nem o mapeamento de relações ou encadeamentos entre técnicas.

Assim, a interpretação humana continua sendo essencial para compreender nuances contextuais, identificar padrões emergentes e estabelecer conexões que podem escapar a modelos automatizados. Dessa forma, ferramentas automatizadas devem ser utilizadas como apoio à expertise dos analistas, aumentando a eficiência e a precisão do processo.

O mapeamento das TTPs é fundamental para compreender o comportamento das ameaças. No entanto, para que essas informações sejam efetivamente aplicadas na defesa cibernética, é necessário desenvolver métodos capazes de detectar TTPs a partir de dados observáveis.

6.2 Uso de TTPs na detecção de ameaças

Diversos estudos investigaram o uso de TTPs para aprimorar a detecção de ameaças, oferecendo maior contextualização e facilitando a identificação de padrões adversários.

Milajerdi *et al.* (2019) [172] apresentaram o *HOLMES*, um sistema de detecção de ataques APTs que correlaciona fluxos de informações suspeitas a um conjunto de técnicas adversárias, permitindo identificar atividades características de campanhas APT. A solução utiliza regras baseadas em TTPs, que possibilitam reconhecer, em grafos de proveniência (*provenance graphs*) gerados a partir de logs de auditoria, técnicas específicas associadas a diferentes etapas do ciclo de vida de um ataque.

Gianvecchio *et al.* (2019) [173] introduziram o conceito de *clusters* semânticos, que agrupam regras de alerta e rótulos de TTPs para fornecer uma visão estruturada da relação entre técnicas adversárias e evidências de baixo nível. Esse modelo foi avaliado em um ambiente de *cybergaming*, demonstrando sua utilidade na análise de estratégias ofensivas e defensivas.

No contexto de sistemas de detecção de intrusão (IDS), Leite *et al.* (2022) [19] propuseram um método para mapear eventos de IDS de rede a TTPs, possibilitando a criação de padrões de ataque específicos e tornando a CTI tática mais acionável para operações defensivas.

Arafune *et al.* (2022) [174] desenvolveram um *framework* de automação do Threat Hunting baseado em TTPs voltado a Sistemas de Controle Industrial (ICS). O *framework* utiliza um classificador baseado em ML para detectar possíveis ataques APTs e prever TTPs futuras que podem ser exploradas por adversários. A abordagem supervisionada se apoia em assinaturas geradas a partir de simulações de ataques a ICS, fundamentadas no MITRE ATT&CK.

Kurniawan *et al.* (2022) [175] apresentaram o *KRYSTAL*, um *framework* modular baseado em grafos de conhecimento (Knowledge Graph – KG) para análise tática de ataques e detecção de ameaças em dados de auditoria. A detecção utiliza regras SPARQL [176], convertidas de regras Sigma [177], para consultas em grafos de proveniência. A solução permite a reconstrução de cenários de ataque enriquecidos com integrações a outros grafos de conhecimento, como o *SEPSES CSKG* [178] e o *ATT&CK-KG* [179], possibilitando o mapeamento de eventos de segurança a TTPs.

Essas pesquisas demonstram o potencial das TTPs na detecção de ameaças, proporcionando uma visão mais detalhada e contextualizada dos comportamentos adversários. No entanto, a aplicação efetiva das TTPs na detecção ainda enfrenta desafios, especialmente no que se refere à sistematização da criação de regras acionáveis.

6.3 Metodologias para detecção baseada em TTPs

Embora diversas pesquisas tenham explorado a extração de TTPs de relatórios de CTI e seu uso na detecção de ameaças, a literatura ainda carece de metodologias sistemáticas que permitam transformar inteligência tática em regras acionáveis para Threat Hunting.

Strom *et al.* (2017) [180] introduziram uma abordagem que utiliza o *framework* ATT&CK para aprimorar a detecção de ameaças cibernéticas, com foco na identificação de atividades adversárias por meio de métodos analíticos baseados em comportamento. A pesquisa propõe uma metodologia fundamentada em cinco princípios e em experiências obtidas em exercícios de simulação conduzidos pela MITRE. A metodologia é composta por sete etapas organizadas ciclicamente: (i) identificar comportamentos, (ii) coletar dados, (iii) desenvolver análises, (iv) criar um cenário de emulação, (v) emular a ameaça, (vi) investigar o ataque e (vii) avaliar o desempenho.

Os produtos analíticos resultantes foram disponibilizados no MITRE Cyber Analytics Repository (CAR) [21], um repositório que reúne hipóteses analíticas, implementações em pseudocódigo, testes unitários e um modelo de dados, facilitando a adaptação das análises a diferentes plataformas. Essa abordagem demonstrou que a detecção baseada em TTPs pode superar as limitações das soluções tradicionais baseadas em IoCs, permitindo uma identificação mais precisa e eficaz de adversários ativos.

Daszczyszak *et al.* (2019) [149] propuseram uma metodologia para detecção baseada em TTPs, destacando as vantagens dessa estratégia em relação às abordagens tradicionais baseadas em IoCs e detecção de anomalias. O estudo enfatiza que os comportamentos adversários são mais difíceis de modificar do que atributos superficiais, tornando a detecção baseada em TTPs mais resiliente frente a ameaças adaptáveis. A metodologia proposta pelos autores é dividida em duas fases e sete etapas. A primeira fase dedica-se à caracterização das atividades maliciosas, reunindo informações de inteligência sobre comportamentos adversários, priorizando TTPs para o desenvolvimento analítico e especificando requisitos de dados para a caça. A segunda fase concentra-se na execução do Threat Hunting, incluindo a coleta de dados, a implementação de hipóteses e a avaliação de resultados. Apesar de utilizar CTI para compreender o comportamento do adversário, a metodologia não define um processo claro para transformar esse conhecimento em regras de detecção acionáveis.

Diferentemente das abordagens anteriores, a metodologia proposta nesta pesquisa fornece inteligência tática acionável que inclui regras de detecção baseadas nas TTPs da ameaça. Além disso, adota um modelo de dados próprio para registrar as informações de forma estruturada e consistente em um documento denominado *Runbook*, promovendo a padronização e o reuso do conhecimento produzido.

A Tabela 6.1 apresenta um quadro comparativo das principais características das metodologias analisadas — Strom *et al.* [180], Daszczyszak *et al.* [149] e a proposta deste trabalho. Observa-se que apenas a metodologia aqui proposta contempla recursos como a identificação de encadeamentos de TTPs e o desenvolvimento de regras de detecção. Considerando a complexidade dessas atividades, a proposta também incorpora o mapeamento de eventos com base em informações contextuais sobre o comportamento do ataque — aspecto ausente nas demais metodologias. Ademais, os trabalhos anteriores não apresentam um modelo de dados, o que dificulta a geração de saídas padronizadas e reutilizáveis.

Tabela 6.1: Características abrangidas pelas metodologias analisadas

	Strom [180]	Daszczyszak [149]	Esta pesquisa
Uso de CTI	✗	✓	✓
Identificação de TTP	✓	✓	✓
Encadeamento de TTP	✗	✗	✓
Definição de fontes de dados	✓	✓	✓
Análise baseada em comportamento	✓	✓	✓
Mapeamento de dados	✗	✗	✓
Desenvolvimento de regras de detecção	✗	✗	✓
Emulação adversária	✓	✗	✓
Coleta de dados	✗	✓	✓
Testes	✓	✓	✓
Avaliação	✓	✓	✓
Modelo de dados	✗	✗	✓
Saída padronizada	✗	✗	✓
Melhoria contínua	✓	✗	✓

Quanto à estrutura metodológica, esta pesquisa distingue-se das anteriores ao combinar um modelo cíclico de quatro fases com uma organização hierárquica composta por estágios, passos e ações. A Tabela 6.2 complementa o quadro anterior, apresentando um comparativo entre as estruturas das metodologias e destacando como a proposta atual oferece maior granularidade e um processo mais detalhado.

Tabela 6.2: Comparação da estrutura metodológica entre as abordagens

	Strom [157]	Daszczyszak [149]	Esta pesquisa
Esquema cíclico	✓	✗	✓
Estrutura hierárquica	✗	✓	✓
Número de fases/estágios	1	2	4
Número de passos	7	7	10
Número de ações	–	–	23

Esses quadros comparativos evidenciam que a metodologia proposta vai além das abordagens existentes, preenchendo lacunas identificadas na literatura. Ao propor um processo estruturado e reprodutível, busca-se viabilizar a aplicação prática do Threat Hunting baseado em TTPs, promovendo maior eficiência, escalabilidade e automação na detecção de comportamentos adversários.

6.4 Playbooks e Runbooks de Threat Hunting

Playbooks e runbooks são documentos formais que descrevem procedimentos técnicos e operacionais, oferecendo às equipes de segurança orientações claras e repetíveis para a execução de tarefas específicas. Embora muitas vezes empregados de forma intercambiável, esses termos apresentam distinções conceituais relevantes.

Um *runbook* é um guia operacional detalhado que descreve as ações necessárias para atingir um objetivo técnico específico [181]. Em contraste, o *playbook* possui caráter mais estratégico, apresentando descrições de alto nível e foco na coordenação de atividades, enquanto o runbook se concentra na execução prática, fornecendo instruções acionáveis e com menor nível de abstração. Essa natureza operacional permite ações mais rápidas e diretas, que podem ser realizadas manualmente ou automatizadas [182]. Como consequência, runbooks tendem a exigir atualizações mais frequentes do que playbooks [183].

Apesar dessas distinções conceituais, o termo *playbook* consolidou-se no ecossistema de ferramentas comerciais e na comunidade de cibersegurança, especialmente nas atividades de resposta a incidentes. Assim, é comum que documentos denominados "playbooks" desempenhem, na prática, funções equivalentes às de runbooks, e vice-versa.

No contexto do Threat Hunting, o projeto *ThreatPlays* propõe playbooks com o objetivo de tornar o processo de caça mais sistemático e repetível [184]. Esses playbooks, disponibilizados em formato Markdown, seguem uma estrutura composta por: propósito da investigação, dados necessários (fontes de eventos), considerações de coleta, técnicas de análise, descrição da estratégia de detecção, notas adicionais e referências.

Outro projeto de código aberto relevante é o *Threat Hunter Playbook*, cujo objetivo é compartilhar a lógica de detecção, o conhecimento técnico sobre o adversário e os recursos necessários para tornar o desenvolvimento de detecções mais eficiente [185]. O projeto disponibiliza uma coleção de playbooks padronizados que incluem: formulação de hipóteses de ataque, contexto técnico associado, descrição do *tradecraft* ofensivo, links para datasets de segurança, análises detalhadas com estratégias e *queries* de detecção, informações sobre possíveis falsos positivos, links para regras correlacionadas e referências adicionais. Os playbooks são apresentados em formato Markdown, acompanhados de

arquivos no formato YAML que reúnem metadados e incluem, entre outras informações, a lista de táticas e (sub)técnicas do framework ATT&CK associadas.

Outra iniciativa de destaque é o MITRE CAR (Cyber Analytics Repository) [21], concebido como uma base de conhecimento de análises fundamentadas no framework ATT&CK. O CAR define um modelo de dados próprio voltado a fornecer um conjunto de análises validadas e documentadas para detecção de comportamentos adversários. Cada análise é representada em formato YAML, seguindo uma estrutura padronizada que contempla: formulação de hipóteses, definição do domínio da informação (*e.g.*, host, rede, processo, externo), táticas e técnicas associadas no ATT&CK, bem como a descrição da implementação em pseudocódigo ou em linguagens específicas para diferentes ferramentas [186]. Além disso, o CAR inclui testes unitários que acionam as hipóteses analisadas, promovendo verificação e reprodutibilidade das detecções.

Diferentemente das iniciativas anteriores, o CACAO (Collaborative Automated Course of Action Operations) [187], desenvolvido pelo OASIS[1], tem como foco a definição de um padrão aberto para orquestração e automação de cursos de ação (COA) em cibersegurança. Seu escopo é mais abrangente, englobando atividades de detecção, investigação, prevenção, mitigação, remediação, emulação e notificação [188, 189]. Os playbooks do CACAO são representados em formato JSON e descrevem *workflows* compostos por etapas manuais ou automatizadas, que podem ser executadas de forma sequencial, paralela ou condicional. Além do fluxo de execução, a especificação contempla metadados, agentes, alvos, informações de autenticação e outros elementos estruturais [187], oferecendo uma base padronizada para o compartilhamento e a automação de playbooks entre diferentes ferramentas e organizações.

A análise dessas iniciativas evidencia que, embora todas possam ser aplicadas à atividade de Threat Hunting, ainda não existe uma base comum e padronizada de informações compartilhadas entre os diferentes modelos de playbooks e runbooks. A Tabela 6.3 sintetiza as informações suportadas nativamente por cada modelo, contrastando-as com a abordagem proposta nesta pesquisa.

O *ThreatPlays* apresenta um modelo simples que, embora aborde estratégias de detecção baseadas em hipóteses, não contempla regras acionáveis para sua implementação. O *Threat Hunter Playbook* e o *MITRE CAR*, por sua vez, oferecem modelos alinhados ao framework ATT&CK e incluem regras de detecção descritas tanto em pseudocódigo (representando a lógica de funcionamento) quanto em linguagens específicas utilizadas em plataformas de Threat Hunting. Apesar de atenderem aos propósitos originais para os quais foram concebidos, suas estruturas mais compactas não são adequadas para representar ameaças complexas, como malwares sofisticados e campanhas de APTs, que envolvem

[1] `www.oasis-open.org`

Tabela 6.3: Informações suportadas nativamente por diferentes modelos de Playbooks/-Runbooks

	ThreatPlays [184]	Threat Hunter Playbook [185]	MITRE CAR [21]	CACAO [187]	Esta pesquisa
Hipóteses	✓	✓	✓	✓	✓
Plataformas	✗	✓	✓	✗	✓
Táticas do ATT&CK	✗	✓	✓	✗	✓
Técnicas do ATT&CK	✗	✓	✓	✗	✓
Subtécnicas do ATT&CK	✗	✓	✓	✗	✓
Procedimentos das TTPs	✗	✗	✗	✗	✓
Encadeamento de TTPs	✗	✗	✗	✗	✓
Técnicas do MITRE D3FEND [190]	✗	✗	✓	✗	✗
Análises	✓	✓	✓	✗	✓
Regras de detecção	✗	✓	✓	✓	✓
Associação entre regras e TTPs	✗	✓*	✓*	✗	✓
Linguagem da regra	✗	✓	✓	✓*	✓
Fonte de dados/eventos	✓	✗	✗	✗	✓
Encadeamento de regras	✗	✗	✗	✓	(✓)
Informações de teste/validação das regras	✗	✓	✓	✗	✓
Anotações ou comentários	✓	✓	✗	✗	✓
Referências externas	✓	✓	✗	✓	✓
Relacionamento com outros playbooks/runbooks	✗	✗	✗	✓	✓

✓* Parcialmente suportado. (✓) Indiretamente suportado.

múltiplas etapas, táticas e técnicas inter-relacionadas, exigindo um conjunto mais extenso e diversificado de regras de detecção.

O *CACAO*, em contraste, apresenta uma estrutura mais robusta e flexível, capaz de lidar com ameaças complexas e fluxos de ação variados. No entanto, no contexto do Threat Hunting, seu foco reside na definição de *workflows* de detecção que podem incluir tanto ações manuais quanto comandos automatizados, semelhantes às regras disponíveis nos projetos *Threat Hunter Playbook* e *MITRE CAR*. Ainda assim, o modelo não contempla explicitamente o mapeamento de TTPs, o que implica maior dependência de informações externas provenientes de inteligência tática. Além disso, apresenta suporte limitado a linguagens de regras de detecção [187], restringindo sua aplicabilidade, sobretudo em abordagens baseadas em TTPs.

O runbook proposto nesta pesquisa busca superar essas limitações ao oferecer um modelo mais abrangente e integrador, que unifica elementos presentes nas iniciativas analisadas em uma estrutura única, orientada à produção de inteligência tática acionável para o Threat Hunting. Diferentemente dos modelos existentes, o modelo de dados proposto suporta não apenas as táticas e técnicas do MITRE ATT&CK, mas também os procedimentos — equivalentes a descrições da implementação das técnicas — informação que não é contemplada de forma explícita nas demais abordagens. Outro diferencial relevante

é o suporte ao encadeamento de TTPs, recurso essencial para a representação de ameaças complexas. O encadeamento de regras de detecção, por sua vez, é suportado de forma indireta, uma vez que, no runbook proposto, as regras são associadas às respectivas TTPs. Outrossim, a estruturação padronizada dessas informações favorece a interoperabilidade, a automatização e a reutilização do conhecimento, viabilizando sua aplicação em cenários de Threat Hunting mais complexos.

Capítulo 7

Modelo de Dados

Esta seção apresenta o modelo de dados do Runbook produzido pela metodologia proposta. O objetivo é fornecer uma estrutura capaz de reunir informações essenciais sobre a ameaça, suas TTPs, as regras de detecção correspondentes, as informações de validação das regras e as referências de CTI utilizadas em todo o processo, de modo a constituir uma fonte de inteligência tática acionável para o analista de Threat Hunting.

Um Runbook de Threat Hunting deve conter instruções ou comandos aplicáveis a ferramentas específicas, com o objetivo de identificar rastros de uma determinada ameaça em grandes volumes de dados observáveis. No contexto desta pesquisa, essas instruções correspondem a regras de detecção elaboradas a partir de Táticas, Técnicas e Procedimentos (TTPs) associados à ameaça.

As TTPs são extraídas de materiais de CTI e encapsuladas em uma estrutura denominada TTP, que inclui, além da tríade Tática–Técnica–Procedimento, outros elementos relevantes para o Threat Hunting baseado em TTP, como o encadeamento de TTPs e a vinculação às regras de detecção desenvolvidas.

A Figura 7.1 apresenta uma visão simplificada dessa estrutura, destacando as Regras de Detecção validadas referentes às TTPs mapeadas da ameaça.

Com base nessa visão, foram identificadas cinco entidades principais que compõem o modelo de dados: Ameaça, TTP, Regra de Detecção, Validação e Referência. O relacionamento entre essas entidades é ilustrado no diagrama da Figura 7.2.

A entidade Ameaça agrega metadados, informações gerais da ameaça e relacionamentos com outras ameaças. Cada Ameaça pode conter n TTPs, que, por sua vez, podem estar relacionadas entre si. Cada TTP pode originar n Regras de Detecção, associadas a uma entidade Validação, responsável por armazenar os resultados da terceira fase da metodologia. As entidades Ameaça, TTP e Validação podem ter n Referências vinculadas, registrando as fontes de CTI utilizadas. A entidade Regra de Detecção, por conter informações produzidas com base na TTP correspondente, não mantém referências próprias.

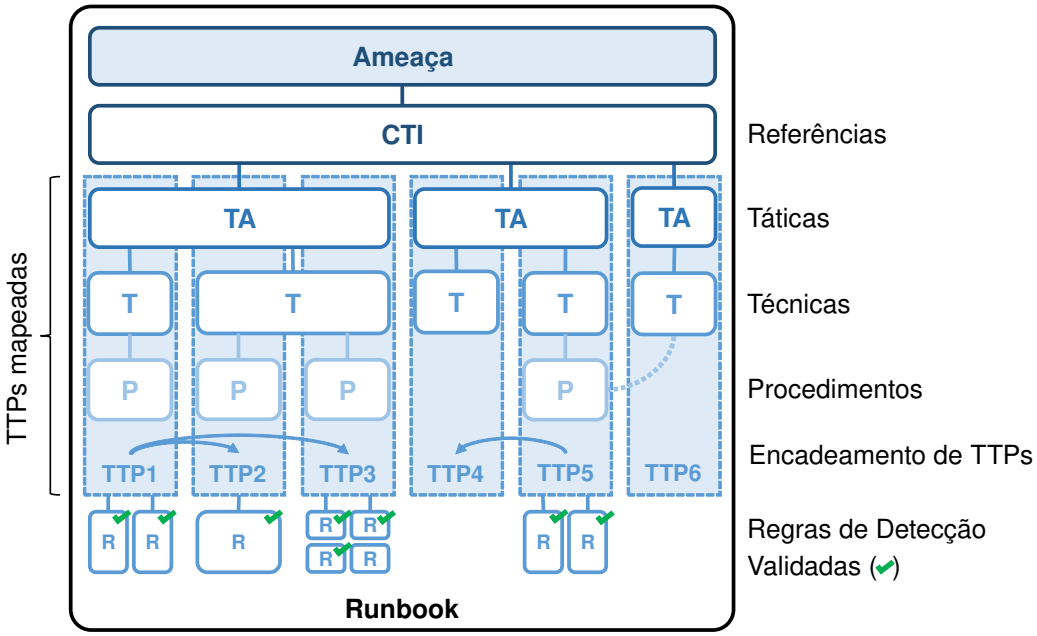


Figura 7.1: Estrutura simplificada do Runbook

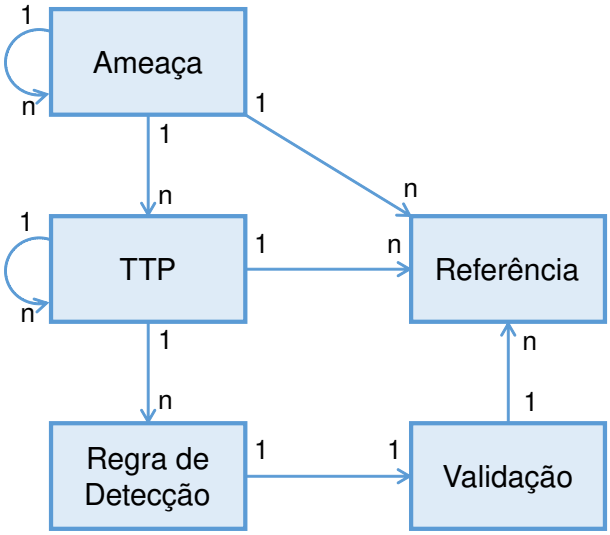


Figura 7.2: Diagrama de entidades e relacionamentos do modelo de dados

Esse mecanismo é essencial para o processo de melhoria contínua, pois permite a revisão, correção e refinamento das informações.

7.1 Diagrama de Classes

Enquanto o diagrama de entidades e relacionamentos mostra como os principais objetos se conectam, o Diagrama de Classes da Figura 7.3 detalha os atributos associados a cada entidade, evidenciando como os dados são organizados na formação do Runbook. Essa

estrutura foi projetada para ser simples e flexível, permitindo o registro incremental das informações ao longo das quatro fases da metodologia.

Além disso, o uso de identificadores únicos para as classes Ameaça, TTP, Regra de Detecção e Referência não apenas possibilita o referenciamento cruzado entre entidades no Runbook, como também viabiliza seu armazenamento em banco de dados, favorecendo o reuso do conhecimento produzido.

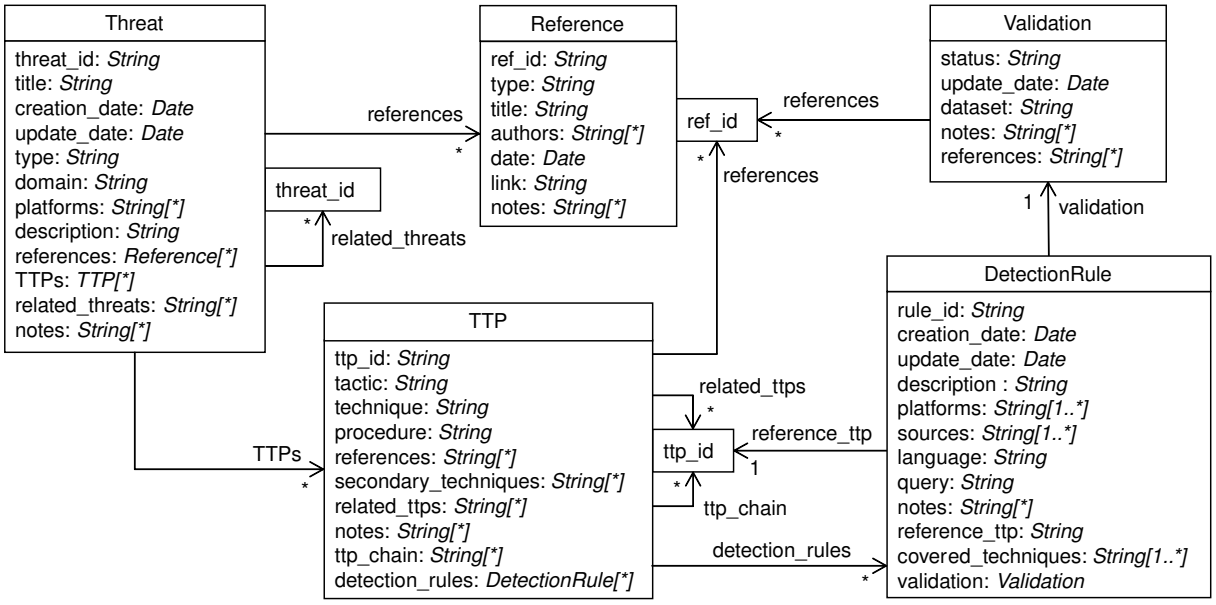


Figura 7.3: Diagrama de Classes do Runbook

A classe `Threat` é responsável por armazenar metadados (como datas de criação e modificação), informações gerais (título, descrição, tipo, domínio, plataformas e anotações) e relacionamentos com outras entidades, incluindo as TTPs mapeadas, referências e ameaças relacionadas. Seus atributos são:

- **`threat_id`**: Identificador único do objeto `Threat`, no formato `THR-xxxx-xxxx-xxxx`, onde x = [0 . . 9].
- **`title`**: Título da ameaça.
- **`creation_date`**: Data de criação do Runbook.
- **`update_date`**: Data da última atualização do Runbook.
- **`type`**: Tipo da ameaça, *e.g.* Malware.
- **`domain`**: Domínio ao qual a ameaça pertence, *e.g.* Enterprise, Mobile.
- **`platforms`**: Lista de plataformas em que a ameaça pode atuar.
- **`description`**: Descrição da ameaça.
- **`notes`**: Lista de anotações relevantes sobre a ameaça.

- **references**: Lista de objetos da classe `Reference`.
- **related_threats**: Lista de identificadores (`threat_id`) de outras ameaças relacionadas.
- **TTPs**: Lista de objetos da classe `TTP`.

A classe `TTP` desempenha papel central na metodologia, armazenando a Tática e a Técnica de acordo com a taxonomia do MITRE ATT&CK, além do Procedimento em forma textual, que descreve em detalhe a implementação da técnica com base nas CTIs referenciadas.

Essa classe também suporta cenários em que um único Procedimento implementa múltiplas Técnicas, permitindo o registro de técnicas secundárias. Adicionalmente, possibilita o registro de TTPs relacionadas e encadeadas, a vinculação a regras de detecção e referências, bem como a inclusão de anotações. Seus atributos são:

- **ttp_id**: Identificador único do objeto `TTP`, no formato `TTP-xxxx-xxxx-xxxx`, onde x = [0 . . 9].
- **tactic**: Código da Tática, conforme o Framework ATT&CK, no formato `TAxxxx`.
- **technique**: Código da Técnica (ou Subtécnica), conforme o Framework ATT&CK, no formato `Txxxx` (ou `Txxxx.xxx`).
- **procedure**: Descrição do Procedimento.
- **references**: Lista dos identificadores (`reference_id`) das referências utilizadas na descrição da TTP.
- **secondary_techniques**: Lista de técnicas ou subtécnicas secundárias implementadas pelo Procedimento descrito.
- **related_ttps**: Lista dos identificadores (`ttp_id`) de outras TTPs da mesma ameaça que tem relação com esta.
- **notes**: Lista de anotações relevantes sobre esta TTP.
- **ttp_chain**: Lista de identificadores (`ttp_id`) das TTPs subsequentes — este campo registra as possíveis TTPs encadeadas pela execução desta.
- **detection_rules**: Lista de objetos da classe `DetectionRule`.

A classe `DetectionRule` contém, além da query propriamente dita, diversas informações relevantes para sua aplicação em atividades de Threat Hunting, como a linguagem utilizada, descrição, plataformas suportadas, fontes de coleta necessárias, técnicas abrangidas e anotações.

Também são armazenados metadados (datas de criação e modificação) e uma referência à TTP que originou a regra. Seus atributos são:

- **`rule_id`**: Identificador único do objeto `DetectionRule`, no formato `DTR-xxxx-xxxx-xxxx`, onde x = [0 . . 9]..
- **`creation_date`**: Data de criação da Regra de Detecção.
- **`update_date`**: Data da última atualização da Regra de Detecção.
- **`description`**: Descrição em alto nível da Regra de Detecção, podendo incluir a estratégia ou abordagem utilizada.
- **`platforms`**: Lista de plataformas às quais a Regra de Detecção é aplicável.
- **`sources`**: Lista das fontes que fornecem os dados buscados pela regra. Exemplos: Windows Security Auditing, Sysmon.
- **`language`**: Linguagem utilizada na formulação da *query*. Exemplos: Sigma, SparkSQL, SPL, ES/QL, KQL.
- **`query`**: Regra de Detecção escrita na linguagem escolhida.
- **`notes`**: Lista de anotações relevantes sobre a Regra de Detecção.
- **`reference_ttp`**: Identificador (`ttp_id`) da TTP de referência.
- **`covered_techniques`**: Lista dos códigos das técnicas, conforme o Framework ATT&CK, que a regra visa detectar.
- **`validation`**: Objeto da classe `Validation`.

A classe `Validation` está sempre associada à classe `DetectionRule` e concentra informações sobre a avaliação da regra, incluindo status, data de modificação, dataset utilizado (quando aplicável), anotações e referências. Seus atributos são:

- **`status`**: Situação da Regra de Detecção. Valores possíveis: '*unchecked*', '*validated*', '*failed*' ou '*outdated*'.
- **`update_date`**: Data da última atualização dos dados de validação.
- **`dataset`**: Nome ou link do dataset utilizado para a validação.
- **`notes`**: Lista de anotações relevantes sobre a validação.
- **`references`**: Lista dos identificadores (`reference_id`) das referências utilizadas no processo de validação.

Por fim, a classe `Reference` registra os dados sobre as fontes de CTI utilizadas, incluindo tipo, título, link, data, autores e anotações. Seus atributos são:

- **`ref_id`**: Identificador único do objeto `Reference`, no formato `REF-xxxx-xxxx-xxxx`, onde x = [0 . . 9].
- **`type`**: Tipo do material de referência, *e.g.* Report, Dataset'.

- **`title`**: Título do material.
- **`authors`**: Lista de autores.
- **`date`**: Data do material.
- **`link`**: Link de referência para consulta ou download.
- **`notes`**: Lista de anotações úteis sobre a referência.

O modelo de dados proposto fornece uma estrutura abrangente que permite ao Runbook armazenar as Táticas, Técnicas e Procedimentos (TTPs) de uma ameaça, e as Regras de Detecção correspondentes, que tornam essa inteligência tática acionável para o Threat Hunting baseado na detecção de TTPs. O próximo capítulo define a metodologia correspondente, que orienta profissionais de cibersegurança na produção e manutenção desses Runbooks.

Capítulo 8

Metodologia

A Seção 1.3 apresentou a visão geral da metodologia proposta, estruturada em quatro fases principais para a produção e atualização do Runbook de Threat Hunting, contendo regras capazes de detectar, nos dados coletados de diversas fontes, comportamentos adversários baseados nas TTPs mapeadas da ameaça. Essas fases tornam a metodologia mais objetiva, ao guiar os analistas na execução dos processos em uma sequência lógica e mantê-los focados no objetivo final.

Para reduzir a complexidade do processo, cada fase é organizada em três níveis de abstração: estágios, passos e ações. A organização hierárquica é ilustrada na Figura 8.1.

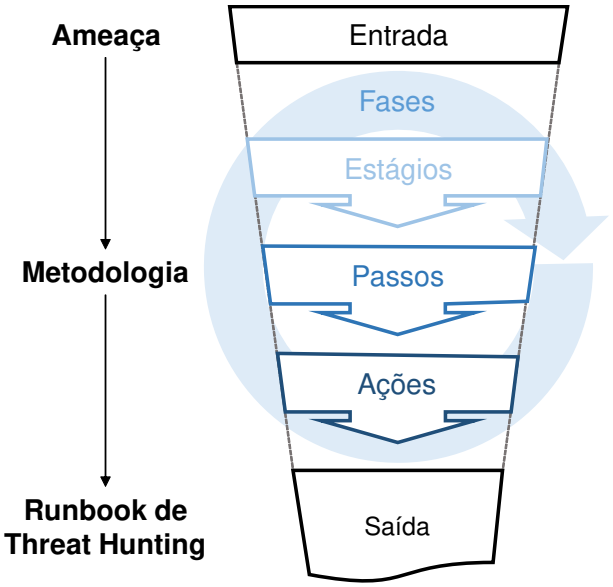


Figura 8.1: Estrutura hierárquica da metodologia

Essa hierarquização confere maior clareza e facilita o gerenciamento da metodologia, pois permite diferentes níveis de granularidade. Além disso, a divisão em etapas sucessivas e menos complexas simplifica a execução e possibilita a distribuição das tarefas de

acordo com a especialidade dos analistas, que podem atuar de forma independente ou colaborativa.

A Figura 8.2 ilustra o primeiro nível da estrutura hierárquica da metodologia, composta por quatro estágios principais: (i) mapeamento ameaça-TTP, (ii) mapeamento TTP-dados, (iii) validação e (iv) consolidação dos resultados. O esquema evidencia o fluxo de execução dos estágios em consonância com as fases do ciclo de melhoria contínua, ao mesmo tempo em que preserva a flexibilidade para o retorno a etapas anteriores, quando necessário, a fim de realizar ajustes ou correções.

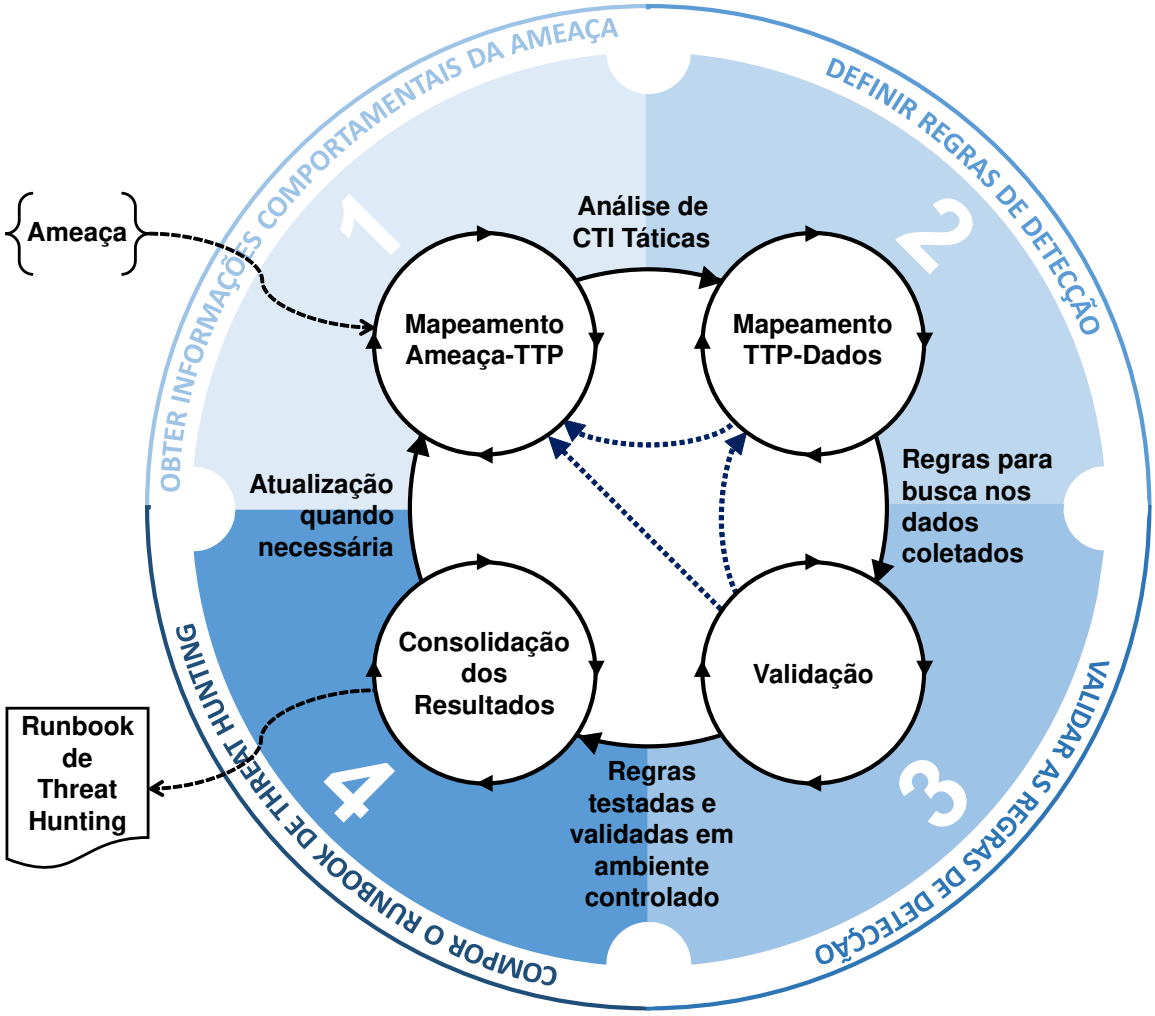


Figura 8.2: Diagrama macro da metodologia

Nas subseções seguintes, cada estágio é detalhado com a descrição de seus passos e ações, que orientam a produção dos resultados esperados em cada fase da metodologia.

8.1 Estágio 1: Mapeamento Ameaça–TTP

Este estágio tem como principal objetivo mapear as Táticas, Técnicas e Procedimentos (TTPs) da ameaça escolhida com base nas informações de inteligência disponíveis. Esse processo pode ser dividido em três passos:

1. **Obtenção de Conhecimento:** Este passo envolve a busca de materiais especializados, como relatórios de CTI, que forneçam detalhes sobre como a ameaça é perpetrada. O objetivo é reunir informações relevantes que permitam caracterizar de forma precisa os comportamentos adversários.

2. **Mapeamento de TTP:** Este passo consiste em identificar as Táticas, as Técnicas e os Procedimentos (TTPs) nos materiais reunidos no passo anterior, com o objetivo de mapear os comportamentos adversários que compõem a referida ameaça.

3. **Identificação de Relacionamentos:** Neste passo são identificadas possíveis relações de dependência e/ou encadeamento entre as TTPs identificadas.

O diagrama da Figura 8.3 ilustra o fluxo de ações deste primeiro estágio da metodologia. As ações, representadas por retângulos, são organizadas em camadas verticais que representam a sequência de passos que compõem o estágio.

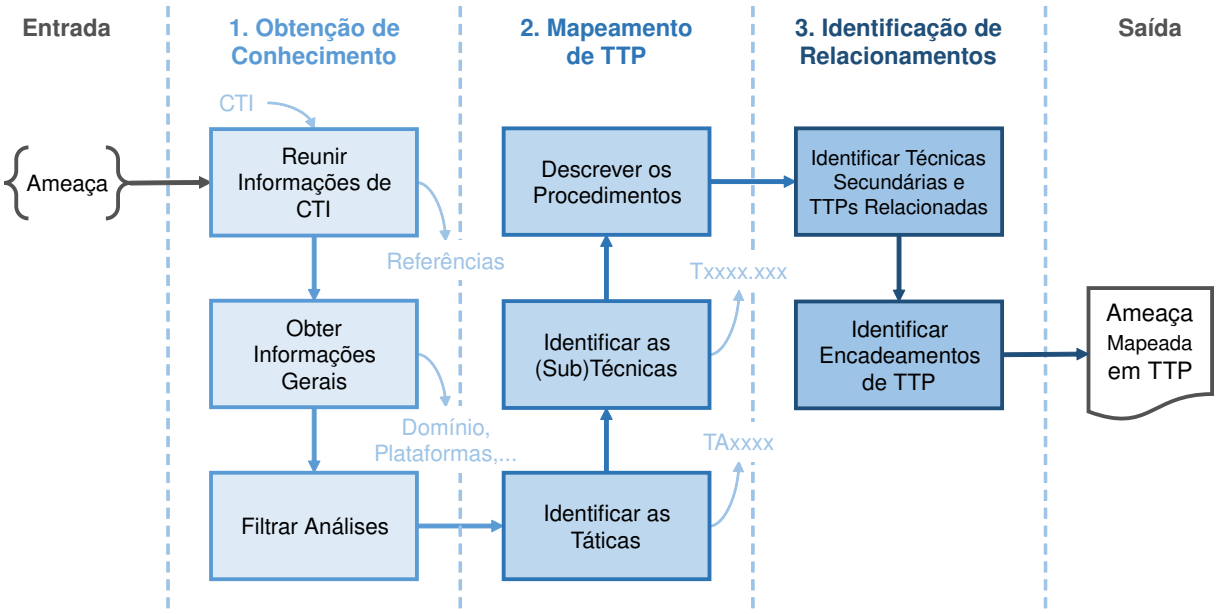


Figura 8.3: Fluxo de ações do 1º estágio: 'Mapeamento Ameaça-TTP'

As ações de cada passo que compõem o presente estágio são detalhadas nas subseções seguintes. A última subseção descreve a saída do estágio, denominada 'Ameaça Mapeada em TTP'.

8.1.1 Obtenção de Conhecimento

Neste passo, os analistas devem buscar informações sobre a ameaça escolhida, a fim de obter conhecimentos relevantes do ponto de vista tático, que permitam compreender o comportamento do agente da ameaça — essenciais para o processamento das etapas subsequentes da metodologia.

A metodologia prioriza o uso do conhecimento já produzido pela comunidade de segurança e, portanto, adota para este passo a seguinte sequência de ações:

1. Reunir Informações de CTI;
2. Obter Informações Gerais;
3. Filtrar Análises.

1. Reunir Informações de CTI

A primeira ação é reunir informações de CTI sobre a ameaça. Para isso, os analistas de segurança devem buscar fontes confiáveis de inteligência de ameaça cibernética, como, por exemplo:

- Empresas de segurança: Mandiant[1], CrowdStrike[2], Kaspersky[3], Symantec[4], ...
- Organizações governamentais: CISA[5] (Agência de Segurança Cibernética e Infraestrutura dos Estados Unidos), NCSC[6] (Centro Nacional de Segurança Cibernética do Reino Unido), ...
- Plataformas de Inteligência de Ameaça Cibernética: OpenCTI[7], MITRE ATT&CK[8], ...

As plataformas de inteligência fornecem uma interface unificada para acessar informações de várias fontes de CTI e, por isso, são a opção mais indicada para o início da busca. Normalmente, as informações obtidas são links para páginas HTML que contêm a análise, ou relatórios em formato PDF. Os materiais selecionados que forem utilizados no mapeamento das TTPs devem ser registrados no Runbook como referências.

[1] `https://www.mandiant.com/resources/blog`
[2] `https://www.crowdstrike.com/resources/reports/`
[3] `https://ics-cert.kaspersky.com/publications/reports/`
[4] `https://symantec-enterprise-blogs.security.com/blogs/threat-intelligence`
[5] `https://www.cisa.gov/news-events/cybersecurity-advisories`
[6] `https://www.ncsc.gov.uk/section/keep-up-to-date/malware-analysis-reports`
[7] `https://filigran.io/solutions/products/opencti-threat-intelligence/`
[8] `https://attack.mitre.org`

2. Obter Informações Gerais

A próxima ação é obter informações gerais relevantes sobre a ameaça, como, por exemplo:

- **Descrição:** texto sucinto que descreva resumidamente a ameaça.
- **Domínio:** refere-se à área de aplicação/atuação da ameaça. A taxonomia do MITRE ATT&CK classifica em três domínios: Mobile, ICS (Industrial Control Systems) e Enterprise.
  - **Mobile:** engloba as ameaças que atuam contra dispositivos móveis.
  - **ICS:** engloba as ameaças que atuam contra sistemas de controle industrial.
  - **Enterprise:** engloba as demais ameaças, que atuam contra infraestruturas de TI tradicionais.
- **Plataformas:** referem-se aos tipos de sistemas ou ambientes em que as ameaças podem atuar. Alguns exemplos de plataformas no domínio Enterprise são: Windows, macOS, Linux, Azure, Microsoft Office, Network, Containers, etc.

A descrição da ameaça tem por objetivo aumentar a clareza e facilitar a utilização do Runbook. A definição do domínio é necessária para selecionar qual sub-framework do MITRE ATT&CK deve ser utilizado no passo de mapeamento das TTPs. Já as plataformas de atuação determinam o universo dos dados que podem ser observados e utilizados para o desenvolvimento das regras de detecção.

3. Filtrar Análises

A última ação deste passo é filtrar as análises contidas nos relatórios de CTI reunidos. É fato que há uma enorme variação de forma e conteúdo nos relatórios de segurança e que nem tudo é informação útil para a criação do Runbook. Sendo assim, é fundamental que o analista saiba filtrar o que é relevante nesta metodologia.

O objetivo deste estágio é mapear as TTPs da ameaça; portanto, o analista deve focar nas informações relacionadas a táticas, técnicas e procedimentos. Além disso, ele deve ter em mente que o objetivo final é a aplicação em Threat Hunting, atividade que busca por rastros de ameaças que ultrapassaram as barreiras de segurança existentes. Informações sobre funcionamento, *modus operandi*, ações e atividades que afetem o alvo devem ser priorizadas na análise. Vale ressaltar que a abordagem utilizada nesta metodologia não considera os IoCs como informações relevantes, tendo em vista os problemas abordados na Seção 1.1.

8.1.2 Mapeamento de TTP

Neste passo, os analistas devem identificar as Táticas, Técnicas e Procedimentos que compõem a ameaça, usando como referência os conhecimentos reunidos no passo anterior. Este é o principal passo deste estágio, pois é onde ocorre o mapeamento propriamente dito. Para isso, devem ser realizadas as seguintes ações:

1. Identificar as Táticas;
2. Identificar as (Sub)Técnicas;
3. Descrever os Procedimentos.

1. Identificar as Táticas

A primeira ação é identificar as táticas utilizadas. Para isso, o analista deve, primeiramente, procurar nos textos selecionados sobre a ameaça por comportamentos, adotando uma visão centrada em ações do adversário ou software, e não em indicadores atômicos, como endereços IP. Em seguida, deve buscar entender o objetivo ou motivo daquela atuação do adversário ou do software e associá-lo a uma tática do framework ATT&CK.

2. Identificar as (Sub)Técnicas

A segunda ação no processo de mapeamento de TTP é identificar qual técnica, ou subtécnica, melhor corresponde ao comportamento observado. Este processo requer uma análise meticulosa do comportamento adversário e uma compreensão clara das (sub)técnicas listadas no framework ATT&CK.

Esta ação pode apresentar certos desafios, tendo em vista a grande quantidade de (sub)técnicas existentes e a similaridade entre algumas delas; contudo, a identificação da tática na etapa anterior reduz o escopo da análise, e a utilização de exemplos disponíveis no site do ATT&CK auxilia na identificação da (sub)técnica mais apropriada.

O processo de identificação de técnicas nos relatórios de ameaça (*threat reports*) é trabalhoso e suscetível a erros de interpretação. Visando superar essas questões, diversos trabalhos acadêmicos exploraram a aplicação de técnicas de IA, como NLP e ML, para a identificação automatizada das Táticas e Técnicas em textos não estruturados, conforme abordado na Seção 6.1 desta pesquisa.

Embora ferramentas automatizadas sejam extremamente valiosas para aumentar a eficiência do mapeamento de TTPs, elas possuem limitações e também são suscetíveis a erros, haja vista que sua eficácia depende, em grande parte, da qualidade e da quantidade de dados de treinamento disponíveis. Por esses motivos, esta metodologia adota como

padrão o processo manual de identificação das TTPs, deixando facultada a utilização de ferramentas complementares de extração automatizada, a critério do analista.

Cabe ressaltar, no entanto, a capacidade humana de fazer conexões intuitivas e interpretar nuances que a inteligência artificial pode não ser capaz de captar. Portanto, essas ferramentas devem ser vistas como complementares à análise humana e não como substitutas, principalmente pelo fato de as ameaças cibernéticas estarem em constante evolução.

3. Descrever os Procedimentos

A última ação do passo de mapeamento de TTPs é descrever os procedimentos referentes às (sub)técnicas identificadas. Como o framework ATT&CK não define código para rotulação de procedimentos, o registro deve ser feito em forma de texto descritivo, que pode ser criado pelo analista ou extraído dos relatórios consultados.

Os relatórios de ameaças podem abranger diversos tipos de inteligência, variando conforme a ênfase do relatório e a profundidade da análise. No entanto, para esta ação da metodologia, devem ser buscadas informações sobre o *modus operandi*, o funcionamento e a execução da ameaça, as quais são fornecidas pela inteligência tática.

Essas informações nem sempre estão disponíveis com o grau de detalhamento desejado, o que pode dificultar ou até mesmo impossibilitar a identificação de algumas Táticas, Técnicas e Procedimentos. Porém, quando disponíveis no relatório de ameaça, os procedimentos podem ser destacados diretamente do texto. A identificação do procedimento no texto requer a compreensão, por parte do analista, de que os procedimentos são implementações específicas de técnicas e subtécnicas para atingir um objetivo tático.

8.1.3 Identificação de Relacionamentos

Há dois tipos de relacionamentos que este passo busca identificar. O primeiro é quando há outras Técnicas relacionadas ao Procedimento destacado. O segundo é quando a execução de uma TTP está relacionada, ou acarreta, a execução de outra. Para isso, devem ser realizadas as seguintes ações:

1. Identificar Técnicas Secundárias e TTPs Relacionadas;
2. Identificar Encadeamentos de TTP.

1. Identificar Técnicas Secundárias e TTPs Relacionadas

Criminosos cibernéticos podem combinar diferentes técnicas para atingir seus objetivos com mais eficácia. Dependendo de como são executadas, as implementações podem con-

ter indícios de outras técnicas. É comum que atacantes combinem múltiplas técnicas, como evasão, para aumentar a efetividade de suas ações. Sendo assim, é possível que o mesmo procedimento englobe mais de uma técnica, sendo uma principal e as demais complementares ou secundárias.

Nesta etapa, o analista deve identificar a presença de técnicas secundárias nos procedimentos das TTPs mapeadas e, caso positivo, registrá-las no campo `secondary_techniques`.

Se a técnica secundária identificada estiver associada a um procedimento de outra TTP documentada, o identificador de cada TTP deve ser registrado no campo `related_ttps`, estabelecendo a relação entre elas.

2. Identificar Encadeamentos de TTP

O objetivo desta ação é identificar se há relação de ordem entre as TTPs. Para alcançar esse objetivo, é necessária uma análise minuciosa dos relatórios de CTI, buscando relações temporais entre as Técnicas identificadas nas etapas anteriores. O analista deve observar a ordem de execução dos Procedimentos, pois são eles que implementam as Técnicas, e discernir se existe uma sequência fixa de execução.

Essa tarefa possui diversos desafios que decorrem da natureza complexa e dinâmica das ameaças cibernéticas. Além disso, nem todos os adversários seguem uma sequência fixa de Procedimentos, muitos adaptam suas Táticas e Técnicas em resposta às defesas que encontram. Por esse motivo, o analista deve estar atento às minúcias das informações de CTI disponíveis.

Nesta ação, o foco do analista deve ser verificar se um Procedimento desencadeia a execução de outro. Se sim, então existe uma relação de dependência, indicando que as TTPs são encadeadas naquela ordem. Uma forma de facilitar a identificação das relações é procurar certos elementos linguísticos nos relatórios, como, por exemplo, os termos *then*, *next*, *after*, *before*, *followed by*, *and*, entre outros. No entanto, a interpretação do analista e a qualidade do relatório são fundamentais.

Alguns Procedimentos podem depender de condicionantes para a sua execução. As condicionantes são critérios que, se satisfeitos, decidem qual a próxima ação, criando desvios de comportamento. Termos como *if*, *whether*, *or*, *otherwise*, entre outros, podem ser procurados. Dessa forma, uma TTP pode encadear diversas outras TTPs conforme as condições especificadas.

A identificação de relações de dependência entre as TTPs de uma ameaça é uma tarefa complexa que exige uma análise cuidadosa e detalhada. No entanto, os *insights* obtidos a partir dessa análise proporcionam uma compreensão mais profunda do comportamento do adversário.

No contexto do Threat Hunting, os encadeamentos mapeados auxiliam os analistas na busca por novas evidências, pois indicam possíveis comportamentos subsequentes. Isso possibilita a identificação da ameaça de maneira mais rápida e precisa. No caso de um ataque em andamento, ter uma base de encadeamentos de TTPs permite que os analistas prevejam as ações futuras, possibilitando uma resposta mais rápida e eficaz das equipes de segurança, interrompendo o ataque e mitigando os danos.

8.1.4 Ameaça Mapeada em TTPs

Para exemplificar o processo de mapeamento de TTPs realizado neste estágio da metodologia, considere o trecho do relatório de CTI da Mandiant sobre a campanha de ciberespionagem "Operation Double Tap" [4], apresentado na Figura 8.4.

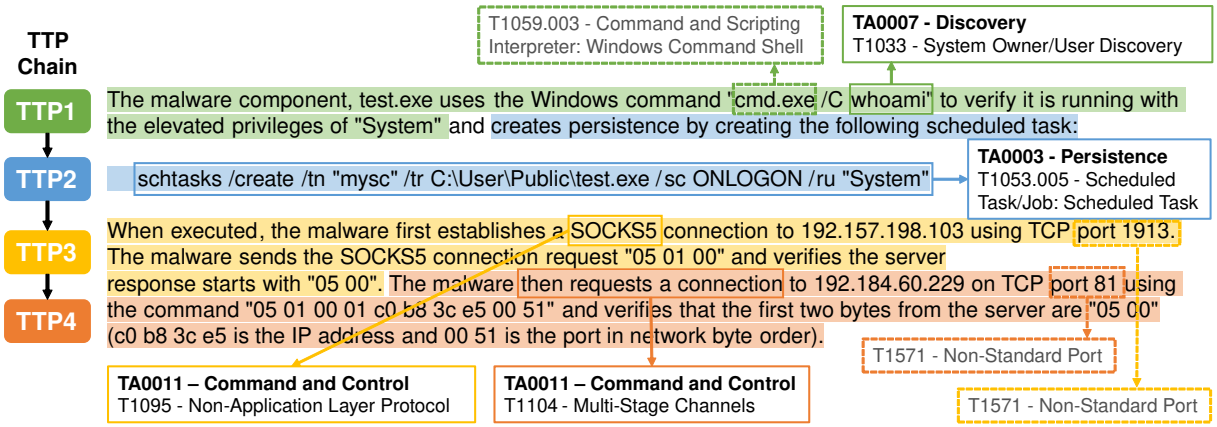


Figura 8.4: Exemplo de mapeamento de TTP de um trecho do relatório da ameaça "Operation Double Tap" [4]

Nesse trecho são identificadas três táticas: *Discovery* (TA0007), *Persistence* (TA0003) e *Command and Control* (TA0011). Associada à tática de *Discovery* está a técnica *System Owner/User Discovery* (T1033). Com base no procedimento destacado em verde, definiu-se a TTP 1 (ID: `TTP-0017-5822-4688`). Nesse mesmo procedimento foi identificada, como técnica secundária, a *Command and Scripting Interpreter: Windows Command Shell* (T1059.003).

A tática de *Persistence* está associada à técnica *Scheduled Task/Job: Scheduled Task* (T1053.005) e ao procedimento destacado em azul, que define a TTP 2 (ID: `TTP-0017-5822-4689`). Para a tática de *Command and Control* foram identificadas duas técnicas: *Non-Application Layer Protocol* (T1095) e *Multi-Stage Channels* (T1104), cujos procedimentos são destacados em amarelo e verde, respectivamente, originando as TTP 3 (ID: `TTP-0017-5822-4690`) e TTP 4 (ID: `TTP-0017-5822-4691`). Em cada um desses procedimentos também foi identificada a técnica secundária *Non-Standard Port* (T1571).

Quanto à sequência de execução observada no relatório, a confirmação de privilégio elevado (TTP 1) encadeia a técnica de persistência (TTP 2), garantindo a execução recorrente do arquivo `test.exe`. Ao ser executado, esse arquivo aciona, em sequência, duas técnicas de comando e controle (TTP 3 e TTP 4). As informações do mapeamento das TTPs referentes ao trecho da Figura 8.4 são:

TTP 1:

- TTP ID: TTP-0017-5822-4688
- Tactic: TA0007 (Discovery)
- Technique: T1033 (System Owner/User Discovery)
- Procedure: `The malware component, test.exe uses the Windows command "cmd.exe /C whoami" to verify it is running with the elevated privileges of "System"`
- Secondary techniques: T1059.003 (Command and Scripting Interpreter: Windows Command Shell)
- TTP Chain: TTP-0017-5822-4689

TTP 2:

- TTP ID: TTP-0017-5822-4689
- Tactic: TA0003 (Persistence)
- Technique: T1053.005 (Scheduled Task/Job: Scheduled Task)
- Procedure: `Creates persistence by creating the following scheduled task: schtasks /create /tn "mysc" /tr C:\User\Public\test.exe /sc ONLOGON /ru "System"`
- Secondary techniques:
- TTP Chain: TTP-0017-5822-4690

TTP 3:

- TTP ID: TTP-0017-5822-4690
- Tactic: TA0011 (Command and Control)
- Technique: T1095 (Non-Application Layer Protocol)
- Procedure: `When executed, the malware first establishes a SOCKS5 connection to 192.157.198.103 using TCP port 1913. The malware sends the SOCKS5 connection request "05 01 00" and verifies the server response starts with "05 00".`

- Secondary techniques: T1571 (Non-Standard Port)
- TTP Chain: TTP-0017-5822-4691

TTP 4:

- TTP ID: TTP-0017-5822-4691
- Tactic: TA0011 (Command and Control)
- Technique: T1104 (Multi-Stage Channels)
- Procedure: `The malware then requests a connection to 192.184.60.␣229 on TCP port 81 using the command "05 01 00 01 c0 b8 3c e5 00 51" and verifies that the first two bytes from the server are "05 00" (c0 b8 3c e5 is the IP address and 00 51 is the port in netwo␣rk byte order).`
- Secondary techniques: T1571 (Non-Standard Port)
- TTP Chain: `none`

Além das informações das TTPs, informações da ameaça e das referências utilizadas também são registradas neste estágio, conforme a estrutura do Diagrama de Classes da Figura 8.5.

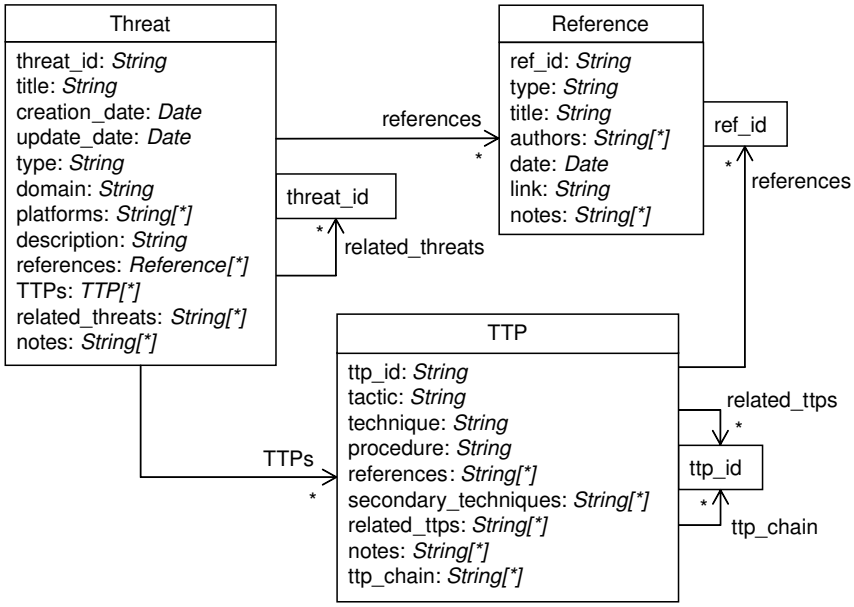


Figura 8.5: Diagrama de classes da Ameaça Mapeada em TTP

O processo de obtenção dessas informações referentes ao primeiro estágio da metodologia é apresentado, de forma simplificada, pelo pseudocódigo da Algoritmo 1.

Algoritmo 1 Pseudocódigo referente ao Estágio 1 da metodologia

```
Entrada: threat                                              ▷ Nome da ameaça
Saída: mapped_threat                                         ▷ Objeto da classe Threat
  function THREAT_TTP_MAPPING(threat)
    ▷ /* Passo 1: Obtenção de Conhecimento */                                ◁
    references ← gathering_cti(threat)
    mapped_threat.save(references)
    threat_info ← get_general_info(references)
    mapped_threat.save(title, type, domain, description, notes ← threat_info)
    cti_report ← filter_analyses(references)
    ▷ /* Passo 2: Mapeamento de TTP */                                       ◁
    tactics ← identify_tactics(cti_report)
    for TA in tactics do
        techniques ← identify_techniques(TA, cti_report)
        for T in techniques do
            P ← describe_procedure(TA, T, cti_report)
            STs ← identify_secondary_techniques(P, T, cti_report)
            new_TTP.save(tactic ← TA, technique ← T, procedure ← P,
            secondary_techniques ← STs)
            TTPs.insert(new_TTP)
    ▷ /* Passo 3: Identificação de Relacionamentos */                        ◁
    for TTP1 in TTPs do
        for TTP2 in TTPs do
            for ST in TTP1.secondary_techniques do
                if are_related(ST, TTP2.technique) then
                    TTP1.related_ttps.insert(TTP2.ttp_id)
                    TTP2.related_ttps.insert(TTP1.ttp_id)
            if execution_sequence_is(TTP1, TTP2) then
                TTP1.ttp_chain.insert(TTP2.ttp_id)
    mapped_threat.save(TTPs)
    return mapped_threat
```

8.2 Estágio 2: Mapeamento TTP-Dados

O objetivo deste estágio é desenvolver regras de detecção capazes de buscar, nos dados fornecidos pelas fontes selecionadas, evidências de comportamentos adversários associados às TTPs mapeadas no estágio anterior. A criação dessas regras é uma tarefa extremamente complexa que depende em grande parte da experiência do desenvolvedor e de sua capacidade de análise.

A dificuldade desse processo é devida majoritariamente à diferença de abstração existente entre as Técnicas adversárias, elencadas no Framework ATT&CK, e os dados brutos observáveis, como os *logs* de eventos gerados pelo Sistema Operacional. A fim de mitigar esse problema, a metodologia utiliza os Procedimentos mapeados no primeiro estágio como alicerce para a identificação do contexto de atuação adversária e análise do comportamento, reduzindo consideravelmente a abstração e, consequentemente, facilitando o mapeamento da TTP aos dados observáveis.

Este estágio é composto por uma sequência de três passos, conforme descritos abaixo:

1. **Análise do Contexto:** Este passo abrange a identificação do escopo de atuação da TTP e dos elementos que caracterizam o comportamento adversário.

2. **Mapeamento de Dados:** Neste passo são identificados, nas fontes de dados coletáveis, quais campos de quais eventos observáveis podem ser usados no mapeamento da TTP.

3. **Desenvolvimento:** Neste passo o analista elabora a regra que tem por objetivo buscar, nos dados mapeados, padrões que visam detectar a Técnica adversária.

O diagrama da Figura 8.6 ilustra o conjunto de ações que constituem o segundo estágio da metodologia. Este estágio recebe como entrada a 'Ameaça Mapeada em TTP' produzida no estágio anterior, processa as TTPs individualmente e produz Regras de Detecção ao final do processo.

As ações que constituem cada passo deste estágio são detalhadas nas próximas subseções. A subseção final apresenta o modelo de dados utilizado em cada Regra de Detecção produzida neste estágio.

8.2.1 Análise do Contexto

Neste passo, o analista deve examinar o contexto de atuação do atacante na execução da ameaça para delimitar a busca por elementos que ajudem a identificar os comportamentos associados às TTPs mapeadas. Para alcançar esse objetivo, o analista deve executar as seguintes ações:

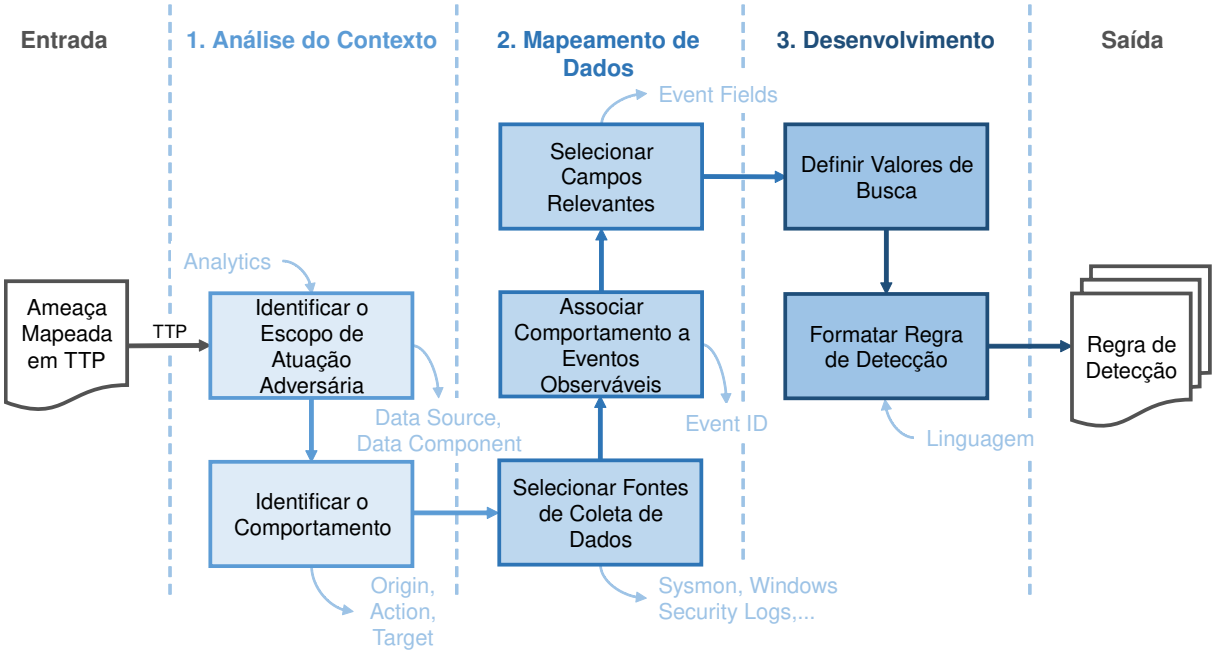


Figura 8.6: Fluxo de ações do 2º estágio: 'Mapeamento TTP-Dados'

1. Identificar o Escopo de Atuação Adversária
2. Identificar o Comportamento

1. Identificar o Escopo de Atuação Adversária

Nesta ação o analista deve, com base no Procedimento da TTP identificado na fase anterior, determinar o escopo das ações maliciosas empregadas na Técnica utilizada. O objetivo é delimitar a busca por dados que caracterizem o comportamento adversário.

A identificação do escopo pode ser realizada com auxílio do framework ATT&CK, por meio do *Data Source* [158] e, mais especificamente, do *Data Component* [159] que melhor corresponde à TTP analisada.

Os *Data Sources*[9] representam diferentes tópicos de informações que podem ser fornecidas por sensores ou *logs*, e os *Data Components*[10] restringem ainda mais o escopo das informações fornecidas pelo *Data Source*.

A fim de melhor identificar a atuação adversária, o analista deve verificar, com base no framework ATT&CK [18], as *Data Sources* e *Data Components* que melhor correspondem à TTP analisada neste estágio. A identificação precisa dessas informações permitirá delimitar o escopo de busca de dados úteis para o desenvolvimento da regra de detecção da Técnica desejada.

[9] `https://attack.mitre.org/datasources/`
[10] `https://attack.mitre.org/datacomponents/`

2. Identificar o Comportamento

Esta ação tem como objetivo identificar o comportamento adversário referente à implementação da Técnica analisada, oferecendo um nível de detalhamento maior da atuação adversária e facilitando o mapeamento dos dados necessários à detecção da TTP.

O comportamento (em inglês, *behavior*) pode ser caracterizado por três componentes: uma **ação**, uma **origem** e um **alvo**. A origem é a componente que inicia a execução da ação, e o alvo é a componente afetada por ela.

Essas componentes oferecem um contexto mais preciso sobre o comportamento a ser detectado, auxiliando na definição de quais eventos, em quais fontes de dados, devem ser monitorados para possibilitar a detecção eficaz das ações do adversário.

Para identificar essas componentes, o analista deve compreender a execução do procedimento descrito na TTP, com base nas informações de inteligência tática ou em testes em laboratório, a fim de reconhecer a principal ação e os demais elementos que a compõem, dentro do escopo de atuação daquela técnica.

O comportamento pode ser expresso pela tripla *(origem, ação, alvo)*, em inglês *(origin, action, target)*, ou diretamente em forma de frase, como por exemplo: processo cria processo; usuário deleta arquivo; processo requisita acesso a arquivo; usuário executa comando; etc.

Tabela 8.1: Exemplos de comportamentos associados a *Data Components*

Data Source	**Data Component**	**Comportamento**		
		Origem	**Ação**	**Alvo**
File	File Access	Process	access	file
		Process	request access to	file
		User	access	file
		User	request access to	file
	File Creation	Process	create	file
		User	create	file
	File Deletion	Process	delete	file
		User	delete	file
	File Modification	Process	modify	file
		User	modify	file
Service	Service Access	User	request access to	service
	Service Creation	User	create	service
	Service metadata	Service	started	–
		Service	stopped	–

Haja vista a grande utilidade de se usar componentes para caracterizar comportamentos, esse artifício é utilizado também por outras iniciativas de segurança, como o projeto OSSEM (Open Source Security Events Metadata) [191] e o MITRE CAR (Cyber Analy-

tics Repository) [21]. O OSSEM utiliza as componentes *source*, *relationship* e *target*, enquanto o MITRE CAR usa *object* e *action* para mapear dados relevantes.

8.2.2 Mapeamento de Dados

A interferência adversária no sistema-alvo deixa rastros que, normalmente, são registrados nos *logs* do sistema ou da aplicação. Portanto, para detectar um comportamento específico, é necessário, primeiramente, mapear quais dados observáveis devem ser coletados e analisados.

No mapeamento a ser realizado neste passo da metodologia, é fundamental considerar as informações do escopo de atuação identificadas no passo anterior. O analista deve, então, determinar quais eventos, de quais fontes, devem ser observados e quais campos devem ser analisados para que seja possível identificar as características comportamentais relacionadas à TTP em questão. Para isso, devem ser executadas as seguintes ações:

1. Selecionar Fontes de Coleta de Dados
2. Associar Comportamento a Eventos Observáveis
3. Selecionar Campos Relevantes

1. Selecionar Fontes de Coleta de Dados

Fonte de dados pode ser definida como sendo uma "fonte de informações coletadas por um sensor ou sistema de *logs* que pode ser usada para obter informações relevantes para identificar a ação que está sendo executada por um adversário, a sequência de ações ou os resultados dessas ações" [157].

Portanto, nesta etapa o analista deve escolher fontes que forneçam dados de eventos de segurança e *logs* de sistema/aplicações que possibilitem a observação do comportamento identificado no passo anterior.

O sistema Windows, por exemplo, possui nativamente várias fontes de eventos que são agrupadas em provedores de *logs*, como: *Microsoft-Windows-Security-Auditing*, *Microsoft-Windows-PowerShell*, etc. A lista completa pode ser obtida com o comando '`wevtutil enum-logs`' [192].

Há também fontes não nativas que podem ser instaladas para expandir a capacidade de monitoramento do sistema, como o Sysmon [5] e o OSQuery [193], que também são suportados por sistemas operacionais Linux.

O Sysmon possui foco em eventos relacionados à segurança e é fornecido pela Microsoft como uma ferramenta utilitária do sistema, devendo ser instalado de forma independente. Por fornecer informações mais detalhadas que auxiliam na contextualização e correlação

com outros eventos, o Sysmon tem sido amplamente utilizado como fonte de dados para o Threat Hunting, oferecendo evidências mais concretas de eventos maliciosos e facilitando a detecção de ameaças. Sua versão 15.15 [5] possui um total de 30 tipos de eventos, conforme descritos na Tabela 8.2.

2. Associar Comportamento a Eventos Observáveis

Nesta ação, os analistas devem selecionar eventos que possam conter, em seus dados, informações relevantes para detectar a atuação da TTP analisada. Os eventos selecionados devem obrigatoriamente pertencer às fontes escolhidas na etapa anterior, mesmo que existam outras opções de outras fontes.

No processo de escolha dos eventos, o analista deve observar o contexto de atuação da TTP, identificado no passo anterior, e utilizá-lo como guia na seleção dos eventos mais adequados fornecidos pelas fontes escolhidas. O *Data Component* [159], juntamente com as componentes do comportamento *(origem, ação, alvo)*, determinam a finalidade dos eventos, servindo como filtro na busca pelos mais adequados conforme a documentação dos provedores de *logs*.

As Tabelas 8.3 e 8.4 mostram, respectivamente, exemplos de eventos do *Windows Security Auditing* e do *Sysmon* que contêm informações pertinentes a alguns comportamentos e que podem ser associados conforme o escopo de atuação da Técnica adversária.

O projeto OSSEM (Open Source Security Events Metadata) [191] é uma iniciativa aberta da comunidade de cibersegurança que se concentra na documentação e padronização de *logs* de eventos de segurança de diversas fontes de dados e sistemas operacionais. É possível encontrar em seu repositório no GitHub[11] arquivos em formato YAML contendo associações entre eventos de segurança e informações de *Data Sources/Data Components* do Framework ATT&CK.

3. Selecionar Campos Relevantes

Esta ação tem como objetivo determinar os campos, dos eventos selecionados na etapa anterior, capazes de conter informações úteis para a detecção do comportamento desejado. Consultas à documentação das fontes de dados são recomendadas para o conhecimento não só da estrutura dos eventos selecionados, mas também dos tipos de informações fornecidas em cada campo.

O analista deve, então, selecionar os campos relevantes visando o desenvolvimento da regra de busca em conformidade com uma estratégia de detecção.

[11] `https://github.com/OTRF/OSSEM-DM`

Tabela 8.2: Eventos do Sysmon v15.15 [5]

ID	Evento	Descrição
1	Process creation	Fornece informações estendidas sobre um processo recém-criado.
2	File creation time changed	Registra alterações no horário de criação de arquivos.
3	Network connection	Registra conexões TCP/UDP na máquina.
4	Sysmon service state changed	Indica mudanças no estado do serviço Sysmon.
5	Process terminated	Registra o término de processos.
6	Driver loaded	Fornece informações sobre o carregamento de drivers no sistema.
7	Image loaded	Registra o carregamento de módulo em um processo específico.
8	CreateRemoteThread	Detecta a criação de threads remotas em um outro processo.
9	RawAccessRead	Detecta operações de leitura da unidade usando a denotação '\\.\'.
10	ProcessAccess	Registra quando um processo abre outro processo.
11	FileCreate	Registra a criação ou sobrescrita de arquivos.
12	RegistryEvent (Object create and delete)	Registra criação ou exclusão de chaves e valores no Registro do Windows.
13	RegistryEvent (Value set)	Registra a modificação de valores no Registro.
14	RegistryEvent (Key and Value rename)	Registra a renomeação de chaves e valores no Registro do Windows.
15	FileCreateStreamHash	Registra a criação de fluxos de arquivo.
16	ServiceConfigurationChange	Registra alterações na configuração do Sysmon.
17	PipeEvent (Pipe created)	Registra a criação de *pipes* nomeados.
18	PipeEvent (Pipe connected)	Registra conexões de *pipes* nomeados.
19	WmiEventFilter	Indica o registro de filtro de eventos WMI.
20	WmiEventConsumer	Indica o registro de consumidores WMI.
21	WmiEventConsumerToFilter	Registra associações entre consumidores e filtros de eventos WMI.
22	DNSEvent	Registra consulta de DNS.
23	FileDelete	Registra a exclusão de arquivos.
24	ClipboardChange	Registra alteração de conteúdo do clipboard.
25	ProcessTampering	Detecta técnicas de ocultação de processos .
26	FileDeleteDetected	Registra a exclusão de arquivo.
27	FileBlockExecutable	Indica quando o Sysmon detecta e bloqueia a criação de arquivos executáveis (formato PE).
28	FileBlockShredding	Indica quando o Sysmon detecta e bloqueia a destruição de arquivos.
29	FileExecutableDetceted	Indica quando o Sysmon detecta a criação de um novo arquivo executável (formato PE).
255	Error	Indica que houve erro no Sysmon.

Tabela 8.3: Exemplo eventos relevantes do *Windows Security Auditing* para identificação de alguns comportamentos

Data Components	**Comportamento**			**Evento**	**Nome do Evento**
	Origem	**Ação**	**Alvo**	**(ID)**	
File Access	Process/User	access	file	**4663**	An attempt was made to access an object
	Process/User	request access to	file	**4656**	A handle to an object was requested
	User	request access to	file	**4661**	A handle to an object was requested
File Deletion	Process/User	delete	file	**4660**	An object was deleted
File Modification	Process/User	modify	file	**4670**	Permissions on an object were changed
Service Access	User	request access to	service	**4656**	A handle to an object was requested
Service Creation	User	create	service	**4697**	A service was installed in the system
	User	create	service	**7045**	A new service was installed in the system
Process Access	Process/User	access	process	**4663**	An attempt was made to access an object
	Process	request access to	process	**4656**	A handle to an object was requested
Process Creation	Process/User	create	process	**4688**	A new process has been created
Process Termination	User	terminate	process	**4689**	A process has exited

Tabela 8.4: Exemplo de eventos relevantes do *Sysmon* para identificação de alguns comportamentos

Data Components	**Comportamento**			Evento	**Nome do Evento**
	Origem	**Ação**	**Alvo**	(ID)	
File Creation	Process	create	file	**11**	FileCreate
File Deletion	Process/User	delete	file	**23**	File Delete archived
	Process/User	delete	file	**26**	File Delete logged
File Modification	Process	modify	file	**2**	A process changed a file creation time
	Process	modify	file	**11**	FileCreate
Service metadata	Service	started/stopped	–	**4**	Sysmon service state changed
Process Access	Process	access	process	**10**	ProcessAccess
	Process	request access to	process	**10**	ProcessAccess
Process Creation	Process/User	create	process	**1**	Process Creation
	Process	create	thread	**8**	CreateRemoteThread
Process Termination	User	terminate	process	**5**	Process terminated

A Tabela 8.5 apresenta os campos do evento 13 do *Sysmon*, RegistryEvent – Value Set, conforme a documentação [5, 6]. Exemplos de campos relevantes para identificar modificação no registro seriam `TargetObject` e `Details`, haja vista que `TargetObject` armazena a chave do registro modificada e `Details` contém o valor adicionado a ela.

Tabela 8.5: Campos do evento 13 (RegistryEvent – Value Set) do *Sysmon* [6]

Campo	**Descrição**
UtcTime	Hora em UTC em que o evento foi criado.
EventType	SetValue
ProcessGuid	GUID do processo que modificou um valor de registro.
ProcessId	ID do processo usado pelo sistema operacional para identificar o processo que modificou um valor de registro.
Image	Caminho do arquivo do processo que modificou um valor do registro.
TargetObject	Caminho completo da chave de registro modificada.
Details	Detalhes adicionados à chave de registro.
User	Nome da conta que acessou o registro. Geralmente contém nome de domínio e nome de usuário.

8.2.3 Desenvolvimento

Este passo tem por objetivo desenvolver uma regra capaz de buscar, nos dados coletados, informações que caracterizem a Técnica implementada. É o passo mais complexo da metodologia e o que demanda maior expertise do analista. Para atingir esse objetivo, o analista deve executar as seguintes ações:

1. Definir Valores de Busca
2. Formatar Regra de Detecção

1. Definir Valores de Busca

Esta ação tem por objetivo definir quais dados ou padrões são necessários buscar nos campos dos eventos coletados, a fim de identificar o comportamento referente à TTP analisada.

A definição desses valores de busca varia conforme a estratégia adotada e depende, sobretudo, do grau de conhecimento obtido da inteligência tática referente àquela ameaça, principalmente no que se refere à compreensão de como o atacante executa as técnicas e dos possíveis efeitos nos alvos.

A análise dos procedimentos identificados no mapeamento das TTPs é fundamental neste processo, juntamente com o conhecimento aprofundado do sistema-alvo. A escolha

dos valores de busca, quando possível, deve evitar dados muito genéricos, para minimizar falsos positivos, e dados muito específicos ou dependentes de versões do Sistema Operacional, a fim de minimizar falsos negativos.

2. Formatar Regra de Detecção

Esta ação tem por objetivo desenvolver, a partir das informações de eventos, campos e valores definidos nas etapas anteriores, uma lógica de busca capaz de identificar, nos dados coletados, uma característica da atuação adversária referente a uma TTP da ameaça analisada.

Em seguida, a lógica de busca deve ser escrita em formato de ***query***, utilizando-se uma sintaxe específica de acordo com a linguagem escolhida. Exemplos de linguagens comuns utilizadas em Threat Hunting são:

- Kibana Query Language (KQL) [194];
- Elasticsearch Query Language (ES|QL) [195];
- Spark SQL [196];
- Sigma [197]; e
- Splunk Search Processing Language (SPL) [198].

A escolha da linguagem deve considerar não apenas os recursos disponíveis para suportar a lógica de busca desenvolvida, como também a compatibilidade com as plataformas e ferramentas de Threat Hunting que se pretende utilizar.

A *query* desenvolvida é o núcleo da Regra de Detecção e, por isso, é comum que os termos *query* e *regra de detecção* sejam utilizados de forma intercambiável. No entanto, conforme o modelo de dados definido no Capítulo 7, a Regra de Detecção (iniciando com letras maiúsculas) é uma entidade estruturada que contém diversas informações adicionais além da *query*.

8.2.4 Regra de Detecção

Para ilustrar o processo de desenvolvimento de regras de detecção proposto neste estágio da metodologia, considere a TTP 1 mapeada no exemplo apresentado na Subseção 8.1.4, referente à campanha de ciberespionagem "Operation Double Tap" [4]. A análise do procedimento descrito nessa TTP evidencia a necessidade de desenvolver uma regra capaz de identificar a execução do comando `whoami` pelo programa suspeito `test.exe`, com o objetivo de detectar a técnica empregada pelo adversário para verificar se o processo está sendo executado com privilégios elevados.

A Figura 8.7 apresenta, de forma integrada, um exemplo das informações produzidas em cada etapa do segundo estágio da metodologia, evidenciando como elas contribuem para a construção da regra de detecção associada à TTP 1 do exemplo em questão. Embora se trate de um cenário simples, o exemplo demonstra o potencial da metodologia em orientar o analista de maneira sistemática na identificação dos dados relevantes para a formulação da lógica de detecção, reduzindo a complexidade do processo de desenvolvimento de regras baseadas em TTPs.

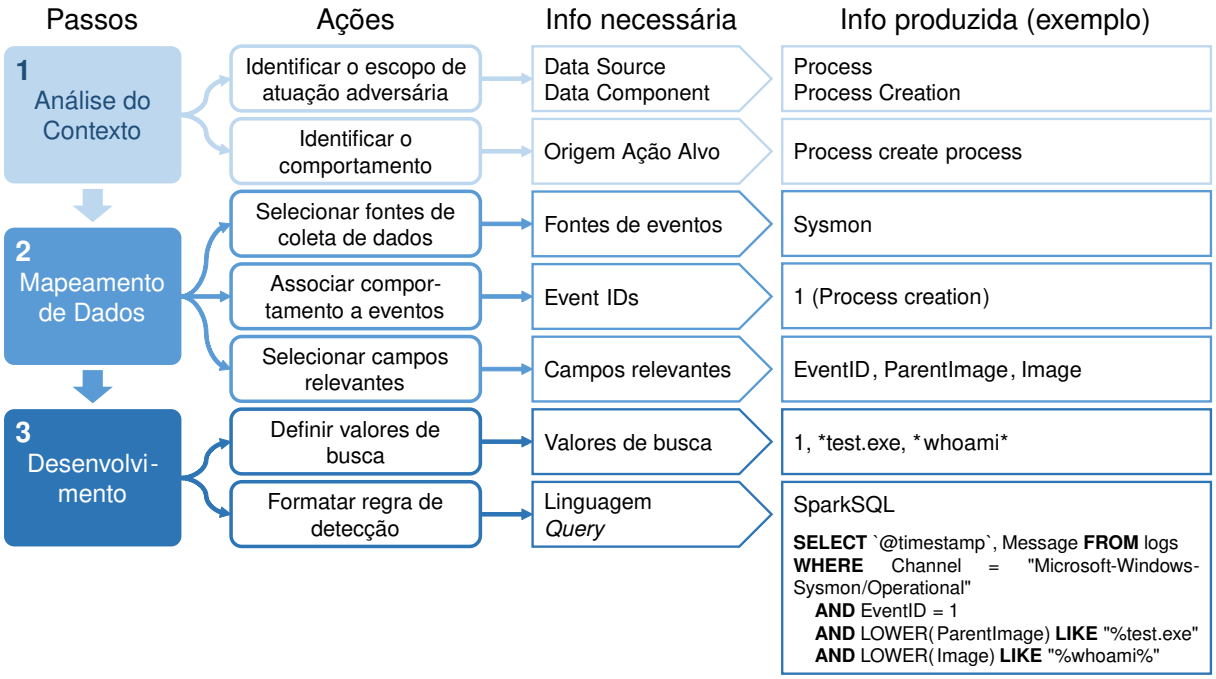


Figura 8.7: Exemplo do processo de desenvolvimento da *query* de uma regra de detecção usando o Estágio 2 da metodologia

O produto deste estágio é um objeto denominado Regra de Detecção, que encapsula diversos dados relevantes para a atividade de Threat Hunting, sendo o principal a *query* responsável pela detecção de uma ou mais técnicas que compõem a ameaça analisada. As informações que integram a Regra de Detecção correspondem à classe `DetectionRule`, definida no modelo de dados apresentado no Capítulo 7, e são posteriormente utilizadas na composição do Runbook de Threat Hunting.

O Algoritmo 2 apresenta o pseudocódigo referente à execução deste estágio da metodologia, enquanto a Figura 8.8 ilustra o Diagrama de Classes que representa a estrutura da Regra de Detecção produzida.

Algoritmo 2 Pseudocódigo referente ao Estágio 2 da metodologia

```
Entrada: TTP                                              ▷ Objeto da classe TTP
Saída: detection_rule                                     ▷ Objeto da classe DetectionRule
  function TTP_DATA_MAPPING(TTP)
    ▷ /* Passo 1: Análise de Contexto */ ◁
    repeat
      DS ← get_attck_DataSource(TTP)
      DC ← get_attck_DataComponent(TTP, DS)
      (origin, action, target) ← identify_behavior(TTP, DC)
      behaviors.insert(origin, action, target)
    until context_analysis_done
    detection_rule.save(reference_ttp ← TTP.ttp_id)
    ▷ /* Passo 2: Mapeamento de Dados */ ◁
    platforms ← define_platforms()
    sources ← select_data_collection_sources(platforms, behaviors)
    detection_rule.save(platforms, sources)
    events ← select_relevant_events(sources, behaviors)
    for ev in events do
      fields[ev] ← select_relevant_fields(TTP, ev)
    ▷ /* Passo 3: Desenvolvimento */ ◁
    for (ev, fd) in events, fields[ev] do
      values[ev, fd] ← define_search_value(ev, fd, TTP)
    language ← define_query_language()
    query ← format_detection_rule(language, fields, values)
    detection_rule.save(language, query)
    return detection_rule
```

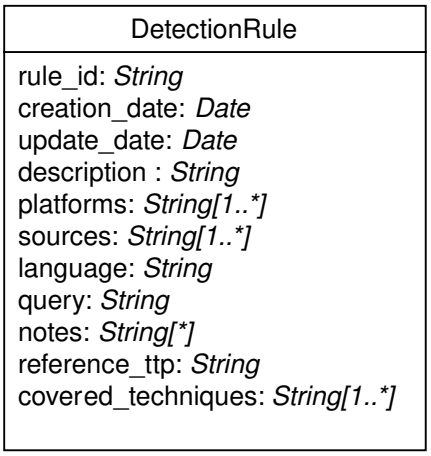


Figura 8.8: Diagrama de classes da Regra de Detecção

8.3 Estágio 3: Validação

Neste estágio, as Regras de Detecção propostas são testadas em dados reais ou simulados, com o objetivo de avaliar suas capacidades individuais de identificar o comportamento adversário correspondente à TTP analisada. Para isso, a metodologia divide este estágio em três passos, conforme descritos abaixo:

1. **Planejamento:** Este passo envolve o planejamento das emulações e a preparação do ambiente necessário aos testes.

2. **Execução:** Este passo engloba a emulação da ameaça para coleta de dados e os testes das regras desenvolvidas.

3. **Avaliação:** Este passo engloba a validação ou rejeição das regras, de acordo com os resultados obtidos nos testes, e a verificação dos encadeamentos mapeados.

O diagrama da Figura 8.9 ilustra o conjunto de ações que constituem o terceiro estágio da metodologia. As regras que passam no teste são validadas, enquanto as que falham retornam para o estágio anterior para novo desenvolvimento. No caso da verificação da cadeia de TTPs, se for identificado um encadeamento incorreto, o processo retorna ao primeiro estágio para as devidas correções.

As ações que constituem cada passo deste estágio são detalhadas nas próximas subseções. A subseção final apresenta o modelo de dados utilizado no registro da Regra de Detecção validada.

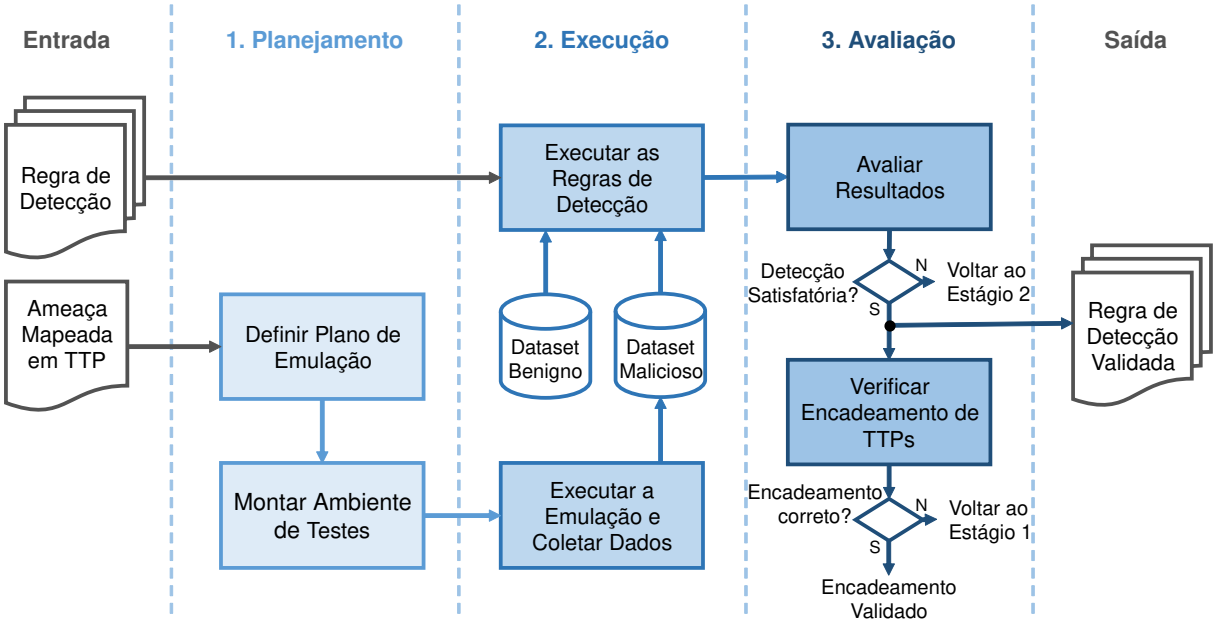


Figura 8.9: Fluxo de ações do 3º estágio: 'Validação'

8.3.1 Planejamento

Neste passo, os analistas devem planejar a emulação da ameaça e preparar as condições para a coleta dos dados necessários à realização dos testes das Regras de Detecção desenvolvidas no estágio anterior. Para isso, devem ser realizadas as seguintes ações:

1. Definir Plano de Emulação
2. Montar Ambiente de Testes

1. Definir Plano de Emulação

Esta ação tem por objetivo desenvolver um plano que descreva detalhadamente o cenário de ataque correspondente à ameaça analisada, especificando como reproduzi-lo em um ambiente controlado. Isso inclui definir quais TTPs mapeadas serão emuladas, os recursos necessários e os procedimentos a serem seguidos durante a execução.

O plano de emulação deve procurar imitar os comportamentos observados do ator da ameaça, baseando-se nos Procedimentos que implementam as Técnicas e Táticas mapeadas. O fluxo operacional da emulação deve ser definido com base nas mesmas referências utilizadas para o mapeamento das TTPs. No caso da execução de *software* ou amostras de *malware*, deve-se preferir a mesma versão ou variante informada nos *reports* de CTI analisados, para que a progressão sequencial do ataque emulado seja o mais fiel possível à ameaça real, buscando-se atingir os mesmos objetivos operacionais do atacante no ambiente da vítima.

Repositórios de planos de emulação que podem ser utilizados como referência, bem como fontes abertas com instruções de simulação de atividades adversárias, incluem:

- Adversary Emulation Library [199, 200];
- APT Attack Simulation [201];
- Atomic Red Team [20].

Após definido, o plano de emulação pode ser registrado como referência ou, em casos de emulações mais simples, como anotações (`notes`) da classe `Validation` do modelo de dados.

2. Montar Ambiente de Testes

Esta ação tem por objetivo implantar um ambiente controlado que permita emular a ameaça conforme estabelecido no plano de emulação, coletar os *logs* fornecidos pelas fontes selecionadas e testar as Regras de Detecção propostas. É fundamental que o ambiente contenha uma plataforma de Threat Hunting que suporte as linguagens das *queries* que precisam ser avaliadas.

A Figura 8.10 apresenta uma estrutura de laboratório de Threat Hunting para coleta de dados e testes de detecção utilizando ferramentas *open-source*. Essa solução suporta regras nas linguagens Lucene, KQL, EQL, ES|QL, SparkSQL e Query DSL [202]. Os principais componentes são descritos a seguir:

- **Elasticsearch** [203]: motor de busca e análise distribuído utilizado para armazenar e indexar grandes volumes de dados de *logs* coletados, permitindo consultas rápidas e eficientes;
- **Kibana** [204]: interface *web* para visualização e análise de dados armazenados no Elasticsearch, que permite consultas e criação de *dashboards* interativos;
- **Logstash** [205]: ferramenta de coleta, transformação e ingestão de dados, responsável por processar os *logs* e enviá-los ao Elasticsearch de forma estruturada;
- **Beats** [206]: conjunto de agentes leves instalados nas máquinas monitoradas para coleta de dados, cujos eventos são encaminhados ao Kafka;
- **Kafka** [207]: plataforma de *streaming* distribuída usada como intermediária entre os Beats e o Logstash, garantindo alta performance e resiliência;
- **Jupyter Notebook** [208]: ambiente interativo de análise de dados, usado para executar consultas e desenvolver *scripts* analíticos;
- **Apache Spark** [209]: *framework* de processamento distribuído em larga escala, empregado em análises correlacionais e detecção comportamental;

- **ES-Hadoop** [210]: conector que integra o Elasticsearch ao ecossistema Hadoop/Spark, permitindo leitura e escrita de dados diretamente em *jobs* Spark.

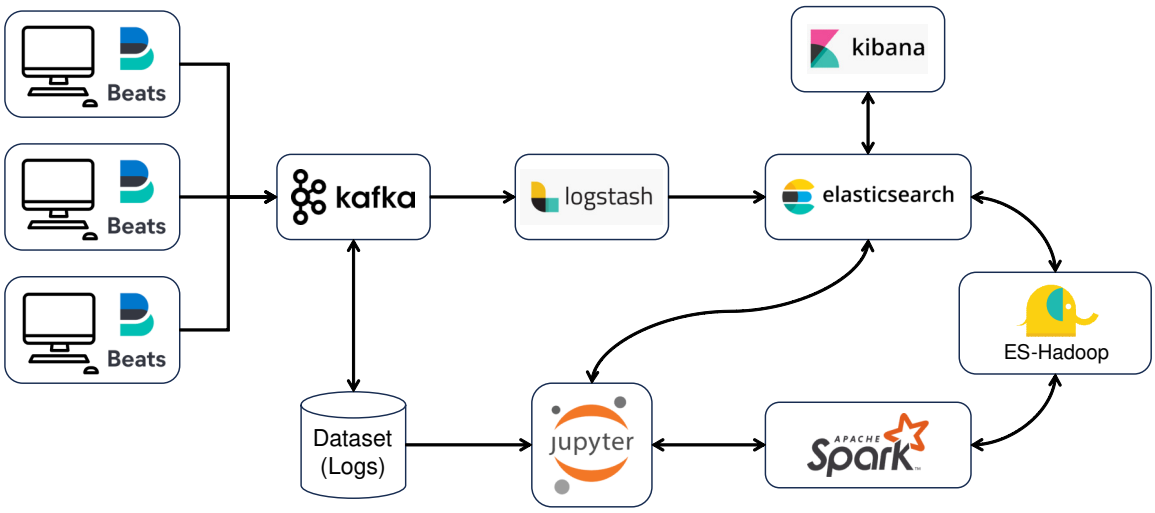


Figura 8.10: Exemplo de estrutura de laboratório de Threat Hunting

Existem também soluções prontas que atendem aos requisitos desta ação. Entre as mais conhecidas *open-source*, destacam-se:

- Attack Range [211];
- HELK [212];
- Detection Lab [213].

8.3.2 Execução

Este passo tem por objetivo executar os testes das Regras de Detecção nos dados coletados da ameaça emulada. Para isso, devem ser realizadas as seguintes ações:

1. Executar a Emulação e Coletar Dados;
2. Executar as Regras de Detecção.

1. Executar a Emulação e Coletar Dados

Esta ação consiste na execução controlada da ameaça, conforme descrito no plano de emulação. Durante o processo, é essencial assegurar que todos os eventos relevantes sejam devidamente gerados e coletados, considerando as fontes de dados previamente identificadas para cada TTP.

Os dados coletados podem ser registrados diretamente no banco de dados da plataforma de Threat Hunting ou armazenados em datasets estruturados que viabilizam

múltiplas execuções e análises offline. Exemplos comuns de formatos incluem JSON, CSV e Parquet. O uso de datasets oferece diversas vantagens, tais como:

- **Isolamento do ambiente de execução:** possibilita a condução dos testes fora da plataforma principal, sem interferência de outras operações ou ingestões de dados.
- **Versionamento e reprocessamento simplificados:** viabilizam análises comparativas entre diferentes variantes de ameaça ou entre distintos cenários de ataque.
- **Integração facilitada com ferramentas externas:** como *notebooks* Jupyter [208], *scripts* analíticos em Python, motores Spark [209] ou bancos alternativos de busca e análise.
- **Execução distribuída e paralela:** permite a realização de testes em múltiplos ambientes e sistemas sem a necessidade de repetir a emulação.
- **Compartilhamento facilitado:** possibilita a distribuição dos datasets entre diferentes equipes ou instituições para revisão, validação cruzada ou replicação de resultados, mesmo sem acesso ao ambiente original.

Considerando a complexidade de algumas ameaças e o grande volume de dados gerado, recomenda-se que a coleta seja realizada de forma segmentada, organizando os registros por fases do ataque ou por TTPs. Essa segmentação reduz a carga computacional durante a execução dos testes, especialmente em ambientes com recursos limitados. Contudo, é fundamental que os eventos maliciosos associados a uma mesma TTP permaneçam consolidados em um único dataset, uma vez que a fragmentação indevida pode comprometer a detecção e gerar falsos negativos.

Outra questão importante é garantir que a estrutura dos datasets maliciosos gerados esteja devidamente formatada e compatível com a plataforma utilizada para a execução das regras. Deve-se evitar o uso de campos aninhados que dificultem o *parsing* ou provoquem erros de leitura. Além disso, divergências na nomenclatura dos campos ou inconsistências na estrutura dos eventos podem inviabilizar a execução das regras ou produzir resultados incorretos, comprometendo a avaliação da metodologia.

O nome ou link do dataset malicioso utilizado na validação de cada regra deve ser registrado no campo `dataset` da classe `Validation`, além de ser incluído na lista de referências.

2. Executar as Regras de Detecção

Esta ação tem como objetivo testar as Regras de Detecção desenvolvidas. Esse processo envolve a execução de cada regra em dois conjuntos de dados:

- **Dataset malicioso**, contendo dados coletados a partir da emulação da ameaça; e

- **Dataset benigno**, contendo dados coletados em ambiente livre de atividade maliciosa.

Para isso, é necessário importar ambos os datasets na plataforma de testes e executar as *queries* associadas a cada regra. Cada consulta deve ser aplicada individualmente nos dois datasets, seguida da análise dos resultados obtidos.

Dependendo do tamanho dos datasets, o intervalo temporal da busca pode ser reduzido para fins de otimização. No entanto, no caso do dataset malicioso, deve-se garantir que a janela de tempo utilizada englobe integralmente os dados referentes à emulação da TTP que se deseja detectar.

Quanto ao dataset malicioso utilizado, ele pode ser próprio — construído com base no plano de emulação definido no passo anterior da metodologia — ou proveniente de terceiros. Exemplos de iniciativas que disponibilizam datasets gratuitos de diversos ataques incluem o *Security Datasets* [214] e o *Attack Data* [215]. Embora esses datasets geralmente representem ameaças simples, envolvendo poucas técnicas, eles podem ser úteis para validar regras de detecção associadas a TTPs amplamente difundidas.

A utilização de datasets maliciosos prontos constitui uma forma eficaz de agilizar o processo de validação, pois permite omitir as três primeiras ações deste estágio. Contudo, é essencial garantir que a ameaça emulada corresponda à considerada pela metodologia e que o dataset malicioso contenha dados provenientes das fontes selecionadas. Ademais, é indispensável ajustar o formato, a estrutura e a nomenclatura dos campos, a fim de evitar incompatibilidades durante a execução das regras.

8.3.3 Avaliação

Este passo tem por objetivo analisar os resultados obtidos nos testes, avaliando a eficácia das Regras de Detecção e verificando a sequência de ações detectadas em conformidade com o encadeamento das TTPs. Ele compreende as seguintes ações:

1. Avaliar Resultados;
2. Verificar Encadeamento de TTPs.

1. Avaliar Resultados

Nesta ação, o analista deve examinar os resultados da execução das Regras de Detecção separadamente em cada dataset selecionado. Esse processo envolve tanto análises quantitativas quanto qualitativas:

- **Análise Quantitativa:** consiste em avaliar a quantidade de respostas obtidas (*hits*). No caso do dataset malicioso, se a regra não produzir qualquer resultado

— *i.e.*, zero *hits* — deve-se, antes de considerá-la inválida, verificar se o problema decorre de falhas na coleta, divergências de campos ou erros estruturais no dataset. Se houver número excessivo de resultados, recomenda-se revisar o dataset quanto à redundância na coleta de eventos, bem como reavaliar a precisão dos critérios definidos na *query*.

- **Análise Qualitativa:** consiste em confirmar se os resultados correspondem efetivamente ao comportamento que se deseja observar, identificando possíveis falsos positivos e verificando se as evidências retornadas são coerentes com a TTP mapeada.

Os resultados obtidos a partir do dataset benigno são igualmente relevantes, pois permitem avaliar a suscetibilidade da regra à geração de falsos positivos. Idealmente, as amostras benignas não devem ser detectadas — ou seja, o resultado esperado é zero *hits*. Contudo, em razão do caráter mais abrangente das regras baseadas em TTPs, o analista deve considerar o contexto operacional e definir, caso a caso, o limite aceitável de detecções benignas.

A validação da regra depende simultaneamente de dois fatores:

- Detecção positiva do comportamento desejado no dataset malicioso; e
- Ausência de detecções — ou detecções dentro do limite aceitável definido pelo analista — no dataset benigno.

Regras que não apresentarem resultados satisfatórios em qualquer uma dessas condições devem ser retornadas ao segundo estágio da metodologia para reformulação. Uma Regra de Detecção é considerada validada quando identifica corretamente apenas o comportamento desejado, em conformidade com a TTP de referência.

2. Verificar Encadeamento de TTPs

Esta ação tem como objetivo confirmar a coerência do encadeamento das TTPs detectadas, com base nos *timestamps* dos eventos identificados pelas Regras de Detecção validadas durante a análise dos testes no dataset malicioso. Inconsistências na sequência observada podem indicar divergências entre o comportamento descrito nas fontes de CTI e aquele realmente reproduzido na emulação. Entre as possíveis causas, destacam-se:

- Plano de emulação impreciso; ou
- Erros na definição dos encadeamentos entre as TTPs.

No primeiro caso, o analista deve revisar o plano de emulação, produzir um novo dataset malicioso e repetir os testes de validação. No segundo, é necessário retornar ao primeiro estágio da metodologia para reavaliar o mapeamento e a ordem de ocorrência das TTPs.

Ameaças mais complexas podem apresentar fluxos de execução variáveis, condicionados a fatores específicos do alvo — como mecanismos de defesa habilitados, privilégios do usuário, softwares instalados ou particularidades de configuração do sistema. Esses fatores podem gerar ramificações no encadeamento das TTPs, tornando necessária a realização de emulações e testes adicionais com diferentes cenários de ataque, a fim de verificar os encadeamentos em todos os ramos possíveis de execução.

8.3.4 Regra de Detecção Validada

O produto final deste estágio é a Regra de Detecção enriquecida com os dados de validação, conforme a classe `Validation` do modelo de dados apresentado no Capítulo 7.

O Algoritmo 3 apresenta o pseudocódigo referente à execução deste estágio, e a Figura 8.11 ilustra o diagrama de classes da Regra de Detecção validada.

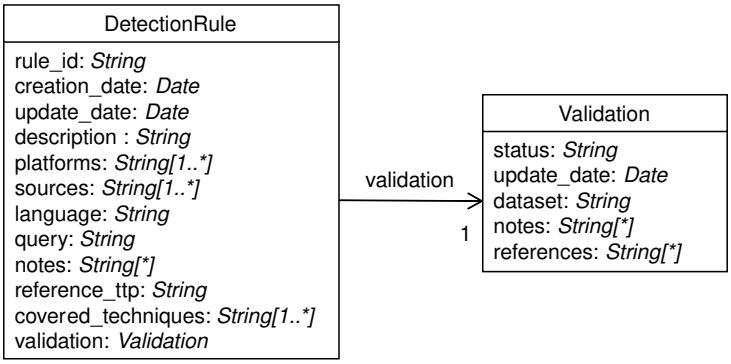


Figura 8.11: Diagrama de classes da Regra de Detecção Validada

8.4 Estágio 4: Consolidação dos Resultados

Este estágio tem como objetivo consolidar as informações produzidas nos estágios anteriores e, com base no modelo de dados proposto, registrá-las em um documento único denominado *Runbook de Threat Hunting*. O Runbook resultante sintetiza o conhecimento técnico obtido ao longo do processo, fornecendo inteligência tática acionável em um formato de serialização simples, padronizado e legível tanto por humanos quanto por máquinas.

O diagrama apresentado na Figura 8.12 ilustra a sequência de ações que compõem este estágio. Por se tratar de uma etapa menos complexa em relação às anteriores, suas ações são organizadas em um único passo denominado 'Criação do Runbook', descrito a seguir.

Algoritmo 3 Pseudocódigo referente ao Estágio 3 da metodologia

```
Entrada: mapped_threat; detection_rules
Saída: validated_detection_rules                    ▷ Lista de objetos DetectionRule
  function VALIDATE(mapped_threat, detection_rules)
    ▷ /* Passo 1: Planejamento */ ◁
    emulation_plan ← define_emul_plan(mapped_threat)
    ▷ /* Passo 2: Execução */ ◁
    dataset ← emulate_threat(emulation_plan)
    for DR in detection_rules do
      response_TP ← run(DR.query, malicious_dataset)
      response_FP ← run(DR.query, benign_dataset)
      ▷ /* Passo 3: Avaliação */ ◁
      hits_TP ← count_true_positive(response_TP)
      hits_FP ← count_false_positive(response_FP)
      FP_limit ← define_FP_limit(DR, response_FP)
      if hits_TP ≥ 1 AND hits_FP ≤ FP_limit then
        DR.validation.status ← "validated"
        detection_timestamp[DR.reference_ttp] ←
        get_timestamp(response_TP)
      else
        DR.validation.status ← "failed"
        send_back_to_stage_2(DR)
      DR.save(validation)
      validated_detection_rules.insert(DR)
    ▷ /* Verificação do encadeamento de TTPs */ ◁
    for TTP in mapped_threat.TTPs do
      ttp_id_1 ← TTP.ttp_id
      for ttp_id_2 in TTP.ttp_chain do
        if detection_timestamp[ttp_id_2] < detection_timestamp[ttp_id_1]
        then
          print("Encadeamento de TTP incorreto")
          send_back_to_stage_1(TTP)
    return validated_detection_rules
```

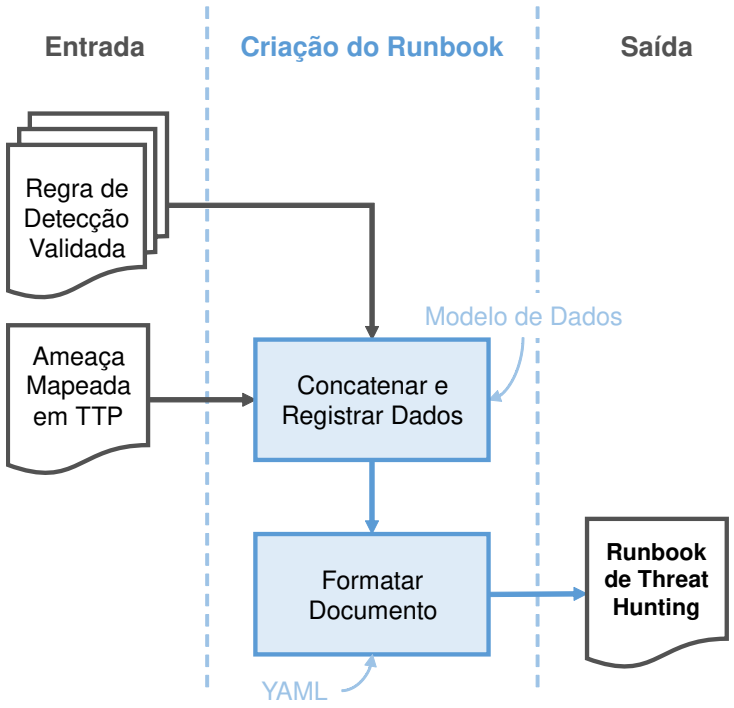


Figura 8.12: Fluxo de ações do 4º estágio: 'Consolidação dos Resultados'

8.4.1 Criação do Runbook

Este passo tem por objetivo criar o Runbook de Threat Hunting consolidando as informações produzidas pela metodologia, estruturadas conforme o modelo de dados definido no Capítulo 7. O Algoritmo 4 apresenta o pseudocódigo correspondente à execução deste estágio, composto pelas seguintes ações:

1. Concatenar e Registrar Dados;
2. Formatar Documento.

1. Concatenar e Registrar Dados

Esta ação consiste em reunir todas as informações geradas nos estágios anteriores, integrando os dados referentes à ameaça, às TTPs mapeadas, às Regras de Detecção propostas, aos resultados de validação e às referências utilizadas ao longo do processo. O registro dessas informações deve seguir o modelo de dados proposto, respeitando a estrutura apresentada no Diagrama de Classes da Figura 7.3, de modo a garantir a correta associação dos objetos e a integridade das relações.

Com o intuito de apoiar os analistas na gestão sistemática dessas informações, foi desenvolvida nesta pesquisa uma ferramenta denominada *Runbook Creator/Editor* [216].

Algoritmo 4 Pseudocódigo referente ao Estágio 4 da metodologia

```
Entrada: mapped_threat; validated_detection_rules
Saída: runbook                                   ▷ Arquivo texto em formato YAML
  function RESULTS_COMPILATION(mapped_threat, validated_detection_rules)
    ▷ /* Concatenar e Registrar Dados */ ◁
    for vDR in validated_detection_rules do
        for TTP in mapped_threat.TTPs do
            if vDR.reference_ttp = TTP.ttp_id then
                TTP.insert(vDR)
                mapped_threat.TTPs.update(TTP)
    ▷ /* Formatar Documento */ ◁
    runbook ← save_to_yaml(mapped_threat)
    return runbook
```

Essa ferramenta implementa o diagrama de classes descrito na Figura 7.3 em um *back-end* robusto, assegurando a consistência e a integridade dos dados registrados.

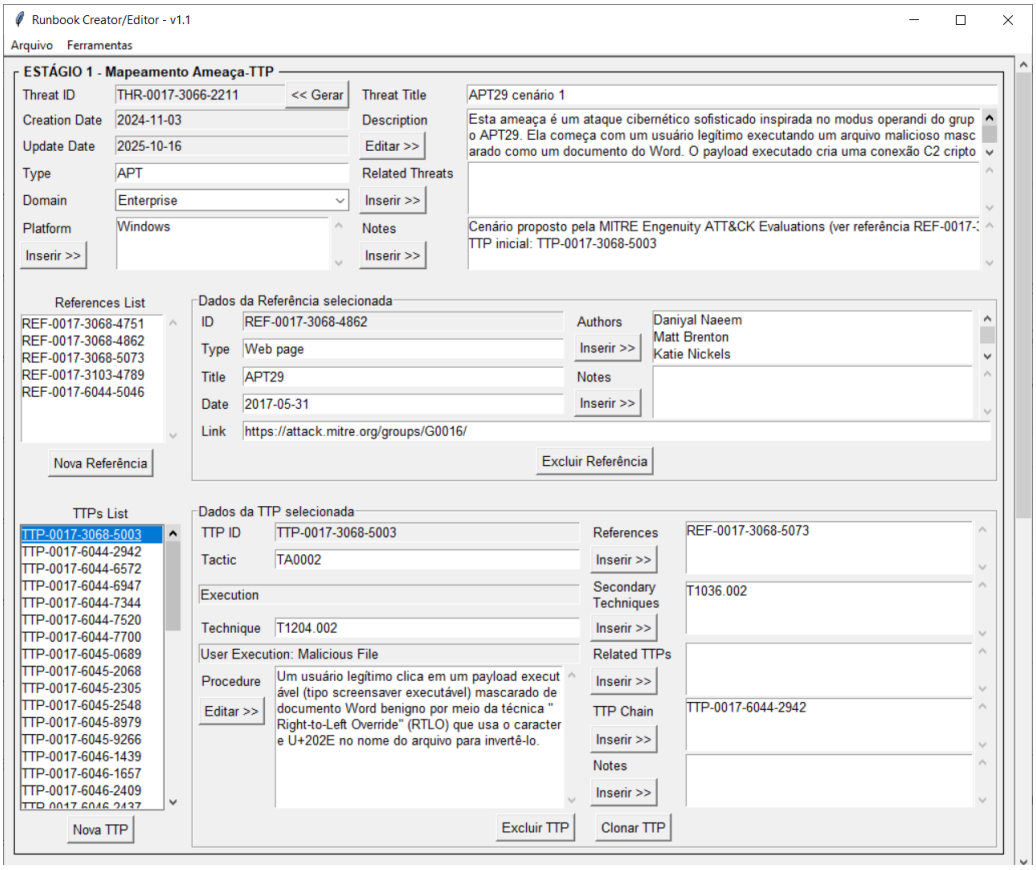
Sua interface gráfica (*front-end*) organiza os campos do modelo de dados de forma lógica e intuitiva, conforme ilustrado na Figura 8.13. Os campos e botões são agrupados de acordo com os estágios da metodologia, o que não apenas simplifica o processo de registro, mas também orienta o usuário durante toda a execução.

Um dos principais benefícios do *Runbook Creator/Editor* é permitir que o analista concentre esforços nas atividades críticas da metodologia, que exigem interpretação especializada e julgamento técnico. Para isso, a ferramenta automatiza tarefas rotineiras, como a geração de identificadores únicos, a inserção automática de datas de criação e atualização, e a recuperação das descrições de Táticas e Técnicas diretamente da base de dados do MITRE ATT&CK [18].

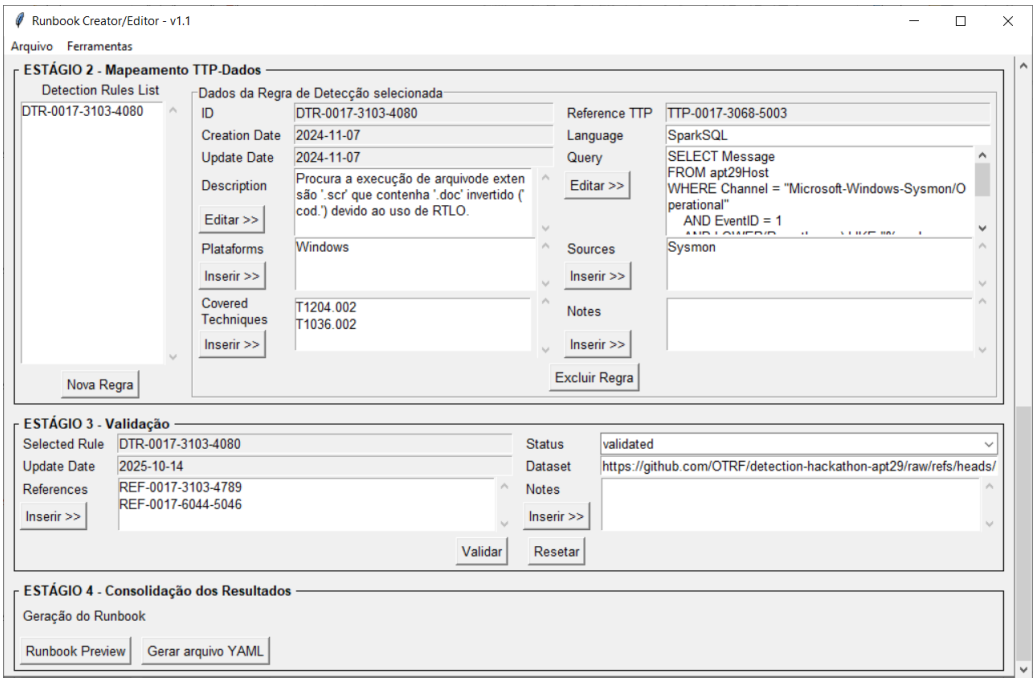
2. Formatar Documento

Esta ação consiste em gerar o Runbook em um formato adequado para armazenamento, transmissão e uso operacional em Threat Hunting. Para isso, os dados estruturados e registrados na etapa anterior devem ser formatados segundo a sintaxe de uma linguagem de representação. Entre as linguagens comumente utilizadas para esse fim destacam-se o JSON, o XML e o YAML.

O YAML é particularmente recomendado para os Runbooks devido à sua sintaxe simples e legiO YAML é particularmente recomendado para a criação de Runbooks devido à sua sintaxe simples e elevada legibilidade, tanto para humanos quanto para sistemas automatizados. Por ser mais conciso e menos verboso que XML e JSON, o YAML facilita a escrita, manutenção e leitura do documento em editores de texto comuns. É essencial, entretanto, garantir que a estrutura do modelo de dados seja respeitada e que a sintaxe



(a) Estágio 1



(b) Estágios 2, 3 e 4

Figura 8.13: Interface gráfica da ferramenta Runbook Creator/Editor

do YAML esteja correta. A conformidade do arquivo pode ser verificada por meio de ferramentas como o YAML Checker[12].

Embora seja possível criar ou editar os Runbooks manualmente em qualquer editor de texto, esse procedimento torna-se suscetível a erros à medida que a complexidade da ameaça aumenta. Para ameaças compostas por múltiplas TTPs e Regras de Detecção, recomenda-se o uso da ferramenta *Runbook Creator/Editor* [216], que permite importar, pré-visualizar e exportar Runbooks em formato YAML em qualquer fase da metodologia, proporcionando maior controle e flexibilidade na gestão das informações.

As primeiras versões das Regras de Detecção — produzidas no primeiro ciclo da metodologia — devem ser entendidas como candidatas iniciais, cuja precisão foi verificada em um ambiente controlado, com dados gerados a partir de um plano de emulação. O ciclo completo de execução da metodologia é sintetizado no pseudocódigo apresentado na Algoritmo 5. Embora a emulação ofereça um cenário adequado para validar a lógica essencial das regras, ela não captura integralmente a complexidade e a variabilidade presentes em ambientes reais.

Algoritmo 5 Pseudocódigo referente à execução completa de um ciclo da metodologia

```
Entrada: threat                                          ▷ Nome da ameaça
Saída: runbook                                           ▷ Arquivo texto em formato YAML
  function METHODOLOGY(threat)
    ▷ /* Estágio 1: Mapeamento Ameaça–TTP */ ◁
    mapped_threat ← threat_ttp_mapping(threat)
    ▷ /* Estágio 2: Mapeamento TTP–Dados */ ◁
    for TTP in mapped_threat.TTPs do
        detection_rule ← ttp_data_mapping(TTP)
        detection_rules.insert(detection_rule)
    ▷ /* Estágio 3: Validação */ ◁
    validated_detection_rules ← validate(mapped_threat, detection_rules)
    ▷ /* Estágio 4: Consolidação dos Resultados */ ◁
    runbook ← results_compilation(mapped_threat, validated_detection_rules)
    return runbook
```

Em ambientes de produção, a efetividade das regras pode ser influenciada por diversos fatores ausentes na etapa controlada de validação, tais como diferenças de configuração, maior diversidade de comportamentos legítimos, particularidades operacionais e ruídos inerentes ao ambiente real. Por essa razão, as regras consolidadas no Runbook inicial devem passar por um processo de ajuste quando aplicadas em contextos reais. Para aprimoramentos consistentes, recomenda-se que dados de produção sejam incorporados nas iterações subsequentes da metodologia, possibilitando a atualização contínua do Runbook de Threat Hunting.

[12]Disponível em `https://yamlchecker.com/`

Capítulo 9

Formalização Matemática

Embora a metodologia proposta tenha sido concebida com foco em sua aplicabilidade prática, seus principais componentes e relações podem ser representados formalmente por meio de uma modelagem matemática. Tal formalização visa oferecer uma base teórica rigorosa que favoreça tanto a análise conceitual quanto o desenvolvimento de abordagens automatizadas baseadas em seus princípios estruturantes.

Seja θ uma ameaça cibernética e $\mathcal{U}_{CTI}$ o universo de informações de inteligência de ameaça (CTI), a metodologia proposta pode ser representada como uma função:

$$\mathfrak{M}_\theta : \mathcal{U}_{CTI} \rightarrow \mathfrak{R}_\theta$$

onde $\mathfrak{M}_\theta$ representa a aplicação da metodologia sobre a ameaça θ, e $\mathfrak{R}_\theta$ representa o conjunto estruturado de informações denominado Runbook da ameaça θ.

O Runbook, conforme definido pelo modelo de dados da metodologia, agrega diversos tipos de informações, que podem ser representadas matematicamente como:

$$\mathfrak{R}_\theta = \{\mathcal{R}_\theta, \mathcal{I}_\theta, \mathcal{T}_\theta, \mathcal{K}_\mathcal{T}, \mathcal{S}_\theta, \mathcal{Q}_\mathcal{T}^V\}$$

tal que:

- $\mathfrak{R}_\theta$ é o Runbook produzido pela metodologia $\mathfrak{M}_\theta$;
- $\mathcal{R}_\theta$ é o conjunto de referências utilizadas na análise da ameaça θ;
- $\mathcal{I}_\theta$ é o conjunto de informações gerais sobre a ameaça θ;
- $\mathcal{T}_\theta$ é o conjunto de Táticas, Técnicas e Procedimentos (TTPs) mapeados para a ameaça θ;
- $\mathcal{K}_\mathcal{T}$ é o conjunto de informações complementares relacionadas ao mapeamento das TTPs de $\mathcal{T}_\theta$, incluindo encadeamentos, técnicas secundárias e relacionamentos;

- $\mathcal{S}_\theta$ é o conjunto de fontes de dados selecionadas para a coleta de eventos observáveis; e
- $\mathcal{Q}_\mathcal{T}^V$ é o conjunto de Regras de Detecção validadas associadas às TTPs de $\mathcal{T}_\theta$.

Esses conjuntos de informações são produzidos nos três primeiros estágios da metodologia: Mapeamento Ameaça-TTP, Mapeamento TTP-Dados e Validação.

O primeiro estágio gera os conjuntos $\mathcal{R}_\theta$, $\mathcal{I}_\theta$, $\mathcal{T}_\theta$ e $\mathcal{K}_\mathcal{T}$, diretamente associados ao mapeamento das TTPs que caracterizam a ameaça. O segundo estágio produz os conjuntos $\mathcal{S}_\theta$ e $\mathcal{Q}_\mathcal{T}$, voltados à construção das regras de detecção com base em dados observáveis. No terceiro estágio, o conjunto $\mathcal{Q}_\mathcal{T}$ é submetido ao processo de validação, originando o subconjunto $\mathcal{Q}_\mathcal{T}^V$. Por fim, o quarto estágio consolida todos os conjuntos produzidos, integrando-os no Runbook. As ações que compõem cada estágio são formalizadas ao longo deste capítulo.

9.1 Estágio 1: Mapeamento Ameaça-TTP

Seja $f_{_R} : \mathcal{U}_\theta \to P(\mathcal{U}_{CTI})$ uma função que busca e seleciona materiais de CTI relacionados à ameaça θ para serem usados como referências na metodologia, tal que $\mathcal{U}_\theta$ é o universo de ameaças cibernéticas e $P(\mathcal{U}_{CTI})$ é o conjunto potência de $\mathcal{U}_{CTI}$, *i.e.*, o conjunto de todos os seus subconjuntos:

$$f_{_R}(\theta) = \{R_i \in \mathcal{U}_{CTI} \mid R_i \mapsto \theta\}$$

onde o símbolo '$\mapsto$' denota "*mapeia para*", significando que a proposição $(R_i \mapsto \theta)$ é verdadeira se R_i pode ser mapeado em θ. No contexto desta metodologia, $(R_i \mapsto \theta) \equiv$ *True* indica que a referência R_i contém informações relevantes sobre a ameaça θ. Assim, $\mathcal{R}_\theta$ pode ser definido como:

$$\mathcal{R}_\theta = \{R_i \mid R_i \in f_{_R}(\theta)\}$$

Seja $\mathcal{N} = \{N_1, \ldots, N_n\}$ o conjunto de informações necessárias que o analista deve buscar para compor o Runbook $\mathfrak{R}_\theta$. Exemplos de $N_i \in \mathcal{N}$ incluem tipo da ameaça, descrição, domínio, plataformas de atuação, entre outras definidas pelo modelo de dados ou pelo analista.

Seja $g_{_I} : \mathcal{N} \to \mathcal{I}_\theta$ uma função que extrai das referências informações gerais da ameaça, para atender à necessidade $N \in \mathcal{N}$. Essa função pode ser definida como:

$$g_{_I}(N) = I \mid \exists R'[(I \mapsto R') \wedge (R' \subseteq \mathcal{R}_\theta) \wedge (N \mapsto I)]$$

onde R' é um subconjunto de $\mathcal{R}_\theta$, $(I \mapsto R') \equiv \mathit{True}$ se a informação I pode ser mapeada nas referências R', e $(N \mapsto I) \equiv \mathit{True}$ se a informação requisitada por N é fornecida por I.

Dessa forma, o conjunto $\mathcal{I}_\theta$ pode ser definido como:

$$\mathcal{I}_\theta = \{I_i \mid \forall N[(I_i = g_I(N)) \wedge (N \in \mathcal{N})]\}$$

Além das informações gerais, as referências $R \in \mathcal{R}_\theta$ podem conter um conjunto $\mathcal{A}_R \subseteq R$ de análises sobre a ameaça, que podem ser relevantes ou não para a execução da metodologia. Seja $f_A : \mathcal{R}_\theta \to P(\mathcal{A}_R)$ uma função que filtra as análises relevantes contidas na referência R, definida como:

$$f_A(R) = \{A_i \in \mathcal{A}_R \mid A_i \mapsto \theta\}$$

onde A_i é uma análise contida na referência R, e $(A_i \mapsto \theta) \equiv \mathit{True}$ se A_i for uma análise relevante da ameaça θ no contexto da metodologia.

Seja $\mathbb{M}$ a base do MITRE ATT&CK [18], $\mathcal{U}_{TA}$ o universo de Táticas e $\mathcal{U}_T$ o universo de Técnicas, tal que $\mathcal{U}_{TA} \subset \mathbb{M}$ e $\mathcal{U}_T \subset \mathbb{M}$. Podemos modelar uma TTP x como uma tupla composta por uma Tática t^a_x, uma Técnica t_x e um Procedimento p_x, tal que $t^a_x \in \mathcal{U}_{TA}$, $t_x \in \mathcal{U}_T$, e p_x é um texto obtido de uma análise de alguma referência de θ, *i.e.* $p_x \subseteq \mathcal{R}'_\theta$.

Sejam $g_{TA} : \mathcal{A}_R \to P(\mathcal{U}_{TA})$ uma função que identifica as Táticas do MITRE ATT&CK contidas em uma análise $A \in f_A(R)$, $g_T : \mathcal{A}_R \times \mathcal{U}_{TA} \to P(\mathcal{U}_T)$ uma função que identifica as Técnicas do MITRE ATT&CK referentes à Tática q da análise A, e $g_P : \mathcal{A}_R \times \mathcal{U}_T \to \mathcal{A}'_R$ uma função que obtém o procedimento referente à técnica t da análise A, ou retorna vazio ($\varnothing$), caso contrário. Essas funções são descritas da seguinte forma:

$$\begin{aligned}
g_{TA}(A) &= \{t^a_i \in \mathcal{U}_{TA} \mid t^a_i \mapsto A'\} \\
g_T(A, t^a_i) &= \{t_j \in \mathcal{U}_T \mid (t_j \mapsto A') \wedge (t_j \to t^a_i)\} \\
g_P(A, t_j) &= \begin{cases} p, & \text{se } \exists p[(p \mapsto A') \wedge (p' \to t_j)] \\ \varnothing, & \text{caso contrário} \end{cases}
\end{aligned}$$

onde A' pode ser igual a A ou uma parte de A, *i.e.* $(A' \leq A)$; analogamente, p' representa p ou uma parte de p, tal que $(p' \leq p)$. A condição $(t^a_i \mapsto A') \equiv \mathit{True}$ indica que a tática t^a_i pode ser identificada na análise A'. De modo semelhante, $(t_j \mapsto A') \equiv \mathit{True}$ se a técnica t_j for mapeável em A', e $(p \mapsto A') \equiv \mathit{True}$ se o procedimento p puder ser derivado de A'. A lógica $(t_j \to t^a_i) \equiv \mathit{True}$ expressa que a técnica t_j está associada à tática t^a_i, no sentido de que a ocorrência de t_j implica a presença de t^a_i. A mesma interpretação se aplica à

sentença $(p' \rightarrow t_j)$, indicando que a ocorrência do procedimento p' implica a utilização da técnica t_j.

Uma TTP x pode ser modelada como uma tupla (t_x^a, t_x, p_x) tal que t_x^a, t_x, p_x são, respectivamente, a Tática, a Técnica e o Procedimento. Dessa forma, o conjunto das TTPs obtidas pela referência $R \in \mathcal{R}_\theta$ é:

$$\mathcal{T}_R = \bigcup_{A \in f_A(R)} \{(t_x^a, t_x, p_x) \mid t_x^a \in g_{TA}(A), t_x \in g_T(A, t_x^a), p_x = g_P(A, t_x)\}$$

Assim, o conjunto de todas as TTPs que compõem a ameaça θ é:

$$\mathcal{T}_\theta = \bigcup_{R \in \mathcal{R}_\theta} \mathcal{T}_R$$

Seja $g_T : A' \rightarrow P(\mathcal{U}_T)$ uma função que identifica as Técnicas do MITRE ATT&CK que compõem um Procedimento p. Essa função pode ser definida como:

$$g_T(p) = \{t_j \in \mathcal{U}_T \mid t_j \mapsto p'\}$$

onde p' pode ser p ou uma parte de p, *i.e.*, $p' \leq p$.

As técnicas secundárias de uma TTP x podem ser representadas pelo conjunto T_x^s, tal que:

$$T_x^s = g_T(p_x) - \{t_x\} = \{t \in g_T(p_x) \mid t \neq t_x\}$$

onde t_x é a técnica principal da TTP x.

Se uma técnica secundária de uma TTP for uma técnica primária de outra TTP, e vice-versa, então há uma relação entre as TTPs. Essa relação pode ser definida por $\rho \subseteq \mathcal{T}_\theta \times \mathcal{T}_\theta$, tal que:

$$\rho = \{(x, y) \in \mathcal{T}_\theta \times \mathcal{T}_\theta \mid \exists t[((t \in T_x^s) \wedge (t = t_y)) \vee ((t \in T_y^s) \wedge (t = t_x))]\}$$
$$x \rho y \Rightarrow (x, y) \in \rho,\ x \in \mathcal{T}_\theta,\ y \in \mathcal{T}_\theta,\ x \neq y$$
$$x \rho y = y \rho x$$

onde t_x e t_y são, respectivamente, as técnicas principais das TTPs x e y, e $x \rho y$ é uma notação que indica que a TTP x tem relação com a TTP y.

O conjunto de relacionamentos de TTPs referentes à TTP x é dado por:

$$\rho_x = \{(x, y) \mid x \rho y\}$$

O encadeamento de TTPs pode ser definido pela relação $\gamma \subseteq \mathcal{T}_\theta \times \mathcal{T}_\theta$, tal que:

$$\gamma = \{(x, y) \in \mathcal{T}_\theta \times \mathcal{T}_\theta \mid (x \to y) \wedge (x \neq y)\}$$
$$x\gamma y \Rightarrow (x, y) \in \gamma,\ x \in \mathcal{T}_\theta,\ y \in \mathcal{T}_\theta,\ x \neq y$$
$$x\gamma y \neq y\gamma x$$

O conjunto de encadeamentos de TTPs com origem em x é dado por:

$$\gamma_x = \{(x, y) \mid x\gamma y\}$$

Essas relações estruturam o conjunto $\mathcal{T}_\theta$ em uma rede semântica, permitindo inferências sobre dependências entre técnicas e caminhos possíveis de ataque. Em especial, γ pode ser interpretada como uma ordenação parcial sobre as TTPs, orientando seu uso sequencial em simulações ou análises dinâmicas.

Formalmente, a cadeia de TTPs da ameaça θ pode ser modelada por um grafo acíclico direcionado (Directed Acyclic Graph – DAG) $\mathcal{G}^c_\theta = (\mathcal{T}_\theta, \gamma)$, no qual os nós correspondem às TTPs de $\mathcal{T}_\theta$ e os arcos dirigidos representam as relações de encadeamento $(x, y) \in \gamma$.

Analogamente, o relacionamento entre as TTPs de $\mathcal{T}_\theta$ pode ser modelado pelo grafo $\mathcal{G}^r_\theta = (\mathcal{T}_\theta, \rho)$; contudo, como ρ é uma relação simétrica ($x\rho y \Leftrightarrow y\rho x$), o grafo $\mathcal{G}^r_\theta$ é não direcionado.

As técnicas secundárias, os relacionamentos e os encadeamentos complementam o mapeamento das TTPs contidas em $\mathcal{T}_\theta$. Para uma TTP $x \in \mathcal{T}_\theta$, essas informações podem ser representadas como uma tupla $(T^s_x, \rho_x, \gamma_x)$. Dessa forma, o conjunto $\mathcal{K}_\mathcal{T}$ pode ser definido como:

$$\mathcal{K}_\mathcal{T} = \{K_x \mid K_x \equiv (T^s_x, \rho_x, \gamma_x),\ x \in \mathcal{T}_\theta\}$$

9.2 Estágio 2: Mapeamento TTP-Dados

Sejam $\mathcal{U}_{DS}$ o universo de *Data Sources* e $\mathcal{U}_{DC}$ o universo de *Data Components*, ambos do MITRE ATT&CK [158], *i.e.*, $\mathcal{U}_{DS} \subset \mathbb{M}$ e $\mathcal{U}_{DC} \subset \mathbb{M}$. Para determinar o escopo de atuação, utilizam-se as funções $g_{DS} : \mathcal{T}_\theta \to \mathcal{U}_{DS}$ e $g_{DC} : \mathcal{U}_{DS} \times \mathcal{T}_\theta \to \mathcal{U}_{DC}$, que identificam, respectivamente, o *Data Source* d^s e o *Data Component* d^c associados a uma TTP x. Essas funções podem ser definidas como:

$$g_{DS}(x) = d^s \mid \exists d^s[(d^s \in \mathcal{U}_{DS}) \wedge (d^s \mapsto \alpha(x))]$$
$$g_{DC}(x, d^s_x) = d^c \mid \exists d^c[(d^c \in \mathcal{U}_{DC}) \wedge (d^c \to d^s_x) \wedge (d^c \mapsto \alpha(x))]$$

onde $\alpha(x)$ representa a TTP x analisada, $(d^s \mapsto \alpha(x)) \equiv \mathit{True}$ se d^s pode ser verificado na análise da TTP x e, analogamente, $(d^c \mapsto \alpha(x)) \equiv \mathit{True}$ se d^c pode ser verificado na análise da TTP x.

Assim, o *Data Source* da TTP x é dado por $d_x^s = g_{DS}(x)$, e o *Data Component* correspondente é $d_x^c = g_{DC}(x, d_x^s) = g_{DC}(x, g_{DS}(x))$.

Sejam $\mathcal{U}_A$ o universo de ações e $\mathcal{U}_C$ o universo de componentes, onde os componentes podem atuar como origem ou alvo em um padrão de comportamento modelado como *(origem, ação, alvo)* ou *(origem/alvo, ação)*. Com base nesse modelo, o conjunto de comportamentos $\mathcal{B}$ pode ser definido como:

$$\mathcal{B} = \{b_i \mid b_i \equiv (c, a, c') \vee b_i \equiv (c, a)\}$$

onde $c, c' \in \mathcal{U}_C$ e $a \in \mathcal{U}_A$.

Seja $g_B : \mathcal{T}_\theta \times \mathcal{U}_{DC} \to P(\mathcal{B})$ uma função que caracteriza os comportamentos adversários de uma TTP x com base em seu escopo de atuação:

$$g_B(x, d_x^c) = \{b_i \in \mathcal{B} \mid (b_i \to d_x^c) \wedge (b_i \mapsto \alpha(x))\}$$

onde d_x^c é o *Data Component* da TTP x, e $(b_i \mapsto \alpha(x)) \equiv \mathit{True}$ indica que o comportamento b_i pode ser verificado na análise da TTP x. O conjunto de comportamentos de uma TTP x é então:

$$B_x = \{b_i \mid b_i \in g_B(x, d_x^c)\}$$

Seja $\mathcal{S}_\pi$ o conjunto das fontes de dados disponíveis em uma plataforma π. Cada fonte de dados $s \in \mathcal{S}_\pi$ fornece um conjunto $E_s = \{e_1, \ldots, e_n\}$ de eventos coletáveis, que pode ser obtido pela função $\varepsilon : \mathcal{S}_\pi \to P(E)$, tal que $\varepsilon(s) = E_s$. Cada evento $e \in E_s$ contém um conjunto $F_e = \{w_1, \ldots, w_m\}$ de campos de dados específicos, que pode ser obtido pela função $\varphi : E \to P(F)$, tal que $\varphi(e) = F_e$.

Define-se $f_S : B_x \to P(\mathcal{S}_\pi)$ como uma função que seleciona as fontes capazes de fornecer dados relevantes à identificação do comportamento b:

$$f_S(b) = \{s_i \mid \exists e[(s_i \in \mathcal{S}_\pi) \wedge (e \in \varepsilon(s_i)) \wedge (b \mapsto \delta(e))]\}$$

onde $\delta(e)$ representa a descrição do evento e com base em sua documentação, e $(b \mapsto \delta(e)) \equiv \mathit{True}$ se, de acordo com a documentação do evento, o comportamento b pode ser mapeado por e.

O conjunto de fontes necessárias para identificar os comportamentos relacionados à TTP x é:

$$S_x = \{s_i \mid \forall b[(s_i \in f_S(b)) \wedge (b \in B_x)]\}$$

Portanto, o conjunto total de fontes utilizadas na detecção da ameaça θ é:

$$\mathcal{S}_\theta = \bigcup_{x \in \mathcal{T}_\theta} S_x$$

Seja $f_{_E} : S_x \times B_x \rightarrow P(E_s)$ uma função que seleciona os eventos relevantes da fonte s para identificação do comportamento b:

$$f_{_E}(s, b) = \{e_i \in \varepsilon(s) \mid b \mapsto \delta(e_i)\}$$

Logo, o conjunto E_x dos eventos selecionados para a identificação da TTP x é:

$$E_x = \{e_i \mid \forall s \forall b[(e_i \in f_{_E}(s, b)) \wedge (b \in B_x)]\}$$

Define-se $f_{_F} : E_s \times \mathcal{T}_\theta \rightarrow P(F_e)$ como a função que seleciona os campos do evento e úteis para a identificação da TTP x:

$$f_{_F}(e, x) = \{w_i \in \varphi(e) \mid w_i \mapsto \alpha(x)\}$$

Sendo assim, o conjunto dos campos selecionados para a identificação da TTP x é:

$$F_x = \{w_i \mid \forall e[(w_i \in f_{_F}(e, x)) \wedge (e \in E_x)]\}$$

Seja $\mathcal{V}_w$ o conjunto de valores possíveis para o campo w, e $g_{_v} : F_e \times \mathcal{T}_\theta \rightarrow \mathcal{V}_w$ a função que determina um valor relevante para esse campo, tal que:

$$g_{_v}(w, x) = v \mid \exists v[(v \mapsto \alpha(x)) \wedge (v \mapsto w)]$$

onde $(v \mapsto \alpha(x)) \equiv \mathit{True}$ se o valor v é identificado na análise da TTP x, e $(v \mapsto w) \equiv \mathit{True}$ se v é mapeado pelo campo w.

Assim, o conjunto V_x dos valores dos campos selecionados para a identificação da TTP x é:

$$V_x = \{v_i \mid \forall w[(v_i = g_{_v}(w, x)) \wedge (w \in F_x)]\}$$

Seja $\mathcal{L}$ um conjunto de sentenças lógicas construídas a partir de um conjunto σ de símbolos — que engloba conectivos lógicos e símbolos de pontuação —, um conjunto τ de termos de variáveis, e um conjunto κ de constantes. Portanto, $\mathcal{L}(\sigma, \tau, \kappa) = \{l_1, \ldots, l_n\}$, tal que $l_i \in \mathcal{L}$ é uma sentença lógica.

A função $g_Q : L \times F \times V \rightarrow P(\mathcal{L})$ gera sentenças lógicas, denominadas *queries*, com base em uma linguagem L, um conjunto de campos F e um conjunto de valores V:

$$g_Q(L, F, V) = \{q_i \mid \exists\sigma[(q_i \in \mathcal{L}(\sigma, F, V)) \wedge (\sigma \mapsto L)]\}$$

onde $(\sigma \mapsto L) \equiv \mathit{True}$ se o conjunto σ de símbolos é mapeado pela linguagem L.

Por exemplo, se a linguagem L escolhida for a Lógica Proposicional, então σ pode ser definido como $\sigma = \{\wedge, \vee, \neg, =, (,)\}$. Assim, uma *query* possível seria:

$$q(w_1, w_2, w_3) \Rightarrow ((w_1 = v_1) \wedge (w_2 = v_2)) \vee ((w_3 = v_3) \wedge \neg(w_3 = v_4))$$

onde w_i são campos de eventos e v_i são valores constantes.

Como as *queries* q geradas pela função $g_Q(L, F, V)$ contêm variáveis w_i em sua sentença lógica, elas podem ser representadas na forma de função $q(w_1, \ldots, w_n)$, tal que $q : P(\mathcal{D}) \rightarrow \{\mathit{True}, \mathit{False}\}$, com $\mathcal{D}$ um conjunto de dados.

O conjunto Q_x^L das *queries* relacionadas à TTP x, que são escritas na linguagem L, é definido como:

$$Q_x^L = \{q_i \mid \exists F \exists V[(q_i \in g_Q(L, F, V)) \wedge (E \subseteq E_x) \wedge (V \subseteq V_x) \wedge (q_i \mapsto \alpha(x))]\}$$

onde $(q_i \mapsto \alpha(x)) \equiv \mathit{True}$ se a *query* q_i desenvolvida está de acordo com a análise da TTP x.

Portanto, o conjunto total das *queries* para as TTPs de $\mathcal{T}_\theta$ é obtido da seguinte forma:

$$\mathcal{Q}_\mathcal{T} = \bigcup_{x \in \mathcal{T}_\theta} Q_x^L$$

9.3 Estágio 3: Validação

Seja $d_t[w, e, s]$ um dado coletado no instante de tempo t, proveniente do campo w de um evento e registrado por uma fonte s. Define-se $D_{t_1,t_2}(\mathcal{S})$ como o conjunto de dados coletados de um conjunto de fontes $\mathcal{S}$ ao longo do intervalo de tempo $[t_1, t_2]$, conforme a expressão:

$$D_{t_1,t_2}(\mathcal{S}) = \{d_t[w, e, s] \mid \forall t \forall w \forall e \forall s[(s \in \mathcal{S}) \wedge (e \in \varepsilon(s)) \wedge (w \in \varphi(e)) \wedge (t_1 \leq t \leq t_2)]\}$$

Sejam D^θ um *dataset malicioso* produzido durante a emulação da ameaça θ e D^β um *dataset benigno* composto por dados não afetados pela ameaça, tais que $D^\theta = D_{t_1,t_2}(\mathcal{S}_\theta)$ e $D^\beta = D_{t_3,t_4}(\mathcal{S}_\theta)$. A validação das Regras de Detecção depende da execução das *queries* $q \in \mathcal{Q}_\mathcal{T}$ em ambos os *datasets* D^θ e D^β.

Para a execução dos testes, define-se a função de filtragem $\Phi_q : \mathcal{D} \to P(\mathcal{D})$, que retorna apenas os dados relevantes à *query* q, *i.e.*, aqueles relacionados aos campos de interesse w_i utilizados na formulação da *query* $q(w_1, \ldots, w_n)$:

$$\Phi_q(D) = \{d_t[w, e, s] \mid (d_t[w, e, s] \in D) \wedge (w \mapsto q)\}$$

onde $(w \mapsto q) \equiv \mathit{True}$ se o campo w está presente como variável na *query* $q(w_1, \ldots, w_n)$.

Para avaliar a *query*, define-se a função de avaliação $f_{ev} : \mathcal{Q}_\mathcal{T} \times \mathcal{D} \to \{\texttt{TRUE}, \texttt{FALSE}\}$, que aplica a lógica da *query* q aos dados previamente filtrados $\Phi_q(D)$ por meio da composição:

$$f_{ev}(q, D) = q \circ \Phi_q(D)$$

A Regra de Detecção é considerada validada quando a avaliação da *query* é positiva sobre o *dataset malicioso* e negativa sobre o *dataset benigno*. Caso contrário, a *query* falha na validação e deve ser revisada ou reformulada..

Dessa forma, o conjunto de Regras de Detecção validadas para todas as TTPs pertencentes ao conjunto $\mathcal{T}_\theta$ é definido por:

$$\mathcal{Q}_\mathcal{T}^V = \{q \in \mathcal{Q}_\mathcal{T} \mid \exists D^\theta, \exists D^\beta [f_{ev}(q, D^\theta) \wedge \neg f_{ev}(q, D^\beta)]\}.$$

9.4 Estágio 4: Consolidação dos Resultados

O produto final da metodologia é o Runbook $\mathfrak{R}$, cujo conteúdo resulta da compilação dos diversos conjuntos de informações produzidos nos estágios anteriores. Um resumo com as especificações desses conjuntos é apresentado na Tabela 9.1, enquanto as definições das principais funções utilizadas na geração desses conjuntos são apresentadas na Tabela 9.2.

A formalização matemática apresentada define, de maneira precisa, os elementos e as relações envolvidos no processo de produção dos Runbooks de Threat Hunting segundo a metodologia proposta.

A modelagem visa abstrair as principais ações do processo, oferecendo uma visão teórica clara e sistemática. No entanto, é importante reconhecer que essa formalização constitui uma abstração da realidade, o que implica simplificações e generalizações de etapas que, na prática, são complexas e fortemente dependentes da expertise dos analistas. A interpretação de relatórios de CTI, a identificação de TTPs e a elaboração de Regras de Detecção envolvem análises e julgamentos qualitativos que não podem ser completamente capturados por modelos matemáticos. A modelagem das *queries*, por exemplo, foi definida com base em linguagens lógicas que retornam valores booleanos ao final da execução. Esse modelo teórico enfatiza a essência conceitual da metodologia, mas não contempla

Tabela 9.1: Definição dos conjuntos de informações produzidos pela metodologia

Estágio	Definição do conjunto	Descrição
1	$\mathcal{R}_\theta = \{R_i \mid R_i \in f_R(\theta)\}$	Conjunto de referências utilizadas na análise da ameaça θ.
1	$\mathcal{I}_\theta = \{I_i \mid \forall N[(I_i = g_I(N)) \wedge (N \in \mathcal{N})]\}$	Conjunto de informações gerais sobre a ameaça θ.
1	$\mathcal{T}_\theta = \bigcup_{R \in \mathcal{R}_\theta} \mathcal{T}_R$, onde $\mathcal{T}_R = \bigcup_{A \in f_A(R)} \{(t_x^a, t_x, p_x) \mid t_x^a \in g_{TA}(A), t_x \in g_T(A, t_x^a), p_x = g_P(A, t_x)\}$	Conjunto de Táticas, Técnicas e Procedimentos (TTPs) mapeados para a ameaça θ.
1	$T_x^s = g_T(p_x) - \{t_x\} = \{t \in g_T(p_x) \mid t \neq t_x\}$	Conjunto de técnicas secundárias de uma TTP x.
1	$\mathcal{K}_\mathcal{T} = \{K_x \mid K_x \equiv (T_x^s, \rho_x, \gamma_x), x \in \mathcal{T}_\theta\}$	Conjunto de informações complementares relacionadas ao mapeamento de TTPs, incluindo encadeamentos, técnicas secundárias e relacionamentos.
2	$B_x = \{b_i \mid b_i \in g_B(x, d_x^c)\}$	Conjunto de comportamentos associados a uma TTP x.
2	$\mathcal{S}_\theta = \bigcup_{x \in \mathcal{T}_\theta} S_x$, onde $S_x = \{s_i \mid \forall b[(s_i \in f_S(b)) \wedge (b \in B_x)]\}$	Conjunto de fontes de dados selecionadas para coleta de eventos observáveis.
2	$E_x = \{e_i \mid \forall s \forall b[(e_i \in f_E(s, b)) \wedge (b \in B_x)]\}$	Conjunto de eventos selecionados para identificar a TTP x.
2	$F_x = \{w_i \mid \forall e[(w_i \in f_F(e, x)) \wedge (e \in E_x)]\}$	Conjunto de campos selecionados para identificar a TTP x.
2	$V_x = \{v_i \mid \forall w[(v_i = g_v(w, x)) \wedge (w \in F_x)]\}$	Conjunto de valores dos campos selecionados para identificar a TTP x.
2	$Q_x^L = \{q_i \mid \exists F \exists V[(q_i \in g_Q(L, F, V)) \wedge (E \subseteq E_x) \wedge (V \subseteq V_x) \wedge (q_i \mapsto \alpha(x))]\}$	Conjunto de Regras de Detecção associadas às TTPs em $\mathcal{T}_\theta$.
3	$D_{t_1,t_2}(\mathcal{S}) = \{d_t[w, e, s] \mid \forall t \forall w \forall e \forall s[(s \in \mathcal{S}) \wedge (e \in \varepsilon(s)) \wedge (w \in \varphi(e)) \wedge (t_1 \leq t \leq t_2)]\}$	Dataset gerado ao longo do intervalo de tempo $[t_1, t_2]$ com dados coletados do conjunto de fontes $\mathcal{S}$.
3	$D^\theta = D_{t_1,t_2}(\mathcal{S}_\theta)$	Dataset malicioso gerado durante a emulação da ameaça θ durante o intervalo de tempo $[t_1, t_2]$.
3	$D^\beta = D_{t_1,t_2}(\mathcal{S}_\theta)$	Dataset benigno gerado durante operações normais ao longo do intervalo de tempo $[t_3, t_4]$.
3	$\mathcal{Q}_\mathcal{T}^V = \{q \in \mathcal{Q}_\mathcal{T} \mid \exists D^\theta, \exists D^\beta[f_{ev}(q, D^\theta) \wedge \neg f_{ev}(q, D^\beta)]\}$	Conjunto de Regras de Detecção validadas associadas às TTPs de $\mathcal{T}_\theta$
4	$\mathfrak{R}_\theta = \{\mathcal{R}_\theta, \mathcal{I}_\theta, \mathcal{T}_\theta, \mathcal{K}_\mathcal{T}, \mathcal{S}_\theta, \mathcal{Q}_\mathcal{T}^V\}$	Runbook de Threat Hunting produzido pela metodologia.

Tabela 9.2: Definição das funções utilizadas na geração dos conjuntos de informações

Definição da função	Descrição
$f_R(\theta) = \{R_i \in \mathcal{U}_{CTI} \mid R_i \mapsto \theta\}$	Recupera e seleciona materiais de CTI relacionados à ameaça θ.
$f_A(R) = \{A_i \in \mathcal{A}_R \mid A_i \mapsto \theta\}$	Filtra as análises relevantes contidas na referência R.
$g_I(N) = I \mid \exists R'[(I \mapsto R') \wedge (R' \subseteq \mathcal{R}_\theta) \wedge (N \mapsto I)]$	Extrai informações da ameaça a partir das referências que atendem à necessidade N.
$g_{TA}(A) = \{t_i^a \in \mathcal{U}_{TA} \mid t_i^a \mapsto A'\}$	Identifica as Táticas ATT&CK contidas em uma análise $A \in f_A(R)$.
$g_T(A, q_i) = \{t_j \in \mathcal{U}_T \mid (t_j \mapsto A') \wedge (t_j \to t_i^a)\}$	Identifica as Técnicas ATT&CK correspondentes à Tática q na análise A.
$g_P(A, t_j) = \begin{cases} p, \text{se } \exists p[(p \mapsto A') \wedge (p' \to t_j)] \\ \varnothing, \text{caso contrário} \end{cases}$	Recupera o Procedimento associado à Técnica t na análise A.
$g_T(p) = \{t_j \in \mathcal{U}_T \mid t_j \mapsto p'\}$	Identifica as Técnicas ATT&CK descritas em um procedimento p.
$g_{DS}(x) = d^s \mid \exists d^s[(d^s \in \mathcal{U}_{DS}) \wedge (d^s \mapsto \alpha(x))]$	Identifica a Fonte de Dados d^s associada a uma TTP x.
$g_{DC}(x, d_x^s) = d^c \mid \exists d^c[(d^c \in \mathcal{U}_{DC}) \wedge (d^c \to d_x^s) \wedge (d \mapsto \alpha(x))]$	Identifica o Componente de Dados d^c associado a uma TTP x.
$g_B(x, d_x^c) = \{b_i \in \mathcal{B} \mid (b_i \to d_x^c) \wedge (b_i \mapsto \alpha(x))\}$	Caracteriza os comportamentos de uma TTP x com base em seu escopo operacional.
$f_S(b) = \{s_i \mid \exists e[(s_i \in \mathcal{S}_\pi) \wedge (e \in \varepsilon(s_i)) \wedge (b \mapsto \delta(e))]\}$	Seleciona fontes capazes de fornecer dados relevantes à identificação do comportamento.
$f_E(s, b) = \{e_i \in \varepsilon(s) \mid b \mapsto \delta(e_i)\}$	Seleciona eventos relevantes da fonte s para identificar o comportamento b.
$f_F(e, x) = \{w_i \in \varphi(e) \mid w_i \mapsto \alpha(x)\}$	Seleciona os campos de eventos úteis à identificação da TTP x.
$g_v(w, x) = v \mid \exists v[(v \mapsto \alpha(x)) \wedge (v \mapsto w)]$	Determina um valor relevante para o campo w.
$g_Q(L, F, V) = \{q_i \mid \exists \sigma[(q_i \in \mathcal{L}(\sigma, F, V)) \wedge (\sigma \mapsto L)]\}$	Gera *queries* com base em uma linguagem L, campos F e valores V.
$\Phi_q(D) = \{d_t[w, e, s] \mid (d_t[w, e, s] \in D) \wedge (w \mapsto q)\}$	Filtra os dados relacionados aos campos w_i usados na *query* $q(w_1, \ldots, w_n)$.
$f_{ev}(q, D) = q \circ \Phi_q(D)$	Avalia a aplicação da *query* q aos dados filtrados $\Phi_q(D)$.

linguagens avançadas, como aquelas baseadas em SQL, que oferecem recursos adicionais e retornam valores de campos específicos para análises contextuais.

Apesar dessas limitações, a formalização estrutura de forma clara as relações e transformações envolvidas, representando fielmente os conceitos centrais da metodologia. Essa

abordagem contribui para a compreensão da lógica subjacente e serve como base sólida para a implementação de ferramentas computacionais.

Capítulo 10

Prova de Conceito

Este capítulo tem por objetivo apresentar a *Prova de Conceito* (PoC) da metodologia proposta, voltada à produção de Runbooks de Threat Hunting. A PoC busca avaliar a eficácia e a aplicabilidade da metodologia sob diferentes condições, por meio de sua execução em três estudos de caso baseados nas seguintes ameaças:

1. LSASS Memory Credential Dump Attack
2. Mosquito Malware
3. APT 29

O primeiro estudo de caso aplica a metodologia a uma microameaça poderosa voltada ao roubo de credenciais do sistema da vítima, amplamente utilizada por grupos APT como parte de ataques sofisticados [217]. Por envolver apenas uma TTP, esse caso permite uma análise mais controlada, com foco na aplicação prática da metodologia no desenvolvimento e validação de regras de detecção.

O segundo estudo de caso tem por objetivo complementar o anterior, restringindo o escopo à execução do primeiro estágio da metodologia, focado no mapeamento detalhado de TTPs. Nesse caso, o foco está na decomposição da ameaça em suas TTPs constituintes e na organização dessas informações de forma clara e estruturada, evidenciando o potencial da metodologia e do modelo de dados em representar comportamentos adversários com base nas TTPs da ameaça.

O terceiro estudo de caso baseia-se em um cenário de ataque proposto pelo programa *MITRE ATT&CK Evaluations* [218], fundamentado em campanhas do grupo APT 29. Este estudo visa executar um ciclo completo da metodologia, testando sua capacidade de produzir um Runbook de Threat Hunting a partir de uma ameaça complexa e multifásica.

Em todos os estudos de caso, a ferramenta *Runbook Creator/Editor* [216] é empregada para o registro das informações geradas durante a execução da metodologia e, em seu estágio final, para a geração do Runbook correspondente.

Assim, esta *Prova de Conceito* busca avaliar, de forma abrangente, a eficácia e a aplicabilidade da metodologia proposta, verificando o desempenho de seus estágios sob diferentes condições e níveis de complexidade de ameaça.

10.1 Estudo de Caso 1: Ataque LSASS Memory Credential Dump

O ataque *LSASS Memory Credential Dump* é uma técnica utilizada por adversários para extrair credenciais de usuários armazenadas na memória do processo LSASS (Local Security Authority Subsystem Service) do Windows. A obtenção de credenciais é um dos principais objetivos de agentes de ameaça, pois essas informações servem como porta de entrada para outros estágios de uma intrusão, como o movimento lateral. O LSASS pode conter não apenas as credenciais de usuários locais, mas também as de administradores de domínio [219]. Por essa razão, essa técnica é amplamente empregada por diversos grupos de cibercriminosos, como APT1, APT5, APT28, GALLIUM, HAFNIUM, Sandworm Team, entre outros [217].

Este estudo de caso demonstra a aplicação completa da metodologia proposta, documentando os resultados de cada estágio até a produção do Runbook de Threat Hunting. No primeiro estágio (Mapeamento Ameaça–TTP), são realizadas a coleta de informações sobre a ameaça, o mapeamento de TTPs e a identificação de relacionamentos. As referências utilizadas foram criteriosamente selecionadas para proporcionar uma compreensão aprofundada do ataque, permitindo um mapeamento preciso das TTPs e fornecendo informações essenciais para os estágios subsequentes. Os dados da Ameaça e das Referências produzidos neste estágio são:

Title: Credentials dump from LSASS process memory
Threat ID: THR-0017-2833-2854
Creation Date: 2024-10-07
Update Date: 2025-01-01
Type: Micro threat
Domain: Enterprise
Platforms: Windows
Description: After a user logs on, the system generates and stores various credentials in the LSASS (Local Security Authority Subsystem Service) process in memory. Adversaries may use resources to access LSASS and extract the credentials from memory.
References:

- ID: REF-0017-2833-3470

Type: Web page
Title: LSASS Memory Read Access
Authors: Roberto Rodriguez
Date: 2017-01-05
Link: `https://threathunterplaybook.com/hunts/windows/170105-LSASSMemoryReadAccess/notebook.html`

- ID: REF-0017-2834-4191
  Type: Web page
  Title: OS Credential Dumping: LSASS Memory
  Authors: MITRE
  Date: 2020-02-11
  Link: `https://attack.mitre.org/techniques/T1003/001/`

Estágio 1: Mapeamento Ameaça–TTP

A análise das referências indicou que essa ameaça está associada à Tática TA0006 (*Credential Access*) [220] e à Técnica T1003 (*OS Credential Dumping*), especificamente à subtécnica T1003.001 (*OS Credential Dumping: LSASS Memory*) [217]. Há diversas formas de implementar esse ataque, incluindo o uso de ferramentas como *ProcDump*, *comsvcs.dll*, *Task Manager* e *Mimikatz* [221, 219], cada uma resultando em procedimentos distintos.

Neste estudo de caso, optou-se por mapear o procedimento baseado no uso da ferramenta *Mimikatz* (e suas variantes), devido à sua ampla utilização e eficácia comprovada na extração de credenciais do LSASS. Essa escolha também permite ilustrar de forma detalhada as etapas críticas da metodologia, fornecendo um exemplo prático e representativo para a PoC.

A etapa final deste estágio consiste na identificação de relacionamentos entre TTPs. Como essa ameaça envolve apenas uma TTP, não há encadeamentos nem TTPs secundárias associadas. As informações resultantes do mapeamento da TTP são:

TTP ID: TTP-0017-2833-3205
Tactic: TA0006
Technique: T1003.001
Procedure: Adversaries may use the Mimikatz tool to access the contents stored in the memory of the LSASS (Local Security Authority Subsystem Service) process. Using the SEKURLSA::LogonPasswords module, Mimikatz attempts to obtain credential data by listing all available provider credentials. The module uses a Kernel32 function called OpenProcess to obtain an identifier for LSASS. It then accesses LSASS and extracts

the NTLM hash of currently (or recently) logged on accounts, as well as the credentials of services running in the user's context.

References: REF-0017-2833-3470; REF-0017-2834-4191

Estágio 2: Mapeamento TTP–Dados

O segundo estágio da metodologia é responsável pelo mapeamento TTP–Dados, cujo objetivo é produzir regras de detecção capazes de identificar a TTP em dados coletados de diferentes fontes. Esse processo é conduzido com base na estratégia de detecção definida pelo analista, a qual resulta da análise detalhada de como a ameaça executa a TTP no sistema-alvo e dos efeitos gerados por essa interação. A definição dessa estratégia é um fator crítico que depende da experiência e julgamento técnico do analista.

Para tornar este estudo de caso mais robusto, foram desenvolvidas quatro Regras de Detecção distintas, cada uma baseada em uma estratégia de detecção diferente, com o propósito de avaliar a aplicabilidade da metodologia sob perspectivas complementares.

Regra de Detecção #1

A primeira Regra de Detecção tem como objetivo identificar contas não pertencentes ao sistema que solicitam ou obtêm acesso ao processo LSASS. Nesse contexto, o escopo da atividade adversária é '*acesso a processo*', e os comportamentos a serem observados são '*processo acessa processo*' e '*processo requisita acesso a processo*'.

Utilizando o *Windows Security Auditing* como fonte de dados, os eventos mais adequados para identificar esses comportamentos são o 4663 (*An attempt was made to access an object*) [222] e o 4656 (*A handle to an object was requested*) [223]. Esses eventos estão disponíveis em plataformas Windows Server 2008 ou superior (incluindo Windows Vista). Assim, a regra gerada não é aplicável a sistemas anteriores. As informações de contexto e o mapeamento de dados estão resumidos na Tabela 10.1.

Tabela 10.1: Mapeamento TTP-Dados da Regra de Detecção #1

Análise de contexto		Mapeamento de dados	
Escopo	Comportamento (Origem Ação Alvo)	Evento*	Campos relevantes
Process access	Process access process	4663	Channel, EventID, ObjectName, SubjectUserName
Process access	Process requests access to process	4656	Channel, EventID, ObjectName, SubjectUserName

*Event provider: Microsoft-Windows-Security-Auditing.

A principal diferença entre os eventos 4663 e 4656 é que o primeiro indica que o direito de acesso foi efetivamente usado, enquanto o segundo representa apenas a requisição do acesso. Além disso, o evento 4663 também inclui registros de falha [222]. O evento 4661, embora semelhante, não é adequado para esta análise, pois se refere a solicitações de

identificadores de objetos do *Active Directory* ou do *Security Account Manager* (SAM) [222], e não a processos do sistema.

De acordo com a estratégia de detecção, a regra deve verificar duas condições nos eventos 4656 ou 4663: (i) se o campo *ObjectName* termina em '*lsass.exe*', indicando tentativa de acesso ao processo LSASS; e (ii) se o campo *SubjectUserName* não termina com o caractere '*$*', o que indica que o usuário não é uma conta de sistema. Os campos *Channel* e *EventID* são utilizados para identificar os eventos relevantes.

A query, escrita em *SparkSQL*, é apresentada a seguir, juntamente com as informações completas da Regra de Detecção resultante.

Rule ID: DTR-0017-2834-5168
Creation Date: 2024-10-07
Update Date: 2024-12-30
Reference TTP: TTP-0017-2833-3205
Description: Search for non-system accounts attempting to access LSASS. While most adversaries inject themselves into a system process to hide among the majority of applications accessing LSASS, there are instances when attackers use administrator rights instead of escalating their access to the system level, as this is the minimum requirement for accessing LSASS.
Platforms: Windows 2008 Server/Vista or higher
Covered Techniques: T1003.001
Sources: Windows Security Auditing
Language: SparkSQL
Query:

```
SELECT `@timestamp`, SubjectUserName, ProcessName, ObjectName,
    ↪  AccessMask, EventID
FROM data_table
WHERE LOWER(Channel) = "security"
AND (EventID = 4663 OR EventID = 4656)
AND ObjectName LIKE "%lsass.exe"
AND NOT SubjectUserName LIKE "%$"
```

Regra de Detecção #2

A segunda Regra de Detecção tem como objetivo identificar processos que tentam acessar o LSASS por meio de identificadores abertos por DLLs não nativas do Windows. Para isso, utiliza-se o *Sysmon* como fonte de dados, sendo o evento 10 (*ProcessAccess*) [5] o mais adequado para representar esse comportamento. As informações resultantes da análise de contexto e do mapeamento de dados estão resumidas na Tabela 10.2.

De acordo com a estratégia de detecção, a regra deve verificar dois critérios no evento 10: (i) se o campo *TargetImage* termina em '*lsass.exe*', indicando tentativa de acesso

Tabela 10.2: Mapeamento TTP-Dados da Regra de Detecção #2

Análise de contexto		Mapeamento de dados	
Escopo	Comportamento (Origem Ação Alvo)	Evento*	Campos relevantes
Process access	Process access process	10	Channel, EventID, TargetImage, CallTrace

*Event provider: Microsoft-Windows-Sysmon.

ao processo LSASS; e (ii) se o campo *CallTrace* contém o termo '*UNKNOWN*', o que sugere uma chamada proveniente de um módulo desconhecido, possivelmente uma cópia maliciosa da função *NtOpenProcess* (ou equivalente) mapeada na memória [224].

Os campos *Channel* e *EventID* são utilizados para identificar os eventos relevantes. A query, escrita em *SparkSQL*, é apresentada abaixo, juntamente com os metadados da Regra de Detecção resultante.

Rule ID: DTR-0017-2891-6445
Creation Date: 2024-10-14
Update Date: 2024-12-30
Reference TTP: TTP-0017-2833-3205
Description: Look for any processes that attempt to access LSASS through handles opened by unknown modules, *i.e.*, not Windows native DLLs.
Platforms: Windows
Covered Techniques: T1003.001
Sources: Windows Sysmon
Language: SparkSQL
Query:

```
SELECT `@timestamp`, SourceImage, TargetImage, GrantedAccess,
    ↪ SourceProcessGUID, CallTrace
FROM data_table
WHERE Channel = "Microsoft-Windows-Sysmon/Operational"
AND EventID = 10
AND TargetImage LIKE "%lsass.exe"
AND CallTrace LIKE "%UNKNOWN%"
```

Regra de Detecção #3

A terceira Regra de Detecção tem como objetivo identificar processos que carregam bibliotecas dinâmicas (DLLs) normalmente associadas aos módulos do Mimikatz, usadas para interação com credenciais. Nesse caso, também se utiliza o *Sysmon* como fonte de dados, sendo o evento 7 (*Image loaded*) [5] o mais apropriado para representar o comportamento observado. As informações de contexto e mapeamento de dados estão resumidas na Tabela 10.3.

O Mimikatz depende de diversos módulos para funcionar, mas há cinco DLLs específicas cuja presença simultânea no mesmo processo caracteriza um comportamento

Tabela 10.3: Mapeamento TTP–Dados da Regra de Detecção #3

Análise de contexto			Mapeamento de dados
Escopo	Comportamento (Origem Ação Alvo)	Evento*	Campos relevantes
Module load	Process load module	7	Channel, EventID, ImageLoaded

*Event provider: Microsoft-Windows-Sysmon.

altamente suspeito: *winscard.dll*, *cryptdll.dll*, *hid.dll*, *samlib.dll* e *vaultcli.dll* [225]. Esses módulos podem ser utilizados como uma espécie de "assinatura comportamental", reduzindo o número de falsos positivos ao detectar a execução do Mimikatz na memória. Quanto maior o número dessas DLLs carregadas por um mesmo processo, maior o nível de suspeita e, portanto, melhor a precisão da detecção da TTP.

Para implementar essa estratégia, a regra deve inspecionar o campo *ImageLoaded* do evento 7, agrupando processos com o mesmo *ProcessGuid* e *Image*. Assim, é possível contabilizar a quantidade de DLLs potencialmente maliciosas carregadas pelo mesmo processo dentro de um intervalo de tempo definido, aprimorando a correlação e reduzindo falsos positivos. Os campos *Channel* e *EventID* são utilizados para identificar os eventos relevantes. A query, escrita em *SparkSQL*, é apresentada a seguir, juntamente com os metadados da regra.

Rule ID: DTR-0017-2893-5282

Creation Date: 2024-10-14

Update Date: 2024-12-30

Reference TTP: TTP-0017-2833-3205

Description: Search for processes that load known DLLs normally loaded by Mimikatz modules to interact with credentials. The more of these DLLs that are loaded simultaneously by the same process, the more suspicious the behavior.

Platforms: Windows

Covered Techniques: T1003.001

Sources: Windows Sysmon

Language: SparkSQL

Query:
```
SELECT ProcessGuid, Image, COUNT(DISTINCT ImageLoaded) AS hits
FROM data_table
WHERE Channel = "Microsoft-Windows-Sysmon/Operational"
AND EventID = 7
AND ( ImageLoaded LIKE "%samlib.dll"
OR ImageLoaded LIKE "%vaultcli.dll"
OR ImageLoaded LIKE "%hid.dll"
```

```
OR ImageLoaded LIKE "%winscard.dll"
OR ImageLoaded LIKE "%cryptdll.dll" )
AND `@timestamp` BETWEEN "2020-06-01 00:00:00.000" AND "2020-08-20
↪  00:00:00.000"
GROUP BY ProcessGuid,Image ORDER BY hits DESC LIMIT 10
```

Regra de Detecção #4

A quarta Regra de Detecção combina as estratégias definidas nas Regras #2 e #3, de modo a aprimorar a precisão na identificação do comportamento adversário. A integração dessas abordagens permite correlacionar processos que, além de acessarem o LSASS por meio de identificadores abertos por módulos desconhecidos, também carregam múltiplas DLLs associadas ao Mimikatz. Essa correlação contribui para reduzir falsos positivos e elevar o nível de confiança na detecção.

Para a integração das estratégias, a regra agrupa processos que possuem o mesmo *ProcessGUID* e que atendem simultaneamente aos critérios das Regras #2 e #3. Na implementação, define-se como requisito mínimo para satisfazer a Regra #3 o carregamento de pelo menos três das cinco DLLs suspeitas identificadas anteriormente. A query proposta em *SparkSQL*, bem como os metadados correspondentes, são apresentados a seguir.

Rule ID: DTR-0017-2893-8676
Creation Date: 2024-10-14
Update Date: 2024-12-30
Reference TTP: TTP-0017-2833-3205
Description: Search for the occurrence of processes that attempt to access LSASS through handles opened by unknown modules and load some specific DLLs that are frequently used by Mimikatz for credential interactions.
Platforms: Windows
Covered Techniques: T1003.001
Sources: Windows Sysmon
Language: SparkSQL
Query:

```
SELECT a.`@timestamp`, b.Image, a.SourceProcessGUID
FROM data_table a
INNER JOIN (
SELECT ProcessGuid, Image, COUNT(DISTINCT ImageLoaded) AS hits
FROM data_table
WHERE Channel = "Microsoft-Windows-Sysmon/Operational"
AND EventID = 7
AND ( ImageLoaded LIKE "%samlib.dll"
OR ImageLoaded LIKE "%vaultcli.dll"
OR ImageLoaded LIKE "%hid.dll"
```

```
OR ImageLoaded LIKE "%winscard.dll"
OR ImageLoaded LIKE "%cryptdll.dll" )
AND `@timestamp` BETWEEN "2020-06-01 00:00:00.000" AND "2020-10-20
↪  00:00:00.000"
GROUP BY ProcessGuid,Image ORDER BY hits DESC LIMIT 10
) b
ON a.SourceProcessGUID = b.ProcessGuid
WHERE a.Channel = "Microsoft-Windows-Sysmon/Operational"
AND a.EventID = 10
AND a.TargetImage LIKE "%lsass.exe"
AND a.CallTrace LIKE "%UNKNOWN%"
AND b.hits >= 3
```

Estágio 3: Validação

O terceiro estágio da metodologia consiste na validação das Regras de Detecção propostas. Nesta PoC, o foco da validação recai exclusivamente sobre a análise das regras em dados maliciosos, com o propósito de avaliar a eficácia da metodologia em produzir Regras de Detecção capazes de identificar os comportamentos adversários associados à TTP mapeada.

Para isso, as regras desenvolvidas são testadas e avaliadas em laboratório, com base em dados coletados durante a emulação da ameaça em ambiente controlado. Neste estudo de caso, contudo, optou-se pela utilização de um dataset pré-gravado disponibilizado publicamente na internet, contendo dados coletados por terceiros a partir da mesma ameaça emulada. Essa escolha visa empregar dados imparciais e assegurar uma validação livre de vieses. Embora essa abordagem não permita ajustes finos para cenários personalizados, ela é adequada para ameaças simples e amplamente conhecidas pela comunidade.

O dataset selecionado, disponível em [226], contém registros de eventos de auditoria de segurança do Windows e do Sysmon capturados durante tentativas de leitura de credenciais da memória do processo LSASS. Segundo o autor, os dados foram gerados a partir da execução do módulo *credentials/mimikatz/logonpasswords* do framework Empire [227], uma ferramenta amplamente utilizada em testes de intrusão e atividades de pós-exploração [228].

As informações referentes ao dataset mencionado estão apresentadas a seguir e devem ser registradas no Runbook juntamente com as demais referências.

ID: REF-0017-3008-1396
Type: Dataset
Title: Empire Mimikatz LogonPasswords
Authors: Roberto Rodriguez
Date: 2020-09-20

Link: https://securitydatasets.com/notebooks/atomic/windows/credential_access/SDWIN-190518202151.html

Notes: Dataset file at https://github.com/OTRF/Security-Datasets/raw/refs/heads/master/datasets/atomic/windows/credential_access/host/empire_mimikatz_logonpasswords.zip

A validação das Regras de Detecção, escritas em *SparkSQL*, foi realizada com o auxílio do Apache Spark, um mecanismo de análise unificado para processamento de dados em larga escala [209]. A interação com o Spark ocorreu por meio da API *PySpark*, utilizando o *Jupyter Notebook* como interface principal, conforme ilustrado na Figura 10.1.

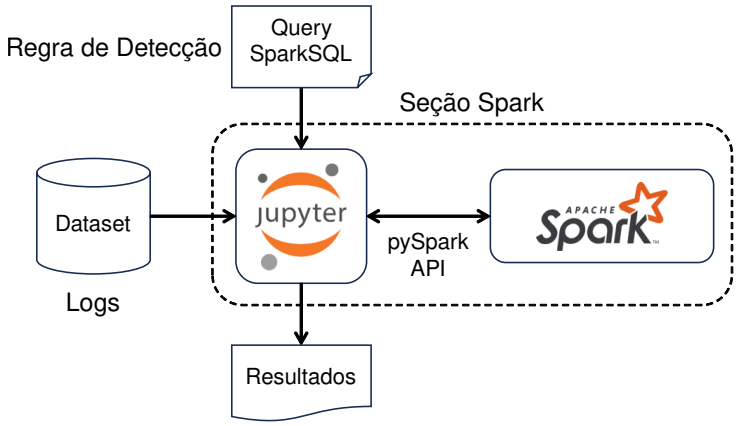


Figura 10.1: Esquema de validação das Regras de Detecção

O processo iniciou-se com a criação de uma sessão Spark, o carregamento do dataset e sua conversão em um *DataFrame*, permitindo a execução das consultas *SparkSQL*, conforme apresentado na Figura 10.2.

```
#Starting Spark session
from pyspark.sql import SparkSession
spark = SparkSession.builder.getOrCreate()
spark.conf.set("spark.sql.caseSensitive", "true")

#Loading Dataset
df = spark.read.json('empire_mimikatz_logonpasswords_2020-08-07103224.json')
df.createTempView('data_table')
```

Figura 10.2: Inicialização da sessão Spark e carregamento do dataset

As Figuras 10.3, 10.4, 10.5 e 10.6 mostram a execução das Regras de Detecção #1, #2, #3 e #4, respectivamente, e os resultados obtidos. O arquivo completo contendo todos os testes realizados encontra-se disponível em https://github.com/pauloferraz80/Runbook_Creator_Editor/blob/main/validation/LSASScredentialDump_EN_SparkSQL_validation.ipynb.

A primeira query (Figura 10.3) retornou dois *hits* associados ao mesmo usuário, sendo um referente ao evento 4656 e outro ao evento 4663. A segunda query (Figura 10.4) identificou um *hit* para o processo de GUID `297bc33e-65d0-5f2d-8207-000000000400`. O resultado da terceira query (Figura 10.5) demonstrou que esse mesmo processo carregou quatro das cinco DLLs monitoradas. Por fim, a quarta query (Figura 10.6), que combina as regras #2 e #3, retornou um único *hit* referente a processos que carregaram três ou mais DLLs suspeitas.

Todos os resultados foram consistentes com o comportamento esperado, demonstrando a eficácia das regras em identificar a execução da TTP em dados coletados durante a emulação da ameaça.

```
df = spark.sql(
'''
SELECT `@timestamp`, SubjectUserName, ProcessName, ObjectName, AccessMask, EventID
FROM data_table
WHERE LOWER(Channel) = "security"
    AND (EventID = 4663 OR EventID = 4656)
    AND ObjectName LIKE "%lsass.exe"
    AND NOT SubjectUserName LIKE "%$"
'''
)
df.show(10,False)
```

```
+-----------------------+--------------+-----------------------------------------------------
+-----------------------------------------------+----------+-------+
|@timestamp             |SubjectUserName|ProcessName                                          |ObjectName
|AccessMask|EventID|
+-----------------------+--------------+-----------------------------------------------------
+-----------------------------------------------+----------+-------+
|2020-08-07T14:32:57.592Z|pgustavo       |C:\Windows\System32\WindowsPowerShell\v1.0\powershell.exe|\Device\Harddisk
Volume2\Windows\System32\lsass.exe|0x1010    |4656   |
|2020-08-07T14:32:57.592Z|pgustavo       |C:\Windows\System32\WindowsPowerShell\v1.0\powershell.exe|\Device\Harddisk
Volume2\Windows\System32\lsass.exe|0x10      |4663   |
+-----------------------+--------------+-----------------------------------------------------
+-----------------------------------------------+----------+-------+
```

Figura 10.3: Validação da Regra de Detecção #1 (DTR-0017-2834-5168)

```
df = spark.sql(
'''
SELECT `@timestamp`, SourceImage, TargetImage, GrantedAccess, SourceProcessGUID
FROM data_table
WHERE Channel = "Microsoft-Windows-Sysmon/Operational"
    AND EventID = 10
    AND TargetImage LIKE "%lsass.exe"
    AND CallTrace LIKE "%UNKNOWN%"
'''
)
df.show(10,False)
```

```
+-----------------------+-----------------------------------------------------+---------------------------
+-------------+-------------------------------------+
|@timestamp             |SourceImage                                          |TargetImage                |Gr
antedAccess|SourceProcessGUID                    |
+-----------------------+-----------------------------------------------------+---------------------------
+-------------+-------------------------------------+
|2020-08-07T14:32:57.589Z|C:\windows\System32\WindowsPowerShell\v1.0\powershell.exe|C:\windows\system32\lsass.exe|0x
1010       |{297bc33e-65d0-5f2d-8207-000000000400}|
+-----------------------+-----------------------------------------------------+---------------------------
+-------------+-------------------------------------+
```

Figura 10.4: Validação da Regra de Detecção #2 (DTR-0017-2891-6445)

```
df = spark.sql(
'''
SELECT ProcessGuid,Image, COUNT(DISTINCT ImageLoaded) AS hits
FROM data_table
WHERE Channel = "Microsoft-Windows-Sysmon/Operational"
    AND EventID = 7
    AND (
        ImageLoaded LIKE "%samlib.dll"
        OR ImageLoaded LIKE "%vaultcli.dll"
        OR ImageLoaded LIKE "%hid.dll"
        OR ImageLoaded LIKE "%winscard.dll"
        OR ImageLoaded LIKE "%cryptdll.dll"
    )
    AND `@timestamp` BETWEEN "2020-06-01 00:00:00.000" AND "2020-08-20 00:00:00.000"
    GROUP BY ProcessGuid,Image ORDER BY hits DESC LIMIT 10
'''
)
df.show(10,False)
```

```
+------------------------------------+-----------------------------------------------------------+----+
|ProcessGuid                         |Image                                                      |hits|
+------------------------------------+-----------------------------------------------------------+----+
|{297bc33e-65d0-5f2d-8207-000000000400}|C:\Windows\System32\WindowsPowerShell\v1.0\powershell.exe|4   |
+------------------------------------+-----------------------------------------------------------+----+
```

Figura 10.5: Validação da Regra de Detecção #3 (DTR-0017-2893-5282)

```
df = spark.sql(
'''
SELECT a.`@timestamp`, b.Image, a.SourceProcessGUID
FROM data_table a
INNER JOIN (
    SELECT ProcessGuid,Image, COUNT(DISTINCT ImageLoaded) AS hits
    FROM data_table
    WHERE Channel = "Microsoft-Windows-Sysmon/Operational"
        AND EventID = 7
        AND (
            ImageLoaded LIKE "%samlib.dll"
            OR ImageLoaded LIKE "%vaultcli.dll"
            OR ImageLoaded LIKE "%hid.dll"
            OR ImageLoaded LIKE "%winscard.dll"
            OR ImageLoaded LIKE "%cryptdll.dll"
        )
        AND `@timestamp` BETWEEN "2020-06-01 00:00:00.000" AND "2020-10-20 00:00:00.000"
        GROUP BY ProcessGuid,Image ORDER BY hits DESC LIMIT 10
    ) b
ON a.SourceProcessGUID = b.ProcessGuid
WHERE a.Channel = "Microsoft-Windows-Sysmon/Operational"
    AND a.EventID = 10
    AND a.TargetImage LIKE "%lsass.exe"
    AND a.CallTrace LIKE "%UNKNOWN%"
    AND b.hits >= 3
'''
)
df.show(10,False)
```

```
+-----------------------+-----------------------------------------------------
+-------------------------------------+
|@timestamp             |Image                                                |SourceProcessGUID
|
+-----------------------+-----------------------------------------------------
+-------------------------------------+
|2020-08-07T14:32:57.589Z|C:\Windows\System32\WindowsPowerShell\v1.0\powershell.exe|{297bc33e-65d0-5f2d-8207-0000000
00400}|
+-----------------------+-----------------------------------------------------
+-------------------------------------+
```

Figura 10.6: Validação da Regra de Detecção #4 (DTR-0017-2893-8676)

Após a análise e verificação dos resultados, as Regras de Detecção foram consideradas validadas. Os dados registrados durante o processo de validação da Regra de Detecção #1, de acordo com o modelo de dados proposto, são os seguintes:

Status: validated
Update Date: 2024-10-27
Dataset: `https://github.com/OTRF/Security-Datasets/raw/refs/heads/master/datasets/atomic/windows/credential_access/host/empire_mimikatz_logonpasswords.zip`
References: REF-0017-3008-1396

Estágio 4: Consolidação dos Resultados

O estágio final da metodologia consiste na consolidação e documentação das informações geradas nas etapas anteriores, com base no modelo de dados descrito no Capítulo 7. A criação do *Runbook* foi realizada com o apoio da ferramenta *Runbook Creator/Editor* [216], desenvolvida como parte desta pesquisa para auxiliar o registro das informações produzidas durante a aplicação da metodologia.

O Runbook de Threat Hunting resultante deste estudo de caso, estruturado em formato YAML, está disponível em: `https://github.com/pauloferraz80/Runbook_Creator_Editor/blob/main/runbooks/LSASScredentialDump_EN_SparkSQL.yml`.

10.2 Estudo de Caso 2: Mosquito Malware

Este estudo de caso aplica a metodologia proposta à ameaça conhecida como Mosquito, um *malware* do tipo *Win32 Backdoor* atribuído ao grupo hacker russo Turla. O grupo Turla, também conhecido pelos nomes Snake, Venomous Bear, Uroburos e WhiteBear, é classificado como um grupo APT (Advanced Persistent Threats) em atividade desde 2004 [229]. Suas operações concentram-se em campanhas de ciberespionagem direcionadas a entidades governamentais, embaixadas, organizações militares e empresas do setor farmacêutico [230].

Diferentemente da microameaça analisada no estudo de caso anterior, o Mosquito é uma ameaça sofisticada, composta por múltiplas Táticas, Técnicas e Procedimentos (TTPs). Essa complexidade permite uma avaliação mais abrangente do primeiro estágio da metodologia, pois requer um mapeamento detalhado das TTPs e de seus relacionamentos ao longo da cadeia de ataque.

Assim, este estudo de caso complementa a Prova de Conceito (PoC) ao avaliar a capacidade da metodologia em identificar e estruturar as TTPs de uma ameaça complexa.

A análise concentra-se exclusivamente no primeiro estágio (Mapeamento Ameaça-TTP) permitindo uma avaliação aprofundada da aplicabilidade e da eficácia da abordagem proposta.

No primeiro passo do mapeamento Ameaça-TTP, correspondente à obtenção de conhecimento sobre a ameaça, foram reunidas informações essenciais sobre o Mosquito. As principais fontes consultadas incluem um *CTI Report* da ESET e a página da MITRE ATT&CK. A seguir, apresentam-se as informações gerais sobre a ameaça, organizadas conforme o modelo de dados proposto, juntamente com as referências utilizadas durante a execução da metodologia.

Title: Mosquito Win32 Backdoor
Threat ID: THR-0017-2739-5252
Creation Date: 2024-09-26
Update Date: 2025-01-31
Type: Malware
Domain: Enterprise
Platforms: Windows
Description: Mosquito is a Win32 backdoor used by the Turla group. It is composed of three parts: the installer, the launcher, and the backdoor.
References:

- ID: REF-0017-2739-5256
  Type: Report
  Title: Diplomats in Eastern Europe bitten by a Turla mosquito
  Authors: ESET, spol. s.r.o.
  Date: 2018-01
  Link: `https://web-assets.esetstatic.com/wls/2018/01/ESET_Turla_Mosquito.pdf`
- ID: REF-0017-2739-5639
  Type: Webpage
  Title: Mosquito
  Authors: MITRE
  Date: 2024-04-11
  Link: `https://attack.mitre.org/software/S0256/`

A execução dos passos seguintes, correspondentes ao mapeamento de TTPs e à identificação de relacionamentos, baseou-se nas informações detalhadas no relatório "Diplomats in Eastern Europe bitten by a Turla mosquito", da ESET [231]. A análise concentrou-se no Capítulo 4 ("Analysis of the WIN32 Backdoor") do documento, que descreve em

profundidade o funcionamento das amostras do Mosquito identificadas em 2017. Essa versão do *malware* é composta por três módulos principais: o *instalador*, responsável pela persistência inicial; o *lançador*, que executa a carga maliciosa; e o *backdoor principal*, que viabiliza o controle remoto do sistema comprometido.

Para o mapeamento das TTPs, as Táticas e Técnicas foram identificadas com base na taxonomia do MITRE ATT&CK [18], enquanto os Procedimentos foram extraídos diretamente do relatório de inteligência. Em seguida, esses Procedimentos foram analisados para identificar técnicas secundárias e inter-relações entre TTPs. Com base nessa análise, foi possível estabelecer sequências de ações e mapear os encadeamentos entre TTPs, oferecendo uma representação estruturada e compreensiva do comportamento da ameaça.

Como resultado, foram identificadas e documentadas **29 TTPs** associadas ao malware Mosquito. A Tabela 10.4 apresenta um resumo com as principais informações obtidas no mapeamento.

Tabela 10.4: Resumo das principais informações das TTPs mapeadas do Mosquito Malware

Nr	TTP ID	Tática*	Técnica*	Técnicas secundárias*	TTP seguinte (encadeamento ou TTP Chain)
1	TTP-0017-2739-5258	TA0002	T1204.002		TTP-0017-2739-5978
2	TTP-0017-2739-5978	TA0005	T1027		TTP-0017-2740-3192
3	TTP-0017-2740-3192	TA0005	T1497.001	T1106	TTP-0017-2740-3422
4	TTP-0017-2740-3422	TA0005	T1620		TTP-0017-2740-3554; TTP-0017-2740-3996; TTP-0017-2744-5189
5	TTP-0017-2740-3554	TA0005	T1036.005		
6	TTP-0017-2740-3996	TA0007	T1518.001	T1047	TTP-0017-2740-4665; TTP-0017-2740-5434
7	TTP-0017-2740-4665	TA0003	T1547.001	T1218.011	TTP-0017-2765-8882
8	TTP-0017-2740-5434	TA0003	T1546.015	T1112	TTP-0017-2744-6446
9	TTP-0017-2744-5189	TA0003	T1136.001		TTP-0017-2744-5342
10	TTP-0017-2744-5342	TA0004	T1134.001		
11	TTP-0017-2744-6446	TA0005	T1574.001	T1218.011	TTP-0017-2765-8882
12	TTP-0017-2765-8882	TA0005	T1070.004		TTP-0017-2765-9210
13	TTP-0017-2765-9210	TA0005	T1027.011	T1027.013	TTP-0017-2769-9896
14	TTP-0017-2769-9435	TA0011	T1573.001		
15	TTP-0017-2769-9896	TA0011	T1102.003	T1573.001; T1071.001	TTP-0017-2770-0651
16	TTP-0017-2770-0651	TA0011	T1071.001	T1132.001; T1573.001	TTP-0017-2770-1458; TTP-0017-2770-1785; TTP-0017-2770-1793; TTP-0017-2770-1875
17	TTP-0017-2770-1458	TA0007	T1016		TTP-0017-2770-1958
18	TTP-0017-2770-1785	TA0007	T1082		TTP-0017-2770-1958
19	TTP-0017-2770-1793	TA0007	T1033		TTP-0017-2770-1958
20	TTP-0017-2770-1875	TA0007	T1057		TTP-0017-2770-1958
21	TTP-0017-2770-1958	TA0011	T1071.001		
22	TTP-0017-2770-4412	TA0011	T1573.001		TTP-0017-2770-1958
23	TTP-0017-2770-5684	TA0011	T1105		TTP-0017-2771-8516
24	TTP-0017-2771-8516	TA0002	T1106		
25	TTP-0017-2771-8918	TA0005	T1070.004		
26	TTP-0017-2771-9069	TA0010	T1041	T1030	
27	TTP-0017-2771-9252	TA0005	T1027.011		
28	TTP-0017-2771-9368	TA0011	T1059.003	T1559; T1041	
29	TTP-0017-2791-9294	TA0002	T1059.001		

*Códigos de acordo com a taxonomia do MITRE ATT&CK [18].

A lista completa dessas TTPs e seus atributos, referentes à classe `TTP` do modelo de dados, encontra-se no Apêndice A. Para fins ilustrativos, apresenta-se a seguir uma delas:

TTP ID: TTP-0017-2770-0651
Tactic: TA0011 (Command and Control)
Technique: T1071.001 (Application Layer Protocol: Web Protocols)
Procedure: `The requests to the C&C server always use the same URL scheme : https://[C&C server domain]/scripts/m/query.php?id=[base64(encrypted data)].` (···)
Secondary Techniques: T1132.001 (Data Encoding: Standard Encoding); T1573.001 (Encrypted Channel: Symmetric Cryptography)
Related TTPs: TTP-0017-2769-9435
References: REF-0017-2739-5256
Notes: Pages 20-21. C&C server communications and backdoor commands. Commander (main backdoor).
TTP Chain: TTP-0017-2770-1458; TTP-0017-2770-1785; TTP-0017-2770-1793; TTP-0017-2770-1875

As informações produzidas pela aplicação da metodologia proporcionam uma visão tática abrangente da ameaça, evidenciando suas TTPs e os relacionamentos entre elas. Para representar visualmente os resultados obtidos no primeiro estágio, são apresentados dois diagramas de TTPs (Figura 10.8 e Figura 10.9), construídos com base nos elementos cuja legenda é apresentada na Figura 10.7.

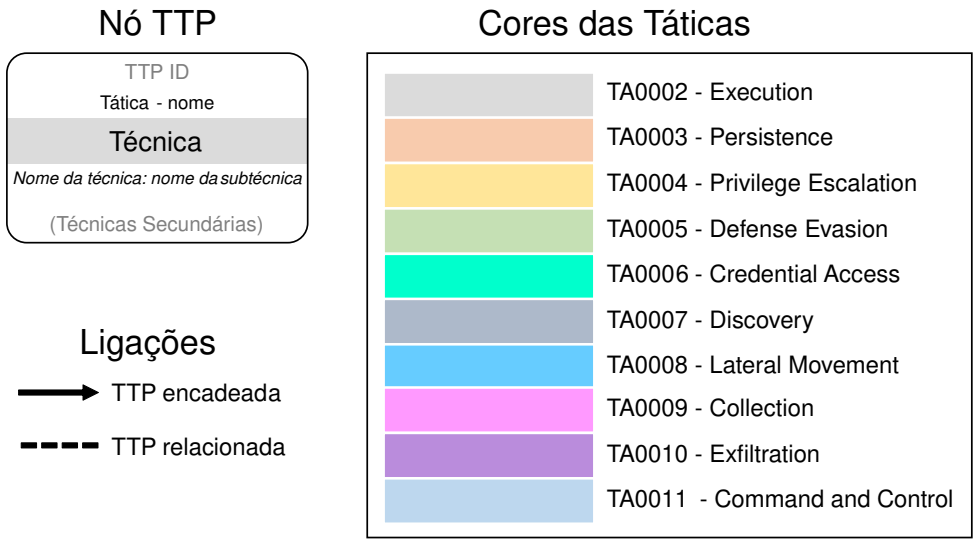


Figura 10.7: Legenda do diagrama de TTPs

A Figura 10.8 mostra o mapeamento das TTPs do módulo *Instalador* do Mosquito, enquanto a Figura 10.9 apresenta os módulos textitLançador e textitBackdoor principal.

Além de representar as Táticas e Técnicas (principais e secundárias) associadas a cada TTP, os diagramas ilustram as relações entre elas, proporcionando uma visão ampla da cadeia de comportamentos que caracterizam a execução desse *malware*. Essas informações são fundamentais para orientar o Threat Hunting baseado em TTPs, contribuindo significativamente para a formulação de estratégias mais eficazes de detecção.

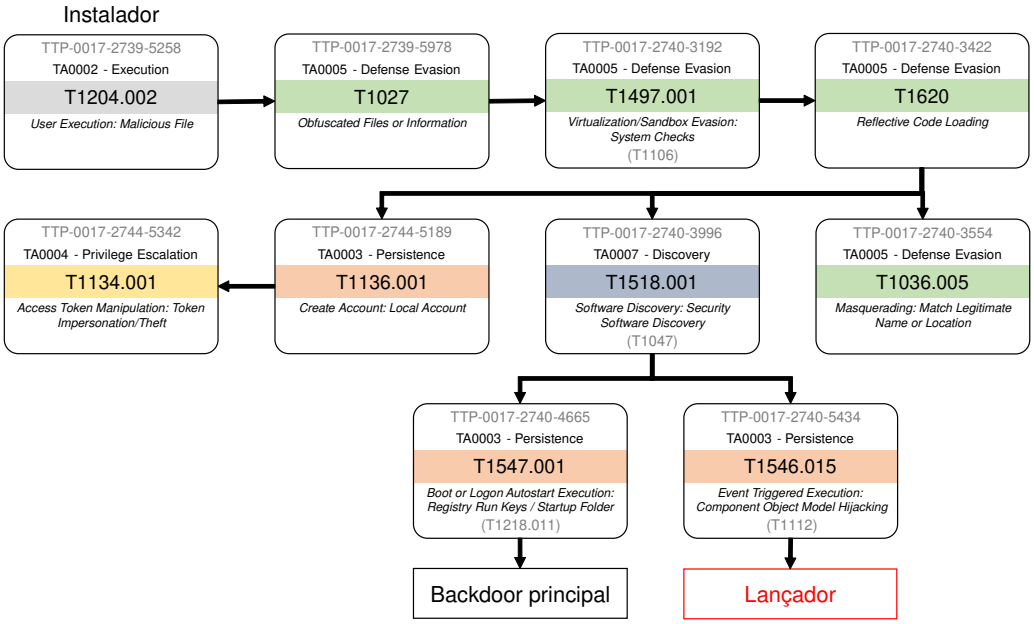


Figura 10.8: Diagrama de TTPs referentes ao Instalador do malware Mosquito

O Runbook gerado a partir dos dados deste estudo de caso, em formato YAML, está disponível em `https://github.com/pauloferraz80/Runbook_Creator_Editor/blob/main/runbooks/Mosquito_EN_stage1.yml`. O arquivo pode ser visualizado em qualquer editor de texto ou, preferencialmente, importado na ferramenta *Runbook Creator/Editor* [216].

10.3 Estudo de Caso 3: APT29

O APT29, também conhecido como Cozy Bear ou The Dukes, é um grupo de ameaças avançadas atribuído ao governo russo, em operação desde pelo menos 2008 [232]. Reconhecido por sua discrição e sofisticação técnica, o grupo emprega um extenso arsenal de *malwares* personalizados e adota táticas que priorizam a furtividade e a persistência.

O APT29 também se distingue por ajustar sua cadência operacional conforme os objetivos da missão, iniciando, quando necessário, campanhas rápidas de espionagem do tipo "smash-and-grab", voltadas à coleta e exfiltração imediata de dados. Quando identifica alvos de maior valor estratégico, o grupo muda para um conjunto distinto de ferramentas

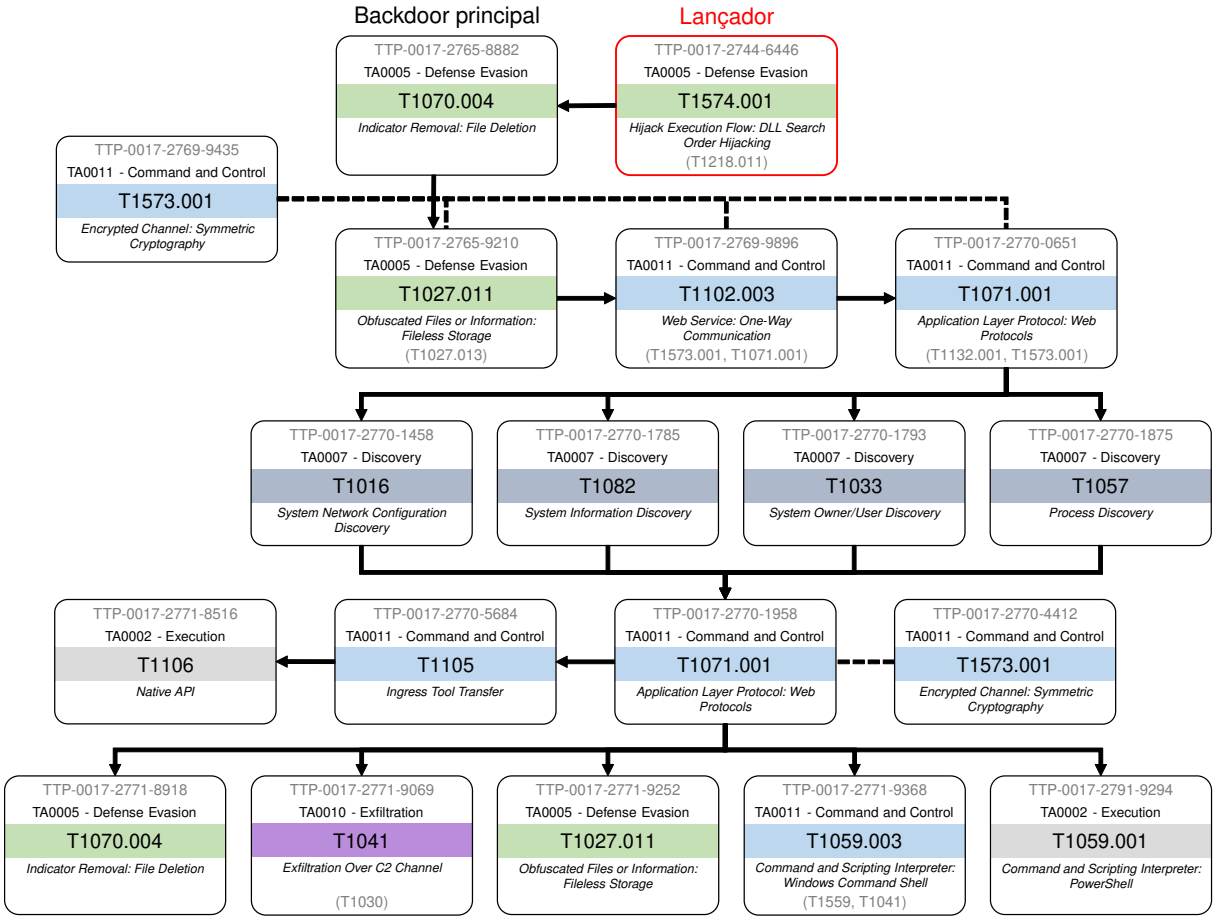


Figura 10.9: Diagrama de TTPs referentes ao Lançador e ao Backdoor principal do malware Mosquito

e adota técnicas mais furtivas, sustentando campanhas prolongadas e altamente direcionadas de coleta de inteligência [233].

Em 2020, o programa *MITRE ATT&CK Evaluations* [218] propôs dois cenários baseados em campanhas reais do APT29, com o objetivo de testar a eficácia de produtos de cibersegurança frente a ameaças avançadas, utilizando uma metodologia aberta baseada no framework ATT&CK. A iniciativa visa aprimorar a capacidade das organizações em detectar e responder a comportamentos adversários conhecidos [234]. Os dois cenários de emulação são descritos a seguir [235]:

Cenário 1: O primeiro cenário simula uma missão inicial de espionagem rápida do tipo "smash-and-grab" (arrombar e roubar), focada na coleta e exfiltração de dados, seguida da adoção de técnicas mais furtivas para obtenção de persistência, coleta adicional de informações, acesso a credenciais e movimento lateral. O cenário encerra-se com a execução dos mecanismos de persistência previamente implantados.

Cenário 2: O segundo cenário apresenta uma abordagem mais lenta e furtiva, na qual o grupo compromete o alvo inicial, estabelece persistência, coleta credenciais e, por fim, expande o comprometimento a todo o domínio. O cenário termina com uma simulação de passagem de tempo, que aciona os mecanismos de persistência estabelecidos.

Este estudo de caso aplica a metodologia proposta ao **Cenário 1**, cujo plano de emulação é composto por 10 etapas descritas a seguir [236]:

1. **Violação inicial:** O cenário inicia-se com a execução, por um usuário legítimo, de um payload disfarçado de documento do Word, mas que na verdade é um executável de protetor de tela. Ao ser executado, o payload estabelece uma conexão C2 pela porta 1234 usando a cifra criptográfica RC4. Em seguida, o atacante utiliza essa conexão ativa para gerar instâncias interativas de `cmd.exe` e `powershell.exe`.
2. **Coleta e exfiltração rápidas:** O invasor executa um comando de linha única para pesquisar o sistema de arquivos em busca de documentos e arquivos de mídia, compactando o conteúdo coletado em um único arquivo, posteriormente exfiltrado pela conexão C2 existente.
3. **Implantação do kit de ferramentas furtivas:** Um novo payload é carregado na máquina comprometida por meio de um arquivo de imagem legítimo contendo um script PowerShell embutido. O atacante eleva privilégios explorando um desvio do Controle de Conta de Usuário (User Account Control – UAC), o que aciona o script e estabelece uma nova conexão C2 via HTTPS na porta 443. Em seguida, remove os artefatos do aumento de privilégio do Registro.
4. **Evasão de defesa e descoberta:** O invasor carrega ferramentas adicionais e inicia um *shell* interativo do PowerShell. Ele enumera processos em execução para encerrar o acesso inicial e apaga arquivos associados a esse vetor. Por fim, executa um script de reconhecimento que utiliza a API do Windows para coletar informações e estabelece dois mecanismos de persistência: criação de um novo serviço e adição de um payload malicioso na pasta de inicialização do Windows.
5. **Persistência:** O atacante mantém dois métodos de persistência: um serviço malicioso e um payload na pasta de inicialização.
6. **Acesso a credenciais:** O invasor utiliza uma ferramenta renomeada para se disfarçar de utilitário legítimo e extrai credenciais armazenadas em navegadores locais, além de coletar chaves privadas e hashes de senha.
7. **Coleta e exfiltração:** O invasor captura capturas de tela, dados da área de transferência e pressionamentos de teclas, além de coletar e criptografar arquivos antes de exfiltrá-los para um compartilhamento WebDAV controlado.

8. **Movimento lateral:** Por meio de consultas LDAP, o atacante enumera outros hosts do domínio e estabelece uma sessão remota do PowerShell em uma segunda vítima. A partir dela, envia e executa um novo payload empacotado com UPX via `PSExec`, utilizando as credenciais roubadas anteriormente.
9. **Coleta:** O invasor faz upload de utilitários adicionais para a vítima secundária, executa comandos do PowerShell para localizar documentos e arquivos de mídia, criptografa e compacta os resultados e exfiltra os dados pela conexão C2 ativa. Por fim, remove os artefatos relacionados.
10. **Execução de persistência:** A máquina original é reinicializada e o usuário legítimo faz login, simulando o uso cotidiano. Essa ação aciona os mecanismos de persistência implantados, executando novamente o serviço malicioso e o payload na pasta de inicialização, o qual utiliza um token roubado para acionar cargas subsequentes.

Estágio 1: Mapeamento Ameaça–TTP

No primeiro estágio da metodologia foi realizado o mapeamento Ameaça–TTP aplicado ao cenário descrito. As informações gerais da ameaça e as referências utilizadas, correspondentes aos atributos das classes `Threat` e `Reference` do modelo de dados, encontram-se no Apêndice B.

Como resultado, foram identificadas e documentadas **45 TTPs** associadas ao APT29. A Tabela 10.5 apresenta um resumo com as principais informações obtidas no mapeamento. Para ilustrar visualmente os relacionamentos e encadeamentos entre as TTPs, são apresentados dois diagramas: a Figura 10.10, correspondente às etapas 1 a 5 do plano de emulação, e a Figura 10.11, referente às etapas 6 a 10. Ambos os diagramas foram elaborados seguindo a padronização descrita na Figura 10.7.

A lista completa dessas TTPs e seus atributos, referentes à classe `TTP` do modelo de dados, encontra-se no Apêndice C. Para fins ilustrativos, apresenta-se a seguir uma delas:

TTP ID: TTP-0017-3068-5003
Tactic: TA0002 (Execution)
Technique: T1204.002 (User Execution: Malicious File)
Procedure: `Um usuário legítimo clica em um payload executável (tipo sc⌟ reensaver executável) mascarado de documento Word benigno por meio da técnica "Right-to-Left Override" (RTLO) que usa o caractere U+202E no nome do arquivo para invertê-lo.`
Secondary Techniques: T1036.002 (Masquerading: Right-to-Left Override); T1573.001 (Encrypted Channel: Symmetric Cryptography)

Tabela 10.5: Resumo das principais informações das TTPs mapeadas do APT29

Nr	TTP ID	Tática*	Técnica*	Técnicas secundárias*	TTP seguinte (encadeamento ou TTP Chain)
1	TTP-0017-3068-5003	TA0002	T1204.002	T1036.002	TTP-0017-6044-2942
2	TTP-0017-6044-2942	TA0011	T1573.001	T1571	TTP-0017-6044-6572
3	TTP-0017-6044-6572	TA0002	T1059.001	T1059.003	TTP-0017-6044-6947
4	TTP-0017-6044-6947	TA0007	T1083	T1005	TTP-0017-6044-7344
5	TTP-0017-6044-7344	TA0009	T1560.001	T1074.001	TTP-0017-6044-7520
6	TTP-0017-6044-7520	TA0010	T1041		TTP-0017-6044-7700
7	TTP-0017-6044-7700	TA0011	T1105	T1027.003	TTP-0017-6045-0689
8	TTP-0017-6045-0689	TA0004	T1548.002	T1546.015	TTP-0017-6045-2068
9	TTP-0017-6045-2068	TA0011	T1071.001	T1573; T1043	TTP-0017-6045-2305
10	TTP-0017-6045-2305	TA0005	T1112		TTP-0017-6045-2548
11	TTP-0017-6045-2548	TA0011	T1105		TTP-0017-6045-8979
12	TTP-0017-6045-8979	TA0005	T1140	T1059.001	TTP-0017-6045-9266
13	TTP-0017-6045-9266	TA0007	T1057		TTP-0017-6046-1439
14	TTP-0017-6046-1439	TA0005	T1070.004		TTP-0017-6046-1657; TTP-0017-6046-2409; TTP-0017-6046-2437; TTP-0017-6046-2449; TTP-0017-6046-2450; TTP-0017-6046-2453; TTP-0017-6046-2456
15	TTP-0017-6046-1657	TA0007	T1083	T1059.001	TTP-0017-6046-4431; TTP-0017-6046-4432
16	TTP-0017-6046-2409	TA0007	T1033	T1059.001	TTP-0017-6046-4431; TTP-0017-6046-4432
17	TTP-0017-6046-2437	TA0007	T1082	T1059.001	TTP-0017-6046-4431; TTP-0017-6046-4432
18	TTP-0017-6046-2449	TA0007	T1016	T1059.001	TTP-0017-6046-4431; TTP-0017-6046-4432
19	TTP-0017-6046-2450	TA0007	T1057	T1059.001	TTP-0017-6046-4431; TTP-0017-6046-4432
20	TTP-0017-6046-2453	TA0007	T1518.001	T1059.001	TTP-0017-6046-4431; TTP-0017-6046-4432
21	TTP-0017-6046-2456	TA0007	T1069	T1106; T1059.001	TTP-0017-6046-4431; TTP-0017-6046-4432
22	TTP-0017-6046-4431	TA0003	T1543.003		TTP-0017-6046-5407
23	TTP-0017-6046-4432	TA0003	T1547.001		TTP-0017-6046-5407
24	TTP-0017-6046-5407	TA0006	T1555.003	T1552.001; T1036.005	TTP-0017-6046-6464; TTP-0017-6046-7568
25	TTP-0017-6046-5816	TA0005	T1036.005		
26	TTP-0017-6046-6464	TA0006	T1552.004		TTP-0017-6049-5413; TTP-0017-6049-7247; TTP-0017-6049-7350; TTP-0017-6049-7604
27	TTP-0017-6046-7568	TA0006	T1003.002		TTP-0017-6049-5413; TTP-0017-6049-7247; TTP-0017-6049-7350; TTP-0017-6049-7604
28	TTP-0017-6049-5413	TA0009	T1113		TTP-0017-6049-8712
29	TTP-0017-6049-7247	TA0009	T1115		TTP-0017-6049-8712
30	TTP-0017-6049-7350	TA0009	T1056.001		TTP-0017-6049-8712
31	TTP-0017-6049-7604	TA0009	T1005	T1560	TTP-0017-6049-8712
32	TTP-0017-6049-8462	TA0009	T1560		
33	TTP-0017-6049-8712	TA0010	T1048		TTP-0017-6049-9015
34	TTP-0017-6049-9015	TA0007	T1018		TTP-0017-6049-9628
35	TTP-0017-6049-9628	TA0008	T1021.006		TTP-0017-6049-9772
36	TTP-0017-6049-9772	TA0007	T1057		TTP-0017-6049-9887
37	TTP-0017-6049-9887	TA0011	T1105	T1027.002	TTP-0017-6050-0048
38	TTP-0017-6050-0048	TA0002	T1569.002	T1021.002; T1078.002	TTP-0017-6050-0265
39	TTP-0017-6050-0265	TA0011	T1105		TTP-0017-6053-3303
40	TTP-0017-6053-3303	TA0007	T1083	T1059.001	TTP-0017-6053-5026
41	TTP-0017-6053-5026	TA0009	T1560.001	T1005; T1074.001	TTP-0017-6053-5295
42	TTP-0017-6053-5295	TA0010	T1041		TTP-0017-6053-5447
43	TTP-0017-6053-5447	TA0005	T1070.004		
44	TTP-0017-6053-5618	TA0002	T1569.002		TTP-0017-6046-5407
45	TTP-0017-6053-5786	TA0003	T1547.002	T1134.002	TTP-0017-6046-5407

*Códigos de acordo com a taxonomia do MITRE ATT&CK [18].

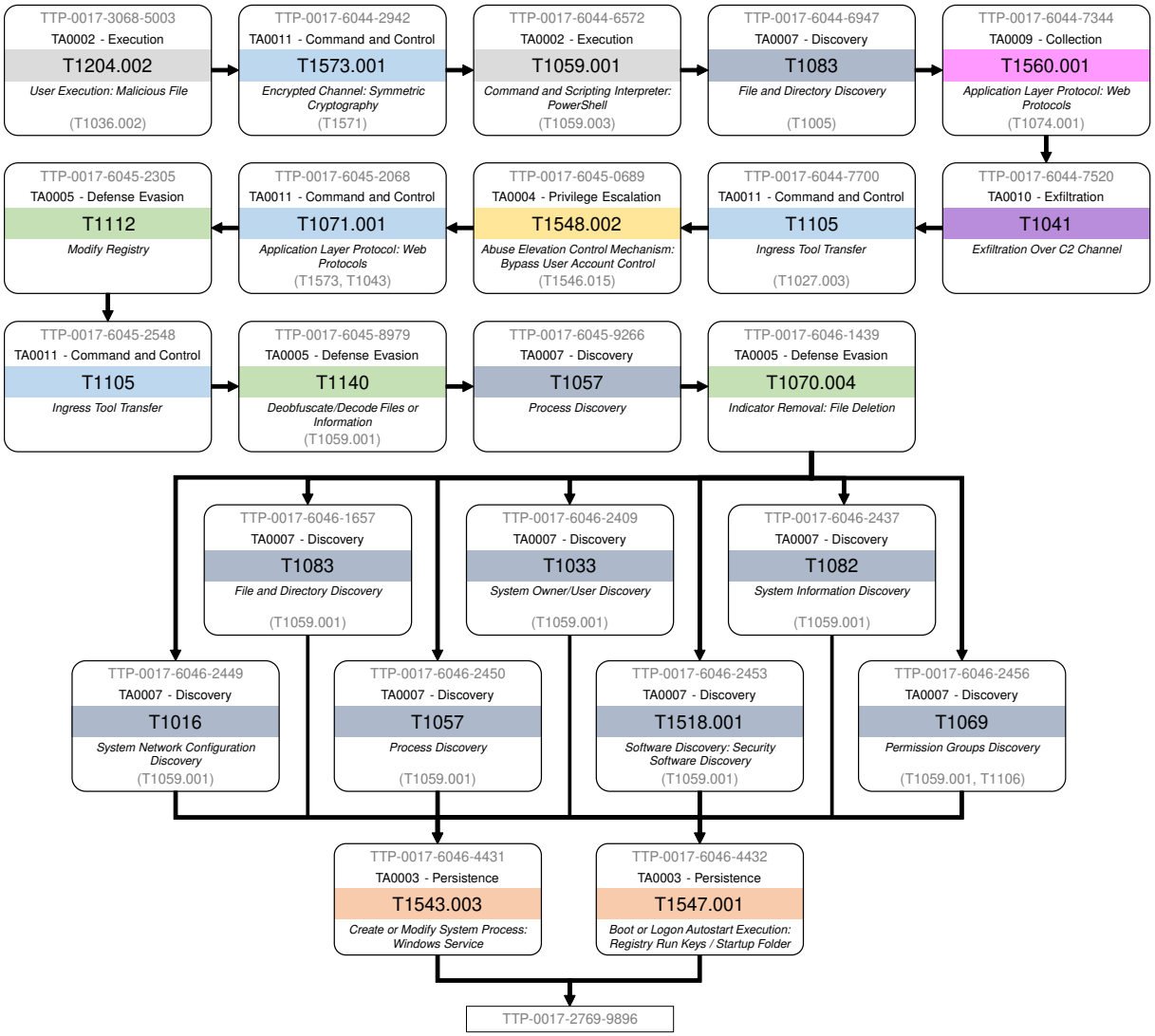


Figura 10.10: Diagrama de TTPs referentes às etapas 1 a 5 do APT29

Related TTPs:
References: REF-0017-3068-5073
Notes:
TTP Chain: TTP-0017-6044-2942

Estágio 2: Mapeamento TTP–Dados

No segundo estágio da metodologia foram desenvolvidas **38 Regras de Detecção**, conforme apresentadas na Tabela 10.6. Essas regras foram derivadas a partir das TTPs identificadas no estágio anterior (TTPs de referência) e estruturadas segundo a classe `DetectionRule` do modelo de dados proposto.

A lista completa das Regras de Detecção, com seus respectivos atributos e metadados, encontra-se no Apêndice D. Para fins ilustrativos, apresenta-se a seguir uma dessas regras:

Tabela 10.6: Metadados resumidos das Regras de Detecção desenvolvidas para o APT29

Nr	ID da regra	TTP de referência	Técnicas cobertas*	Fontes de coleta
1	DTR-0017-3103-4080	TTP-0017-3068-5003	T1204.002; T1036.002	Sysmon
2	DTR-0017-6092-4557	TTP-0017-6044-2942	T1571	Sysmon
3	DTR-0017-6092-7552	TTP-0017-6044-6572	T1059.001	Sysmon
4	DTR-0017-6099-4535	TTP-0017-6044-6947	T1083; T1059.001	Sysmon, PowerShell
5	DTR-0017-6099-9496		T1560.001; T1059.001	Sysmon, PowerShell
6	DTR-0017-6124-7922	TTP-0017-6044-7344	T1560.001; T7074.001; T1560	Sysmon
7	DTR-0017-6100-1386	TTP-0017-6044-7700	T1105	Sysmon
8	DTR-0017-6100-4576	TTP-0017-6045-0689	T1546.015	Sysmon
9	DTR-0017-6100-6059		T1548.002	Sysmon
10	DTR-0017-6104-7163	TTP-0017-6045-2305	T1112	Sysmon
11	DTR-0017-6104-8503	TTP-0017-6045-2548	T1105	Sysmon
12	DTR-0017-6104-9721	TTP-0017-6045-8979	T1140; T1059.001	Sysmon, PowerShell
13	DTR-0017-6105-0296	TTP-0017-6045-9266	T1057; T1059.001	Sysmon, PowerShell
14	DTR-0017-6105-0810	TTP-0017-6046-1439	T1070.003	Sysmon
15	DTR-0017-6105-2497	TTP-0017-6046-1657	T1083; T1059.001	Sysmon, PowerShell
16	DTR-0017-6105-3592	TTP-0017-6046-2409	T1033; T1059.001	Sysmon, PowerShell
17	DTR-0017-6105-5552	TTP-0017-6046-2437	T1082; T1059.001	Sysmon, PowerShell
18	DTR-0017-6105-6366	TTP-0017-6046-2449	T1016; T1059.001	Sysmon, PowerShell
19	DTR-0017-6105-6754	TTP-0017-6046-2450	T1057; T1059.001	Sysmon, PowerShell
20	DTR-0017-6105-7294	TTP-0017-6046-2453	T1518.001; T1059.001	Sysmon, PowerShell
21	DTR-0017-6106-2801	TTP-0017-6046-2456	T1069; T1059.001; T1106	Sysmon, PowerShell
22	DTR-0017-6106-4244	TTP-0017-6046-4431	T1543.003; T1059.001	Sysmon, PowerShell
23	DTR-0017-6106-5518	TTP-0017-6046-4432	T1547.001	Sysmon
24	DTR-0017-6106-5995	TTP-0017-6046-5816	T1036.005	Sysmon
25	DTR-0017-6106-6816	TTP-0017-6049-5413	T1113; T1106	Sysmon
26	DTR-0017-6106-7870		T1113; T1059.001	Sysmon, PowerShell
27	DTR-0017-6106-7534	TTP-0017-6049-7247	T1115; T1059.001	Sysmon, PowerShell
28	DTR-0017-6106-8933	TTP-0017-6049-8462	T1560; T1059.001	Sysmon, PowerShell
29	DTR-0017-6106-9592	TTP-0017-6049-8712	T1048; T1059.001	Sysmon
30	DTR-0017-6107-0164	TTP-0017-6049-9015	T1018; T1059.001	Sysmon
31	DTR-0017-6107-0843	TTP-0017-6049-9628	T1021.006	Sysmon
32	DTR-0017-6107-2091	TTP-0017-6049-9772	T1057; T1059.001	Sysmon, PowerShell
33	DTR-0017-6107-2726	TTP-0017-6050-0048	T1569.002	Windows Security Auditing
34	DTR-0017-6107-3395	TTP-0017-6050-0265	T1105	Sysmon
35	DTR-0017-6107-4117	TTP-0017-6053-3303	T1083; T1059.001	Sysmon, PowerShell
36	DTR-0017-6107-4769	TTP-0017-6053-5026	T1560.001	Sysmon
37	DTR-0017-6107-5368	TTP-0017-6053-5447	T1070.004	Sysmon
38	DTR-0017-6107-5740	TTP-0017-6053-5618	T1569.002	Sysmon

*Códigos de acordo com a taxonomia do MITRE ATT&CK [18].

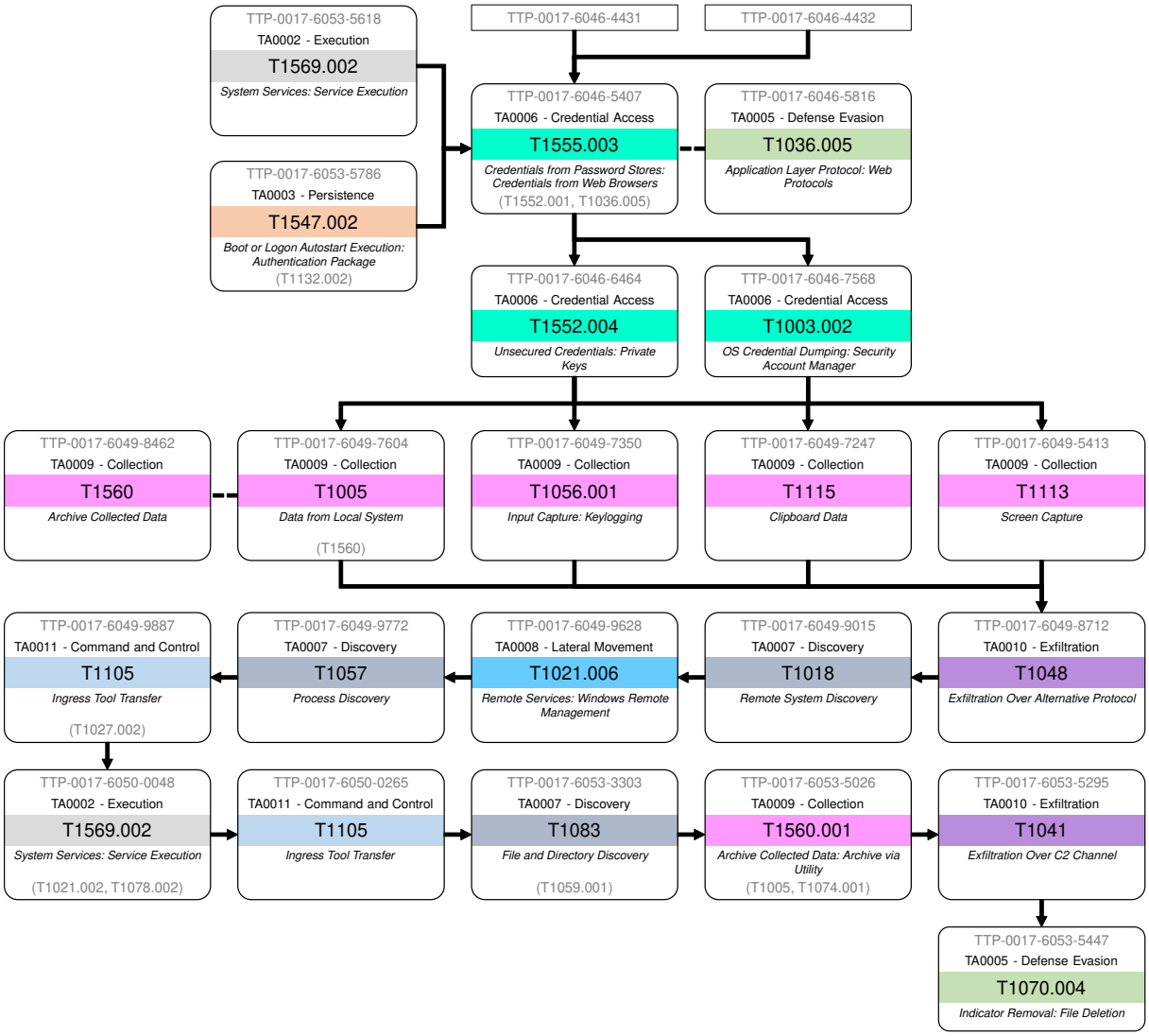


Figura 10.11: Diagrama de TTPs referentes às etapas 6 a 10 do APT29

Rule ID: DTR-0017-3103-4080

Creation Date: 2024-11-07

Update Date: 2025-10-25

Reference TTP: TTP-0017-3068-5003

Description: `Detecta a execução de um arquivo suspeito com extensão '.scr' e nome contendo 'cod.', indicando uma possível tentativa de utilização da técnica RTLO (Right-to-Left Override) para disfarçar um arquivo executável como um '.doc'.`

Platforms: Windows

Covered Techniques: T1204.002 (User Execution: Malicious File); T1036.002 (Masquerading: Right-to-Left Override)

Sources: Sysmon

Language: SparkSQL

Query:
```
SELECT `@timestamp`,UtcTime,Message
FROM apt29
WHERE Channel = "Microsoft-Windows-Sysmon/Operational"
AND EventID = 1
AND LOWER(ParentImage) LIKE "%explorer.exe"
AND LOWER(Image) LIKE '%cod.%'
AND LOWER(Image) LIKE '%.scr'
```

Estágio 3: Validação

A validação das regras foi realizada utilizando o dataset malicioso disponibilizado pelo projeto *Mordor* [237], contendo logs de eventos do Windows e Sysmon coletados durante a emulação do APT29 (Cenário 1) [238, 236], conforme o plano de execução proposto pela *MITRE ATT&CK Evaluations* [218]. Esse dataset foi produzido durante a *Detection Hackathon 2020*, um evento colaborativo voltado à criação de conjuntos de dados realistas para experimentação e validação de técnicas de detecção.

A execução das regras foi realizada no *Apache Spark* [209], por meio do ambiente *Jupyter Notebook*, conforme esquema ilustrado na Figura 10.1. A inicialização da sessão Spark e o carregamento do dataset são apresentados na Figura 10.12.

```
#Iniciando a sessão Spark
from pyspark.sql import SparkSession
spark = SparkSession.builder.getOrCreate()
spark.conf.set("spark.sql.caseSensitive", "true")

#Carregando o Dataset
df = spark.read.json('apt29_evals_day1_manual_2020-05-01225525.json')
df.createTempView('apt29')
```

Figura 10.12: Inicialização da sessão Spark e carregamento do dataset do APT29

A Figura 10.13 apresenta um exemplo do teste da regra DTR-0017-3103-4080 no ambiente Jupyter. O arquivo completo contendo todos os testes realizados encontra-se disponível em `https://github.com/pauloferraz80/Runbook_Creator_Editor/blob/main/validation/APT29cenario1_PT_SparkSQL_validation.ipynb`.

A Tabela 10.7 apresenta o resumo da avaliação das regras aplicadas ao dataset malicioso, incluindo a quantidade de ocorrências (*hits*) identificadas por cada uma e o instante em que o primeiro evento correspondente foi detectado. Esses tempos foram empregados para verificar a coerência da sequência de detecção em relação ao encadeamento de TTPs (*TTP Chain*) identificado no primeiro estágio da metodologia.

Como resultado, todas as Regras de Detecção foram validadas com sucesso, e a ordem cronológica das detecções confirmou o encadeamento previsto das TTPs mapeadas para

Tabela 10.7: Resumo dos testes de validação das regras de detecção do APT29 no dataset malicioso

Nr	Regra de detecção	TTP de referência	Sequência dada pela TTP Chain	Hits	Detectou?	Tempo do evento* (primeiro hit)	TTP Chain correta?
1	DTR-0017-3103-4080	TTP-0017-3068-5003	1	1	Sim	2020-05-02 02:55:56.157	Sim
2	DTR-0017-6092-4557	TTP-0017-6044-2942	2	1	Sim	2020-05-02 02:55:59.631	Sim
3	DTR-0017-6092-7552	TTP-0017-6044-6572	3	3	Sim	2020-05-02 02:56:14.894	Sim
4	DTR-0017-6099-4535	TTP-0017-6044-6947	4	1	Sim	2020-05-02 02:56:17	Sim
5	DTR-0017-6099-9496	TTP-0017-6044-7344	5	1	Sim	2020-05-02 02:56:17	Sim
6	DTR-0017-6124-7922			1	Sim	2020-05-02 02:56:18.032	Sim
7	DTR-0017-6100-1386	TTP-0017-6044-7700	6	1	Sim	2020-05-02 02:57:00.933	Sim
8	DTR-0017-6100-4576	TTP-0017-6045-0689	7	1	Sim	2020-05-02 02:58:30.649	Sim
9	DTR-0017-6100-6059			1	Sim	2020-05-02 02:58:43.008	Sim
10	DTR-0017-6104-7163	TTP-0017-6045-2305	8	1	Sim	2020-05-02 02:59:15.911	Sim
11	DTR-0017-6104-8503	TTP-0017-6045-2548	9	1	Sim	2020-05-02 02:59:42.544	Sim
12	DTR-0017-6104-9721	TTP-0017-6045-8979	10	1	Sim	2020-05-02 03:00:33	Sim
13	DTR-0017-6105-0296	TTP-0017-6045-9266	11	4	Sim	2020-05-02 03:01:03	Sim
14	DTR-0017-6105-0810	TTP-0017-6046-1439	12	2	Sim	2020-05-02 03:03:37.794	Sim
15	DTR-0017-6105-2497	TTP-0017-6046-1657	13	1	Sim	2020-05-02 03:04:03.	Sim
16	DTR-0017-6105-3592	TTP-0017-6046-2409		2	Sim	2020-05-02 03:04:03	Sim
17	DTR-0017-6105-5552	TTP-0017-6046-2437		1	Sim	2020-05-02 03:04:03	Sim
18	DTR-0017-6105-6366	TTP-0017-6046-2449		1	Sim	2020-05-02 03:04:03	Sim
19	DTR-0017-6105-6754	TTP-0017-6046-2450		3	Sim	2020-05-02 03:04:03	Sim
20	DTR-0017-6105-7294	TTP-0017-6046-2453		1	Sim	2020-05-02 03:04:03	Sim
21	DTR-0017-6106-2801	TTP-0017-6046-2456		1	Sim	2020-05-02 03:04:03	Sim
22	DTR-0017-6106-4244	TTP-0017-6046-4431	14	1	Sim	2020-05-02 03:04:15	Sim
23	DTR-0017-6106-5518	TTP-0017-6046-4432		1	Sim	2020-05-02 03:04:23.681	Sim
24	DTR-0017-6106-5995	TTP-0017-6046-5816	15	1	Sim	2020-05-02 03:04:34.959	Sim
25	DTR-0017-6106-6816	TTP-0017-6049-5413	16	1	Sim	2020-05-02 03:06:42.583	Sim
26	DTR-0017-6106-7870			1	Sim	2020-05-02 03:05:58.	Sim
27	DTR-0017-6106-7534	TTP-0017-6049-7247		1	Sim	2020-05-02 03:06:42	Sim
28	DTR-0017-6106-8933	TTP-0017-6049-8462		1	Sim	2020-05-02 03:08:35	Sim
29	DTR-0017-6106-9592	TTP-0017-6049-8712	17	5	Sim	2020-05-02 03:08:50.846	Sim
30	DTR-0017-6107-0164	TTP-0017-6049-9015	18	2	Sim	2020-05-02 03:09:04.482	Sim
31	DTR-0017-6107-0843	TTP-0017-6049-9628	19	4	Sim	2020-05-02 03:09:27.510	Sim
32	DTR-0017-6107-2091	TTP-0017-6049-9772	20	1	Sim	2020-05-02 03:09:28	Sim
33	DTR-0017-6107-2726	TTP-0017-6050-0048	21	16	Sim	2020-05-02 03:11:40	Sim
34	DTR-0017-6107-3395	TTP-0017-6050-0265	22	2	Sim	2020-05-02 03:15:31.530	Sim
35	DTR-0017-6107-4117	TTP-0017-6053-3303	23	1	Sim	2020-05-02 03:15:59	Sim
36	DTR-0017-6107-4769	TTP-0017-6053-5026	24	1	Sim	2020-05-02 03:16:19.395	Sim
37	DTR-0017-6107-5368	TTP-0017-6053-5447	25	3	Sim	2020-05-02 03:16:52.587	Sim
38	DTR-0017-6107-5740	TTP-0017-6053-5618	26	1	Sim	2020-05-02 03:19:14.118	Sim

*Tempo no padrão UTC.

```
df = spark.sql(
'''
SELECT `@timestamp`,UtcTime,Message
FROM apt29
WHERE Channel = "Microsoft-Windows-Sysmon/Operational"
    AND EventID = 1
    AND LOWER(ParentImage) LIKE "%explorer.exe"
    AND LOWER(Image) LIKE '%cod.%'
    AND LOWER(Image) LIKE '%.scr'
'''
)
df.show(100,truncate = False, vertical = True)
```

```
-RECORD
0-----------------------------------------------------------------------------------------------------------------------------------
-----------------------------------------------------------------------------------------------------------------------------------
-----------------------------------------------------------------------------------------------------------------------------------
-----------------------------------------------------------------------------------------------------------------------------------
-----------------------------------------------------------------------------------------------------------------------------------
-----------------------------------------------------------------------------------------------------------------------------------
-----------------------------------------------------------------------------------------------------------------------------------
-----------------------------------------------------------
 @timestamp | 2020-05-02T02:55:57.730Z
 UtcTime    | 2020-05-02 02:55:56.157
 Message    | Process Create:
RuleName: -
UtcTime: 2020-05-02 02:55:56.157
ProcessGuid: {47ab858c-e13c-5eac-a903-000000000400}
ProcessId: 8524
Image: C:\ProgramData\victim\â€®cod.3aka3.scr
FileVersion: -
Description: -
Product: -
Company: -
OriginalFileName: -
CommandLine: "C:\ProgramData\victim\â€®cod.3aka3.scr" /S
CurrentDirectory: C:\ProgramData\victim\
User: DMEVALS\pbeesly
LogonGuid: {47ab858c-dabe-5eac-f331-370000000000}
LogonId: 0x3731F3
TerminalSessionId: 2
IntegrityLevel: Medium
Hashes: SHA1=4B7FA56A4E85F88B98D11A6E018698AE3FBA5E62,MD5=9D1C5EF38E6073661C74660B3A71A76E,SHA256=0DF38A55D940F49847
8EB03683C94D4584236E100125B526A67650BA54DF4AE4,IMPHASH=F00447512A354E59D39D2818AABA4A17
ParentProcessGuid: {47ab858c-dac4-5eac-f202-000000000400}
ParentProcessId: 4440
ParentImage: C:\Windows\explorer.exe
ParentCommandLine: C:\windows\Explorer.EXE
```

Figura 10.13: Validação da Regra de Detecção DTR-0017-3103-4080

o APT29, reforçando a consistência e a efetividade da metodologia proposta. Para fins ilustrativos, apresenta-se a seguir um exemplo das informações de validação registradas em uma Regra de Detecção.

Status: validated
Update Date: 2025-10-19
Dataset: `https://github.com/OTRF/detection-hackathon-apt29/raw/refs/heads/master/datasets/day1/apt29_evals_day1_manual.zip`
References: REF-0017-3103-4789; REF-0017-6044-5046
Notes:

Estágio 4: Consolidação dos Resultados

No último estágio da metodologia, as informações foram registradas e consolidadas por meio da ferramenta *Runbook Creator/Editor* [216], que também foi utilizada para a gera-

ção do Runbook de Threat Hunting no formato YAML. O arquivo resultante encontra-se disponível em `https://github.com/pauloferraz80/Runbook_Creator_Editor/blob/main/runbooks/APT29cenario1_PT_SparkSQL.yml`.

Capítulo 11

Avaliação e Discussão

Este capítulo apresenta uma avaliação detalhada da metodologia proposta no Capítulo 8, verificando sua eficácia e corretude com base nos resultados obtidos na Prova de Conceito (PoC) descrita no Capítulo 10. Adicionalmente, o modelo de dados introduzido no Capítulo 7 é avaliado quanto à sua adequação para a aplicação pretendida. Por fim, são discutidos os principais benefícios e apresentadas as limitações da metodologia.

11.1 Avaliação da Metodologia

A avaliação da metodologia foi conduzida com base em dois critérios fundamentais: **eficácia** e **corretude**. A eficácia refere-se à capacidade da metodologia de atingir seus objetivos propostos, enquanto a corretude está relacionada à precisão, consistência e confiabilidade dos resultados gerados. A análise conjunta desses critérios assegura que, além de alcançar seus propósitos, a metodologia produza resultados tecnicamente sólidos e de alta qualidade.

Uma metodologia pode ser eficaz ao cumprir seus objetivos, mas sem corretude, os resultados podem ser inconsistentes ou imprecisos, comprometendo sua aplicabilidade e credibilidade. Assim, avaliar a corretude é essencial para garantir que os procedimentos adotados sigam padrões bem definidos e produzam informações precisas e verificáveis. Enquanto a eficácia se concentra nos resultados alcançados, a corretude foca na exatidão dos procedimentos que levaram a esses resultados.

11.1.1 Eficácia

A eficácia da metodologia foi avaliada a partir da análise dos resultados obtidos em cada um dos quatro estágios. Essa avaliação buscou verificar a capacidade da metodologia em

mapear TTPs, desenvolver regras de detecção, validar essas regras e produzir Runbooks conforme o modelo de dados proposto.

No primeiro estudo de caso, a metodologia demonstrou sua eficácia ao desenvolver quatro regras de detecção de diferentes níveis de complexidade, utilizando lógicas que correlacionam múltiplos campos provenientes de distintas fontes de dados — Sysmon e Sistema de Auditoria do Windows. O processo de validação das *queries*, conduzido no ambiente *Jupyter Notebook* com o *Apache Spark* e ilustrado na Figura 10.1, reforçou a eficácia da abordagem, evidenciando sua capacidade de validar regras de detecção em datasets de forma precisa e consistente.

A eficácia da metodologia em proporcionar uma compreensão detalhada dos comportamentos adversários ficou evidente nos diagramas apresentados nas Figuras 10.8 e 10.9, que ilustram o mapeamento estruturado das TTPs e seus relacionamentos, conforme a legenda da Figura 10.7. Esses diagramas evidenciam o potencial da metodologia em produzir e organizar informações de forma integrada, oferecendo uma visão comportamental abrangente — elemento essencial para a compreensão e análise de ameaças complexas.

O terceiro estudo de caso reafirmou a eficácia da metodologia em um cenário de maior complexidade, aplicado a uma ameaça persistente avançada (APT). Esse tipo de ataque, de natureza multifásica e furtiva, é reconhecidamente difícil de detectar por abordagens tradicionais. Nesse estudo de caso, todos os quatro estágios da metodologia foram executados com sucesso, resultando no mapeamento de 45 TTPs e no desenvolvimento e validação de 38 Regras de Detecção.

A utilização da ferramenta *Runbook Creator/Editor* [216] nos três estudos de caso garantiu a eficácia na consolidação e registro das informações, possibilitando a geração de Runbooks de Threat Hunting padronizados, reutilizáveis e evolutivos.

Em síntese, a PoC comprovou não apenas a eficácia de cada estágio da metodologia, mas também sua capacidade de gerar Runbooks acionáveis, capazes de servir como fonte estruturada de inteligência tática para fortalecer as operações de Threat Hunting baseadas em TTPs.

11.1.2 Corretude

A corretude da metodologia foi avaliada sob dois aspectos principais: (i) a aderência das TTPs mapeadas às técnicas documentadas em fontes reconhecidas, como o MITRE ATT&CK [18]; e (ii) a precisão das regras de detecção na identificação de comportamentos adversários em dados coletados, em conformidade com a estratégia de detecção adotada.

No primeiro estudo de caso, a corretude das regras de detecção foi comprovada por meio do processo de validação. A análise dos resultados demonstrou que as *queries* desenvolvidas foram capazes de identificar elementos específicos nos dados coletados, caracteri-

zando de forma precisa os comportamentos adversários associados à TTP mapeada. Para eliminar a possibilidade de falsos positivos e, assim, garantir a confiabilidade dos resultados, as consultas foram realizadas em dataset contendo dados de emulação da ameaça em ambiente controlado.

No segundo estudo de caso, a metodologia demonstrou alta capacidade de gerar mapeamentos estruturados e consistentes. A corretude foi validada pela comparação das técnicas identificadas com a base de conhecimento do MITRE ATT&CK [18]. Inicialmente, confirmou-se a precisão das TTPs principais; em seguida, técnicas adicionais foram verificadas por meio de inferência e análise comparativa das descrições oficiais e dos exemplos de procedimentos registrados na base ATT&CK. Esse processo confirmou a precisão e o detalhamento alcançado pela metodologia.

O terceiro estudo de caso ampliou o escopo da validação, permitindo avaliar simultaneamente a corretude das regras de detecção e a consistência do encadeamento das TTPs. As regras associadas a diferentes TTPs foram executadas sobre um mesmo conjunto de dados, possibilitando a verificação da cronologia das detecções em conformidade com o encadeamento das TTPs (*TTP Chain*) identificado no primeiro estágio. Os resultados confirmaram tanto a precisão das regras quanto a coerência do encadeamento, assegurando a corretude global da metodologia.

11.2 Avaliação do Modelo de Dados

A avaliação do modelo de dados teve como objetivo verificar a adequação de sua estrutura para atender às necessidades da metodologia na criação de Runbooks padronizados e eficazes. Os resultados dos estudos de caso realizados durante a PoC confirmaram que o modelo de dados fornece uma estrutura clara, modular e eficiente para organizar e correlacionar informações.

A definição de classes especializadas foi essencial para estruturar o conhecimento de forma organizada, favorecendo a integração entre ameaças, TTPs e Regras de Detecção. A adoção de identificadores únicos para entidades-chave (Ameaça, TTP e Regra de Detecção) facilitou a correlação e o reuso das informações, assegurando consistência e integridade entre os estágios da metodologia.

Os campos definidos em cada classe mostraram-se adequados para capturar e registrar os dados essenciais, garantindo que o conhecimento produzido fosse integralmente representado no Runbook de Threat Hunting.

A implementação do modelo também comprovou sua compatibilidade com diferentes formatos de serialização, facilitando o intercâmbio e a interoperabilidade entre sistemas. A adoção do formato YAML para a estruturação dos Runbooks revelou-se uma solução

eficiente, legível e acessível, permitindo tanto a leitura humana quanto o processamento automatizado. Além disso, sua conversão nativa para formatos como JSON e XML amplia as possibilidades de integração e aplicação em diferentes contextos tecnológicos.

11.3 Benefícios e Limitações

A avaliação detalhada da metodologia permitiu identificar diversos aspectos positivos que contribuem para sua efetividade e aplicabilidade. A seguir, apresentam-se os principais benefícios observados:

- **Padronização:** A utilização de etapas bem definidas e documentadas, aliada a um modelo de dados estruturado, garante um processo uniforme e resultados consistentes. Isso não apenas facilita a execução da metodologia, mas também aprimora a interpretação dos resultados por diferentes analistas.
- **Clareza:** A descrição detalhada das ações e de suas interconexões confere transparência ao processo, permitindo um entendimento claro do fluxo de trabalho e dos objetivos específicos de cada etapa.
- **Objetividade:** A definição explícita das entradas e saídas esperadas em cada fase, combinada com o uso de um modelo de dados padronizado, assegura objetividade ao processo. Essa característica contribui para a obtenção de resultados precisos e consistentes, refletidos na produção de Runbooks de alta qualidade.
- **Estrutura hierárquica:** A organização da metodologia em três níveis de granularidade (estágios, passos e ações) facilita a coordenação das atividades e delimita claramente o escopo de cada tarefa. Essa estrutura é especialmente vantajosa em operações que envolvem múltiplos analistas, permitindo melhor distribuição de responsabilidades e aumentando a eficiência colaborativa.
- **Sequenciamento:** A definição clara da ordem das etapas, estruturada sob a forma de um *pipeline*, assegura um fluxo de trabalho coerente, reduzindo desvios e garantindo a progressão lógica da análise.
- **Paralelismo:** O caráter modular da metodologia permite a execução simultânea de tarefas por diferentes analistas. Por exemplo, no caso do Mosquito Malware, equipes distintas poderiam atuar paralelamente no mapeamento das TTPs dos módulos Instalador, Lançador e Backdoor Principal. Da mesma forma, o desenvolvimento de regras de detecção para diferentes TTPs pode ser distribuído entre especialistas, acelerando o processo e aumentando a eficiência operacional.
- **Melhoria contínua:** A estruturação da metodologia conforme o ciclo PDCA (*Plan–Do–Check–Act*) possibilita revisão e aprimoramento constantes. Além disso,

o modelo de dados favorece a reutilização do conhecimento gerado, transformando o Runbook em uma valiosa fonte de *Cyber Threat Intelligence* (CTI) em nível tático.

Embora a metodologia proposta apresente múltiplas vantagens, algumas limitações foram identificadas durante a PoC e devem ser consideradas em sua aplicação prática. As principais limitações observadas são:

- **Dependência da qualidade das fontes de CTI:** A abrangência e a precisão das informações produzidas pela metodologia dependem diretamente da qualidade das fontes de *Cyber Threat Intelligence* utilizadas. A identificação de Táticas, Técnicas e, especialmente, Procedimentos está condicionada ao nível de detalhamento presente nas análises originais. Quanto mais claras, completas e ricas em informações forem as descrições dos comportamentos adversários, melhores serão as condições para o desenvolvimento de regras de detecção precisas e eficazes.
- **Cobertura limitada de variantes:** A existência de variantes de uma mesma ameaça pode resultar em diferenças significativas nas técnicas e procedimentos empregados, refletindo comportamentos distintos entre campanhas ou versões do mesmo agente malicioso. Essas variações podem impactar o mapeamento das TTPs e reduzir a efetividade das regras de detecção desenvolvidas. Para mitigar esse problema, recomenda-se restringir o escopo do Runbook a uma única variante por análise, assegurando maior precisão, consistência e validade dos resultados.
- **Dependência da análise humana:** Embora existam estudos que investigam a automatização da identificação de táticas e técnicas do MITRE ATT&CK em relatórios, a metodologia ainda depende fortemente da análise humana em etapas críticas, como a identificação dos procedimentos associados às técnicas, a definição dos relacionamentos e encadeamentos entre TTPs, e o desenvolvimento e validação das regras de detecção. Essas atividades demandam elevado nível de conhecimento técnico, interpretação contextual e julgamento analítico — competências que, até o momento, não podem ser plenamente replicadas por mecanismos automatizados.
- **Validação sensível ao cenário:** A validação das regras de detecção fundamenta-se em dados coletados em laboratório, geralmente derivados da simulação controlada da ameaça. A precisão desse processo depende fortemente da capacidade da simulação de reproduzir, com alto grau de fidelidade, as condições e dinâmicas observadas em um ambiente operacional real. Qualquer discrepância relevante entre o cenário modelado e a realidade prática pode introduzir distorções nos resultados, comprometendo tanto a corretude do processo de validação quanto a efetividade das regras validadas quando aplicadas em contextos reais de detecção.

Apesar dessas limitações, a metodologia demonstrou viabilidade e eficácia na sistematização do processo de produção de Runbooks acionáveis para Threat Hunting. O modelo de dados, aliado ao uso da ferramenta *Runbook Creator/Editor* [216], assegura consistência e padronização, tornando a proposta uma contribuição relevante para analistas e pesquisadores que atuam na detecção baseada em TTPs.

Capítulo 12

Conclusão e Trabalhos Futuros

A presente pesquisa apresentou uma metodologia estruturada para a criação de Runbooks de Threat Hunting a partir de Cyber Threat Intelligence (CTI), oferecendo uma abordagem sistemática e objetiva para mapear TTPs e, a partir delas, desenvolver regras de detecção acionáveis. Em complemento, foi proposto um modelo de dados que possibilita o registro organizado das informações geradas durante o processo, além de viabilizar a estruturação padronizada dos Runbooks, favorecendo sua interoperabilidade e reutilização em diferentes contextos operacionais.

A Prova de Conceito (PoC) realizada demonstrou a eficácia e a corretude da metodologia em múltiplos cenários, comprovando sua capacidade de mapear TTPs de ameaças complexas e gerar regras de detecção precisas e acionáveis. Os resultados indicam que a abordagem proposta é viável e eficaz, produzindo Runbooks que não apenas consolidam regras de detecção baseadas em comportamento adversário, mas também apresentam um mapeamento detalhado das TTPs que compõem cada ameaça, constituindo-se como uma fonte valiosa de inteligência tática para atividades de Threat Hunting.

Ao identificar os relacionamentos entre técnicas e o encadeamento das TTPs, a metodologia permite uma compreensão mais profunda das ações adversárias, possibilitando tanto a detecção precoce de atividades maliciosas quanto a inferência dos possíveis passos subsequentes de um ataque. Além disso, sua estrutura hierárquica — organizada em estágios, passos e ações — e a adoção do ciclo PDCA (*Plan-Do-Check-Act*) asseguram a aplicação iterativa do método, promovendo aprimoramentos contínuos e resultando na geração de Runbooks cada vez mais consistentes e eficazes.

Apesar dos resultados promissores, a aplicação prática da metodologia evidenciou desafios relevantes que precisam ser superados para ampliar sua eficácia e alcance. Um dos principais desafios reside na obtenção de fontes de CTI de alta qualidade, que apresentem análises aprofundadas sobre as técnicas adversárias e sua implementação. O nível de detalhamento dessas informações impacta diretamente a precisão do mapeamento das

TTPs e a correta identificação de seus encadeamentos, sendo um fator crítico para a efetividade da metodologia.

Outro desafio relevante reside na compreensão dos comportamentos associados às TTPs, necessária para a formulação de estratégias de detecção eficazes. Cada técnica identificada requer o entendimento detalhado da interação entre o adversário e o ambiente-alvo em nível de procedimento, bem como a elaboração de abordagens específicas para sua detecção. Esse processo demanda elevado nível de expertise dos analistas e, frequentemente, a realização de experimentos adicionais em ambiente de laboratório, a fim de validar hipóteses e refinar as estratégias de detecção.

A fidelidade do plano de emulação utilizado na validação das regras de detecção constitui um fator crítico para a confiabilidade dos resultados. A precisão da validação depende diretamente da capacidade de reproduzir o comportamento da ameaça de forma realista e em conformidade com as informações de inteligência utilizadas como referência. Embora a coleta de dados em ambientes reais possa parecer ideal, ela impõe desafios significativos, como o controle rigoroso da geração dos datasets e a separação precisa entre dados maliciosos e benignos. Ademais, em casos em que existam múltiplas variantes da ameaça, é essencial garantir a correspondência entre a variante analisada e a emulada, de modo a preservar a validade e a consistência dos resultados obtidos.

Por fim, o maior desafio consiste em aplicar a metodologia a um conjunto mais amplo de ameaças, com o objetivo de construir uma base de Runbooks abrangente e consolidada. A formação dessa base representaria um avanço significativo na operacionalização do Threat Hunting baseado em TTPs, ao possibilitar a identificação de comportamentos adversários com maior precisão e a ampliação da capacidade de detecção frente a ameaças sofisticadas.

No horizonte futuro, a criação de um banco de dados estruturado de Runbooks produzidos pela metodologia poderá trazer benefícios expressivos não apenas para o Threat Hunting, mas para todo o ecossistema de defesa cibernética. Entre as vantagens esperadas, destacam-se:

- **Aumento da eficácia na detecção de ameaças:** A detecção baseada em CTI tática, em vez de Indicadores de Comprometimento (CTIs), é mais resiliente e duradoura. Enquanto os CTIs são facilmente modificáveis pelos adversários, as TTPs representam padrões comportamentais mais estáveis, garantindo detecções mais robustas ao longo do tempo.
- **Automatização do Threat Hunting baseado em TTPs:** As regras de detecção acionáveis permitem automatizar a busca por comportamentos adversários em grandes volumes de dados. Mecanismos automatizados podem correlacionar as

TTPs detectadas com os Runbooks armazenados, tornando o processo mais eficiente e escalável.

- **Detecção precoce de ataques:** A detecção de TTPs iniciais possibilita antecipar a identificação de ameaças antes que alcancem fases mais destrutivas. Uma base de Runbooks abrangente favorece essa detecção antecipada, permitindo resposta proativa e mitigação de impactos.
- **Predição de etapas futuras:** O encadeamento de TTPs registrado nos Runbooks pode ser explorado para prever os próximos passos de um adversário durante um ataque em andamento. Essa capacidade preditiva fortalece a resposta e amplia a eficiência das contramedidas.
- **Atribuição de ataques:** Grupos adversários tendem a reutilizar táticas e técnicas específicas. Um banco de dados consolidado de TTPs permite correlacionar padrões de ataque, auxiliando na atribuição e na contextualização de novas campanhas maliciosas.
- **Treinamento de modelos de *Machine Learning*:** As regras de detecção extraídas dos Runbooks podem ser aplicadas para rotular datasets e treinar modelos de aprendizado de máquina voltados à detecção automatizada de ameaças, aprimorando a capacidade preditiva e a eficiência dos sistemas de defesa.

Em síntese, esta pesquisa oferece, primeiramente, uma contribuição relevante ao estudo das taxonomias de ataques ao apresentar um levantamento sistemático de seus requisitos e atributos estruturais, que fundamentou a proposição de critérios de avaliação qualitativos e quantitativos. Esses critérios constituem um recurso valioso não apenas para profissionais de segurança que desejam selecionar a taxonomia mais adequada a uma aplicação específica, mas também para desenvolvedores interessados em criar taxonomias mais úteis e com maior potencial de adoção pela comunidade de segurança cibernética.

Em segundo lugar, este trabalho contribui para o avanço das práticas de Threat Hunting ao propor uma metodologia robusta que integra rigor científico, aplicabilidade prática e interoperabilidade técnica. A solução apresentada estabelece uma base sólida para pesquisas futuras, reforçando a importância da inteligência tática baseada em TTPs como elemento central da detecção comportamental moderna e impulsionando avanços significativos na defesa contra ameaças cibernéticas.

Pesquisas futuras poderão aprofundar o estudo de cada estágio da metodologia, visando ao aprimoramento de procedimentos específicos e ao desenvolvimento de soluções de automação que reduzam a dependência da intervenção humana. Nesse contexto, trabalhos futuros podem ainda investigar a integração da metodologia proposta com frameworks complementares do ecossistema MITRE, como o MITRE D3FEND [190] e o MITRE En-

gage [239], com o objetivo de enriquecer, respectivamente, o mapeamento entre técnicas adversárias e contramedidas defensivas, bem como a modelagem de estratégias proativas de engajamento e dissuasão de adversários ao longo do ciclo de Threat Hunting. Ademais, a evolução da ferramenta *Runbook Creator/Editor* poderá ampliar significativamente a usabilidade e a eficiência do processo por meio da incorporação de recursos avançados, como mecanismos de versionamento de Runbooks, suporte a colaboração entre analistas, visualizações gráficas interativas dos encadeamentos de TTPs e integração com ambientes de teste.

Referências

[1] C. Simmons, C. Ellis, S. Shiva, D. Dasgupta, and Q. Wu, "AVOIDIT: A Cyber Attack Taxonomy," *9th Annual Symposium on Information Assurance (ASIA'14)*, pp. 2–12, 2014. xi, 13, 19, 27, 37, 39

[2] D. Bianco, "The pyramid of pain," Mar. 2013, acesso em: 10 ago. 2022. [Online]. Available: http://detect-respond.blogspot.com/2013/03/the-pyramid-of-pain.html xi, 3, 45, 46, 56

[3] V. Palacin, *Practical Threat Intelligence and Data-Driven Threat Hunting*, 1st ed. Packt Publishing, Feb. 2021. [Online]. Available: https://www.packtpub.com/product/practical-threat-intelligence-and-data-driven-threat-hunting/9781838556372 xi, 42, 47, 48, 51

[4] N. Moran, M. Scott, M. Oppenheim, and J. Homan, "Operation Double Tap," Nov. 2014, acesso em: 22 dez. 2023. [Online]. Available: https://www.mandiant.com/resources/blog/operation-doubletap xi, 79, 91

[5] M. Russinovich and A. Mihaiuc, "Sysmon v15.15," Jul. 2024, acesso em: 19 dez. 2024. [Online]. Available: https://learn.microsoft.com/en-us/sysinternals/downloads/sysmon xiii, 86, 87, 88, 90, 122, 123

[6] S. Monk, "Understanding Sysmon Events using SysmonSimulator," Jan. 2022, acesso em: 25 out. 2024. [Online]. Available: https://rootdse.org/posts/understanding-sysmon-events/ xiii, 90

[7] M. Sahrom Abu, S. Rahayu Selamat, A. Ariffin, and R. Yusof, "Cyber Threat Intelligence – Issue and Challenges," *Indonesian Journal of Electrical Engineering and Computer Science*, vol. 10, no. 1, p. 371, Apr. 2018. [Online]. Available: http://ijeecs.iaescore.com/index.php/IJEECS/article/view/11065 1, 2

[8] P. Institute and I. Security, "Cost of a Data Breach Report 2024," 2024, acesso em: 10 nov. 2024. [Online]. Available: https://www.ibm.com/reports/data-breach 1

[9] A. Petrosyan, "Global cybercrime estimated cost 2029," Jul. 2024, acesso em: 10 nov. 2024. [Online]. Available: https://www.statista.com/forecasts/1280009/cost-cybercrime-worldwide 1

[10] K. Baker, "Cyber Threat Intelligence Explained," Mar. 2025, acesso em: 22 maio 2025. [Online]. Available: https://www.crowdstrike.com/en-us/cybersecurity-101/threat-intelligence/ 2, 43

[11] J. Friedman and M. Bouchard, *Definitive Guide to Cyber Threat Intelligence: Using Knowledge About Adversaries to Win the War Against Targeted Attacks*. CyberEdge Group, 2015. 2, 43

[12] C. Johnson, L. Badger, D. Waltermire, J. Snyder, and C. Skorupka, "Guide to Cyber Threat Information Sharing," NIST, Tech. Rep. 800-150, Oct. 2016. [Online]. Available: https://csrc.nist.gov/pubs/sp/800/150/final 2, 43

[13] Y. You, J. Jiang, Z. Jiang, P. Yang, B. Liu, H. Feng, X. Wang, and N. Li, "TIM: Threat context-enhanced TTP intelligence mining on unstructured threat data," *Cybersecurity*, vol. 5, no. 1, p. 3, Feb. 2022. [Online]. Available: https://doi.org/10.1186/s42400-021-00106-5 2, 4

[14] D. Chismon and M. Ruks, "Threat intelligence: Collecting, analysing, evaluating," *MWR InfoSecurity Ltd*, vol. 3, no. 2, pp. 36–42, 2015. 2, 3, 43, 45

[15] J. Matilainen, "Using cyber threat intelligence as a part of organisational cybersecurity," Master's thesis, University of Jyväskylä, 2021. [Online]. Available: https://jyx.jyu.fi/handle/123456789/76092 2, 45

[16] W. Tounsi and H. Rais, "A survey on technical threat intelligence in the age of sophisticated cyber attacks," *Computers & Security*, vol. 72, pp. 212–233, Jan. 2018. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0167404817301839 2, 43

[17] Z. Zhu and T. Dumitras, "ChainSmith: Automatically Learning the Semantics of Malicious Campaigns by Mining Threat Intelligence Reports," in *2018 IEEE European Symposium on Security and Privacy (EuroS&P)*, IEEE Computer Society. London, UK: IEEE, Apr. 2018, pp. 458–472. [Online]. Available: https://ieeexplore.ieee.org/document/8406617 2

[18] MITRE, "MITRE ATT&CK®," 2025, acesso em: 5 maio 2023. [Online]. Available: https://attack.mitre.org/ 3, 5, 6, 49, 51, 55, 56, 84, 104, 109, 132, 138, 140, 147, 148

[19] C. Leite, J. den Hartog, D. Ricardo dos Santos, and E. Costante, "Actionable Cyber Threat Intelligence for Automated Incident Response," in *Secure IT Systems*, ser. NordSec 2022, H. P. Reiser and M. Kyas, Eds., Reykjavik University. Reykjavic, Iceland: Springer Lecture Notes, 2022, pp. 368–385. [Online]. Available: https://link.springer.com/chapter/10.1007/978-3-031-22295-5_20 3, 58

[20] R. Canary, "Atomic Red Team," 2024, acesso em: 21 dez. 2024. [Online]. Available: https://atomicredteam.io/atomic-red-team 4, 96

[21] MITRE, "MITRE Cyber Analytics Repository," 2022, acesso em: 18 dez. 2024. [Online]. Available: https://car.mitre.org/ 4, 59, 62, 63, 86

[22] S. Isniah, H. Hardi Purba, and F. Debora, "Plan do check action (PDCA) method: Literature review and research issues," *Jurnal Sistem dan Manajemen Industri*, vol. 4, no. 1, pp. 72–81, Jul. 2020. [Online]. Available: https://e-jurnal.lppmunsera.org/index.php/JSMI/article/view/2186 7

[23] P. R. d. P. F. Santos, P. A. A. Resende, J. J. C. Gondim, and A. C. Drummond, "Towards Robust Cyber Attack Taxonomies: A Survey with Requirements, Structures, and Assessment," *ACM Computing Surveys*, vol. 57, no. 8, pp. 199:1–199:36, Mar. 2025. [Online]. Available: https://doi.org/10.1145/3717606 10

[24] S. Vegas, N. Juristo, and V. R. Basili, "Maturing Software Engineering Knowledge through Classifications: A Case Study on Unit Testing Techniques," *IEEE Transactions on Software Engineering*, vol. 35, no. 4, pp. 551–565, Jul. 2009. [Online]. Available: https://ieeexplore.ieee.org/abstract/document/4775907 11

[25] C. Linnaeus, *Systema Naturae per regna tria naturae, secundum classes, ordines, genera, species, cum characteribus, differentiis, synonymis, locis*, 10th ed. Stockholm: Laurentii Salvii, 1758, vol. 1. [Online]. Available: https://biodiversitylibrary.org/page/726886 11

[26] M. Usman, R. Britto, J. Börstler, and E. Mendes, "Taxonomies in software engineering: A Systematic mapping study and a revised taxonomy development method," *Information and Software Technology*, vol. 85, pp. 43–59, May 2017. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0950584917300472 11

[27] R. R. Sokal, "The Principles and Practice of Numerical Taxonomy," *Taxon*, vol. 12, no. 5, pp. 190–199, 1963. [Online]. Available: https://www.jstor.org/stable/1217562 11

[28] A. J. Cain, "Taxonomy | Encyclopedia Britannica," Apr. 2023, acesso em: 15 ago. 2023. [Online]. Available: https://www.britannica.com/science/taxonomy 11

[29] P. H. Raven, B. Berlin, and D. E. Breedlove, "The Origins of Taxonomy," *Science*, vol. 174, no. 4015, pp. 1210–1213, Dec. 1971. [Online]. Available: https://www.science.org/doi/abs/10.1126/science.174.4015.1210 11

[30] Collins, "Taxonomy | Collins Dictionary," 2022, acesso em: 1 jun. 2022. [Online]. Available: https://www.collinsdictionary.com/pt/dictionary/english/taxonomy 11

[31] Cambridge, "Taxonomy | Cambridge Dictionary," 2022, acesso em: 1 jun. 2022. [Online]. Available: https://dictionary.cambridge.org/pt/dicionario/ingles/taxonomy 11

[32] K. D. Bailey, *Typologies and Taxonomies: An Introduction to Classification Techniques*, 1st ed., ser. Quantitative Applications in the Social Sciences. SAGE Publications, Inc., 1994, no. 07-102. 11

[33] B. McKelvey, "Organizational Systematics: Taxonomic Lessons from Biology," *Management Science*, vol. 24, no. 13, pp. 1428–1440, Sep. 1978. [Online]. Available: http://www.jstor.org/stable/2630648 11

[34] A. A. Ghorbani, W. Lu, and M. Tavallaee, *Network Intrusion Detection and Prevention: Concepts and Techniques*. Springer Science & Business Media, Nov. 2009, vol. 47. 11

[35] V. M. Igure and R. D. Williams, "Taxonomies of attacks and vulnerabilities in computer systems," *IEEE Communications Surveys Tutorials*, vol. 10, no. 1, pp. 6–19, 2008. 11, 15, 16, 17, 26

[36] R. Nickerson, J. Muntermann, U. Varshney, and H. Isaac, "Taxonomy development in information systems: Developing a taxonomy of mobile applications," in *European Conference in Information Systems*, 2009. 11

[37] F. Fagerholm, M. Felderer, D. Fucci, M. Unterkalmsteiner, B. Marculescu, M. Martini, L. G. W. Tengberg, R. Feldt, B. Lehtelä, B. Nagyváradi, and J. Khattak, "Cognition in Software Engineering: A Taxonomy and Survey of a Half-Century of Research," *ACM Comput. Surv.*, vol. 54, no. 11s, pp. 226:1–226:36, Sep. 2022. [Online]. Available: https://dl.acm.org/doi/10.1145/3508359 11

[38] J. D. Howard and T. A. Longstaff, "A common language for computer security incidents," Sandia National Lab. (SNL-NM), Albuquerque, NM (United States); Sandia National Lab. (SNL-CA), Livermore, CA (United States), Tech. Rep. SAND98-8667, Oct. 1998. [Online]. Available: https://www.osti.gov/biblio/751004 11, 13, 16, 17, 21

[39] W. S. McPhee, "Operating system integrity in OS/VS2," *IBM Systems Journal*, vol. 13, no. 3, pp. 230–252, 1974. 11

[40] R. P. Abbott, J. S. Chin, J. E. Donnelley, W. L. Konigsford, S. Tokubo, and D. A. Webb, "Security Analysis and Enhancements of Computer Operating Systems," National Bureau of Standards Washingtondc Inst for Computer Sciences and Technology, Tech. Rep., Apr. 1976. [Online]. Available: https://apps.dtic.mil/sti/citations/ADA436876 11

[41] R. Bisbey and D. Hollingsworth, "Protection Analysis Project: Final Report," Information Sciences Institute of University of Southern California, Research ISI/SR-78-13, May 1978. [Online]. Available: https://csrc.nist.rip/publications/history/bisb78.pdf 11

[42] I. V. Krsul, "Software vulnerability analysis," Ph.D. dissertation, Purdue University, 1998. [Online]. Available: https://www.proquest.com/dissertations-theses/software-vulnerability-analysis/docview/304449527/se-2?accountid=26646 11, 13, 15, 17

[43] S. Weber, P. A. Karger, and A. Paradkar, "A software flaw taxonomy: Aiming tools at security," *ACM SIGSOFT Software Engineering Notes*, vol. 30, no. 4, pp. 1–7, May 2005. [Online]. Available: https://doi.org/10.1145/1082983.1083209 11

[44] W. Du and A. P. Mathur, "Testing for software vulnerability using environment perturbation," *Quality and Reliability Engineering International*, vol. 18, no. 3, pp. 261–272, 2002. [Online]. Available: https://onlinelibrary.wiley.com/doi/abs/10.1002/qre.480 11

[45] K. Tsipenyuk, B. Chess, and G. McGraw, "Seven pernicious kingdoms: A taxonomy of software security errors," *IEEE Security & Privacy*, vol. 3, no. 6, pp. 81–84, Nov. 2005. 11

[46] F. Piessens, "A taxonomy of causes of software vulnerabilities in Internet software," in *Supplementary Proceedings of the 13th International Symposium on Software Reliability Engineering*, 2002, pp. 47–52. 11

[47] K. Jiwnani and M. Zelkowitz, "Susceptibility matrix: A new aid to software auditing," *IEEE Security & Privacy*, vol. 2, no. 2, pp. 16–21, Mar. 2004. 11

[48] M. Ristenbatt, "Methodology for network communication vulnerability analysis," in *MILCOM 88, 21st Century Military Communications - What's Possible?'. Conference Record. Military Communications Conference.* IEEE, Oct. 1988, pp. 493–499 vol.2. 11

[49] H. Sara, H. Faramarz, and B. Mehdi, "A Taxonomy For Network Vulnerabilities," *INTERNATIONAL JOURNAL OF INFORMATION AND COMMUNICATION TECHNOLOGY RESEARCH*, vol. 2, no. 1, pp. 29–44, 2010. [Online]. Available: https://www.sid.ir/en/Journal/ViewPaper.aspx?ID=208213 11, 26

[50] V. Pothamsetty and B. A. Akyol, "A vulnerability taxonomy for network protocols: Corresponding engineering best practice countermeasures." in *Communications, Internet, and Information Technology*, 2004, pp. 168–175. 11

[51] S. East, J. Butts, M. Papa, and S. Shenoi, "A Taxonomy of Attacks on the DNP3 Protocol," in *Critical Infrastructure Protection III*, ser. IFIP Advances in Information and Communication Technology, C. Palmer and S. Shenoi, Eds. Berlin, Heidelberg: Springer, 2009, pp. 67–81. 11

[52] S. Kamara, S. Fahmy, E. Schultz, F. Kerschbaum, and M. Frantzen, "Analysis of vulnerabilities in Internet firewalls," *Computers & Security*, vol. 22, no. 3, pp. 214–232, Apr. 2003. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0167404803003109 11

[53] H. Debar, M. Dacier, and A. Wespi, "Towards a taxonomy of intrusion-detection systems," *Computer Networks*, vol. 31, no. 8, pp. 805–822, Apr. 1999. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1389128698000176 11

[54] M. Gyanchandani, JL. Rana, and RN. Yadav, "Taxonomy of anomaly based intrusion detection system: A review," *International Journal of Scientific and Research Publications*, vol. 2, no. 12, pp. 1–13, 2012. 11

[55] M. Tehranipoor and F. Koushanfar, "A Survey of Hardware Trojan Taxonomy and Detection," *IEEE Design & Test of Computers*, vol. 27, no. 1, pp. 10–25, Jan. 2010. 12

[56] P. Prinetto and G. Roascio, "Hardware security, vulnerabilities, and attacks: A comprehensive taxonomy." in *ITASEC*, 2020, pp. 177–189. 12

[57] T. Ghasempouri, J. Raik, C. Reinbrecht, S. Hamdioui, and M. Taouil, "Survey on Architectural Attacks: A Unified Classification and Attack Model," *ACM Computing Surveys*, vol. 56, no. 2, pp. 42:1–42:32, Sep. 2023. [Online]. Available: https://doi.org/10.1145/3604803 12

[58] S. Rizvi, A. Kurtz, J. Pfeffer, and M. Rizvi, "Securing the Internet of Things (IoT): A Security Taxonomy for IoT," in *2018 17th IEEE International Conference On Trust, Security And Privacy In Computing And Communications/ 12th IEEE International Conference On Big Data Science And Engineering (TrustCom/BigDataSE)*, Aug. 2018, pp. 163–168. 12

[59] S. Khanam, I. B. Ahmedy, M. Y. Idna Idris, M. H. Jaward, and A. Q. Bin Md Sabri, "A Survey of Security Challenges, Attacks Taxonomy and Advanced Countermeasures in the Internet of Things," *IEEE Access*, vol. 8, pp. 219 709–219 743, 2020. 12

[60] C. Alex, G. Creado, W. Almobaideen, O. A. Alghanam, and M. Saadeh, "A Comprehensive Survey for IoT Security Datasets Taxonomy, Classification and Machine Learning Mechanisms," *Computers & Security*, vol. 132, p. 103283, Sep. 2023. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0167404823001931 12

[61] K. Ivaturi and L. Janczewski, "A Taxonomy for Social Engineering attacks," in *CONF-IRM 2011 Proceedings*, vol. 1. Seoul, South Korea: International Conference on Information Management (Conf-IRM), Jun. 2011, pp. 914–924. [Online]. Available: https://aisel.aisnet.org/confirm2011/15 12

[62] K. Krombholz, H. Hobel, M. Huber, and E. Weippl, "Advanced social engineering attacks," *Journal of Information Security and Applications*, vol. 22, pp. 113–122, Jun. 2015. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S2214212614001343 12

[63] T. Limbasiya and N. Doshi, "An analytical study of biometric based remote user authentication schemes using smart cards," *Computers & Electrical Engineering*, vol. 59, pp. 305–321, Apr. 2017. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0045790617302112 12

[64] O. Labazova, T. Dehling, and A. Sunyaev, "From Hype to Reality: A Taxonomy of Blockchain Applications," in *Proceedings of the 52nd Hawaii International Conference on System Sciences (HICSS 2019)*, 2019, p. 10. [Online]. Available: http://hdl.handle.net/10125/59893 12

[65] M. C. Man and V. Wei, "A taxonomy for attacks on mobile agent," in *EUROCON'2001. International Conference on Trends in Communications. Technical Program, Proceedings (Cat. No.01EX439)*, vol. 2. IEEE, Jul. 2001, pp. 385–388 vol.2. 12

[66] A. Qamar, A. Karim, and V. Chang, "Mobile malware attacks: Review, taxonomy & future directions," *Future Generation Computer Systems*, vol. 97, pp. 887–909,

Aug. 2019. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0167739X18331601 12

[67] V. L. Thing and J. Wu, "Autonomous Vehicle Security: A Taxonomy of Attacks and Defences," in *2016 IEEE International Conference on Internet of Things (iThings) and IEEE Green Computing and Communications (GreenCom) and IEEE Cyber, Physical and Social Computing (CPSCom) and IEEE Smart Data (SmartData)*. IEEE, Dec. 2016, pp. 164–170. [Online]. Available: https://ieeexplore.ieee.org/abstract/document/7917080 12

[68] I. Pekaric, C. Sauerwein, S. Haselwanter, and M. Felderer, "A taxonomy of attack mechanisms in the automotive domain," *Computer Standards & Interfaces*, vol. 78, p. 103539, Oct. 2021. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0920548921000349 12

[69] H. T. Reda, A. Anwar, A. N. Mahmood, and Z. Tari, "A Taxonomy of Cyber Defence Strategies Against False Data Attacks in Smart Grids," *ACM Computing Surveys*, vol. 55, no. 14s, pp. 331:1–331:37, Jul. 2023. [Online]. Available: https://doi.org/10.1145/3592797 12

[70] A. Bazaz and J. D. Arthur, "Towards a Taxonomy of Vulnerabilities," in *2007 40th Annual Hawaii International Conference on System Sciences (HICSS'07)*, Jan. 2007, pp. 163a–163a. 12

[71] R. R. Krishna, A. Priyadarshini, A. V. Jha, B. Appasani, A. Srinivasulu, and N. Bizon, "State-of-the-Art Review on IoT Threats and Attacks: Taxonomy, Challenges and Solutions," *Sustainability*, vol. 13, no. 16, p. 9463, Jan. 2021. [Online]. Available: https://www.mdpi.com/2071-1050/13/16/9463 12

[72] S. Hansman and R. Hunt, "A taxonomy of network and computer attacks," *Computers & Security*, vol. 24, no. 1, pp. 31–43, Feb. 2005. [Online]. Available: https://linkinghub.elsevier.com/retrieve/pii/S0167404804001804 12, 13, 18, 20

[73] S. Specht and R. Lee, "Taxonomies of distributed denial of service networks, attacks, tools and countermeasures," Princeton University, Tech. Rep. CE-L2003-03, May 2003. [Online]. Available: http://www.princeton.edu/~rblee/ELE572Papers/Fall04Readings/DDoSSurveyPaper_20030516_Final.pdf 12

[74] S. Ramanauskaite and A. Cenys, "Taxonomy of DoS attacks and their countermeasures," *Central European Journal of Computer Science*, vol. 1, no. 3, pp. 355–366, Sep. 2011. [Online]. Available: https://doi.org/10.2478/s13537-011-0024-y 12

[75] C. Douligeris and A. Mitrokotsa, "DDoS attacks and defense mechanisms: A classification," in *Proceedings of the 3rd IEEE International Symposium on Signal Processing and Information Technology (IEEE Cat. No.03EX795)*, Dec. 2003, pp. 190–193. 12

[76] J. Mirkovic and P. Reiher, "A taxonomy of DDoS attack and DDoS defense mechanisms," *ACM SIGCOMM Computer Communication Review*, vol. 34, no. 2, pp. 39–53, Apr. 2004. [Online]. Available: http://doi.org/10.1145/997150.997156 12

[77] N. Hoque, M. H. Bhuyan, R. C. Baishya, D. K. Bhattacharyya, and J. K. Kalita, "Network attacks: Taxonomy, tools and systems," *Journal of Network and Computer Applications*, vol. 40, pp. 307–324, Apr. 2014. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1084804513001756 12

[78] Z. Shi, K. Graffi, D. Starobinski, and N. Matyunin, "Threat Modeling Tools: A Taxonomy," *IEEE Security & Privacy*, vol. 20, no. 4, pp. 29–39, Jul. 2022. 12

[79] A. Chakrabarti and G. Manimaran, "Internet infrastructure security: A taxonomy," *IEEE Network*, vol. 16, no. 6, pp. 13–21, Nov. 2002. 12

[80] D. Dagon, G. Gu, C. P. Lee, and W. Lee, "A Taxonomy of Botnet Structures," in *Twenty-Third Annual Computer Security Applications Conference (ACSAC 2007)*, Dec. 2007, pp. 325–339. 12

[81] Z. Wu, Y. Ou, and Y. Liu, "A Taxonomy of Network and Computer Attacks Based on Responses," in *2011 International Conference of Information Technology, Computer Engineering and Management Sciences*, vol. 1. Nanjing, China: IEEE, Sep. 2011, pp. 26–29. 12, 18

[82] S. Souissi and A. Serhrouchni, "AIDD: A novel generic attack modeling approach," in *2014 International Conference on High Performance Computing Simulation (HPCS)*. IEEE, Jul. 2014, pp. 580–583. 12

[83] R. Heartfield and G. Loukas, "A Taxonomy of Attacks and a Survey of Defence Mechanisms for Semantic Social Engineering Attacks," *ACM Computing Surveys*, vol. 48, no. 3, pp. 37:1–37:39, Dec. 2015. [Online]. Available: https://doi.org/10.1145/2835375 12

[84] D. N. Alharthi, M. M. Hammad, and A. C. Regan, "A Taxonomy of Social Engineering Defense Mechanisms," in *Advances in Information and Communication*, ser. Advances in Intelligent Systems and Computing, K. Arai, S. Kapoor, and R. Bhatia, Eds. Cham: Springer International Publishing, 2020, pp. 27–41. 12

[85] International Organization for Standardization, "ISO/IEC 27000:2018," International Organization for Standardization, International Standard ISO/IEC 27000:2018(E), Feb. 2018. [Online]. Available: https://www.iso.org/standard/73906.html 12

[86] M. Bishop and D. Bailey, "A Critical Analysis of Vulnerability Taxonomies," CALIFORNIA UNIV DAVIS DEPT OF COMPUTER SCIENCE, Tech. Rep., Sep. 1996. [Online]. Available: https://apps.dtic.mil/sti/citations/ADA453251 12, 17

[87] P. Mishra, E. S. Pilli, V. Varadharajan, and U. Tupakula, "Intrusion detection techniques in cloud environment: A survey," *Journal of Network and Computer Applications*, vol. 77, pp. 18–47, Jan. 2017. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1084804516302417 12, 20, 34, 37, 38, 39

[88] U. Lindqvist and E. Jonsson, "How to systematically classify computer security intrusions," in *Proceedings. 1997 IEEE Symposium on Security and Privacy (Cat. No.97CB36097)*. IEEE, May 1997, pp. 154–163. 12, 13, 18

[89] J. D. Howard, "An analysis of security incidents on the Internet 1989-1995," Ph.D. dissertation, Carnegie Mellon University, 1997. [Online]. Available: https://www.proquest.com/dissertations-theses/analysis-security-incidents-on-internet-1989-1995/docview/304361939/se-2?accountid=26646 13, 16, 17

[90] D. L. Lough, "A Taxonomy Of Computer Attacks With Applications To Wireless Networks," Ph.D. dissertation, Virginia Polytechnic Institute and State University, 2001. 13

[91] Y. Sadqi and Y. Maleh, "A systematic review and taxonomy of web applications threats," *Information Security Journal: A Global Perspective*, vol. 31, no. 1, pp. 1–27, Jan. 2022. [Online]. Available: https://doi.org/10.1080/19393555.2020.1853855 13

[92] C. Joshi, U. K. Singh, and K. Tarey, "A review on taxonomies of attacks and vulnerability in computer and network system," *International Journal*, vol. 5, no. 1, pp. 742–747, 2015. 13, 17, 18

[93] E. G. Amoroso, *Fundamentals of Computer Security Technology*. USA: Prentice-Hall, Inc., Aug. 1994. 13

[94] R. Derbyshire, B. Green, D. Prince, A. Mauthe, and D. Hutchison, "An Analysis of Cyber Security Attack Taxonomies," in *2018 IEEE European Symposium on Security and Privacy Workshops (EuroS PW)*, Apr. 2018, pp. 153–161. 13, 15

[95] M. Bishop, "Vulnerabilities analysis," in *Proceedings of the Recent Advances in Intrusion Detection*, 1999, pp. 125–136. 13, 15, 19

[96] N. V. Juliadotter and K.-K. R. Choo, "Chapter 3 - CATRA: Conceptual cloud attack taxonomy and risk assessment framework," in *The Cloud Security Ecosystem*, R. Ko and K.-K. R. Choo, Eds. Boston: Syngress, 2015, pp. 37–81. [Online]. Available: https://www.sciencedirect.com/science/article/pii/B9780128015957000033 13, 33, 37, 39

[97] F. B. Cohen, "Intentional Events," in *Protection and Security on the Information Superhighway*. John Wiley & Sons, Inc., 1992, pp. 39–51. 16

[98] F. Cohen, "Information system attacks: A preliminary classification scheme," *Computers & Security*, vol. 16, no. 1, pp. 29–46, Jan. 1997. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0167404897857859 16

[99] K. Harrison and G. White, "A Taxonomy of Cyber Events Affecting Communities," in *2011 44th Hawaii International Conference on System Sciences*. IEEE, Jan. 2011, pp. 1–9. [Online]. Available: https://ieeexplore.ieee.org/abstract/document/5718647 17

[100] C. E. Landwehr, A. R. Bull, J. P. McDermott, and W. S. Choi, "A taxonomy of computer program security flaws," *ACM Computing Surveys*, vol. 26, no. 3, pp. 211–254, Sep. 1994. [Online]. Available: https://doi.org/10.1145/185403.185412 18

[101] T. Aslam, "A Taxonomy of Security Faults in the Unix Operating System," Ph.D. dissertation, Purdue University, 1995. 18

[102] C. A. Meyers, S. S. Powers, and D. M. Faissol, "Taxonomies of Cyber Adversaries and Attacks: A Survey of Incidents and Approaches," Lawrence Livermore National Lab. (LLNL), Livermore, CA (United States), Tech. Rep. LLNL-TR-419041, Oct. 2009. [Online]. Available: https://www.osti.gov/biblio/967712 18, 20

[103] X. Yue, X. Qiu, Y. Ji, and C. Zhang, "P2P attack taxonomy and relationship analysis," in *2009 11th International Conference on Advanced Communication Technology*, ser. ICACT'09, vol. 02. Gangwon, South Korea: IEEE, Feb. 2009, pp. 1207–1210. [Online]. Available: https://ieeexplore.ieee.org/document/4809630 18

[104] I. M. Chapman, S. P. Leblanc, and A. Partington, "Taxonomy of cyber attacks and simulation of their effects," in *Proceedings of the 2011 Military Modeling & Simulation Symposium*, ser. MMS '11. San Diego, CA, USA: Society for Computer Simulation International, Apr. 2011, pp. 73–80. 18, 20, 26, 37, 39

[105] C. Joshi and U. Kumar Singh, "ADMIT- A Five Dimensional Approach towards Standardization of Network and Computer Attack Taxonomies," *International Journal of Computer Applications*, vol. 100, no. 5, pp. 30–36, Aug. 2014. [Online]. Available: http://research.ijcaonline.org/volume100/number5/pxc3898091.pdf 21, 28, 37, 39

[106] N. V. Juliadotter and K.-K. R. Choo, "Cloud Attack and Risk Assessment Taxonomy," *IEEE Cloud Computing*, vol. 2, no. 1, pp. 14–20, Jan. 2015. 21, 33, 35, 37, 39

[107] A. T. Tunggal, "What is an Attack Vector? 16 Common Attack Vectors in 2023," Aug. 2023, acesso em: 13 fev. 2023. [Online]. Available: https://www.upguard.com/blog/attack-vector 22, 27

[108] H. Hindy, D. Brosset, E. Bayne, A. K. Seeam, C. Tachtatzis, R. Atkinson, and X. Bellekens, "A Taxonomy of Network Threats and the Effect of Current Datasets on Intrusion Detection Systems," *IEEE Access*, vol. 8, pp. 104 650–104 675, 2020. 28, 37, 39

[109] W. Stallings, *Handbook of Computer-Communications Standards; Vol. 1: The Open Systems Interconnection (OSI) Model and OSI-related Standards*, 2nd ed. USA: Macmillan Publishing Co., Inc., Sep. 1987, vol. 1. 28

[110] S. Kumar, "Smurf-based Distributed Denial of Service (DDoS) Attack Amplification in Internet," in *Second International Conference on Internet Monitoring and Protection (ICIMP 2007)*. IEEE, Jul. 2007, pp. 25–25. 29

[111] A. D. Householder, "CERT Advisory CA-1998-01 Smurf IP Denial-of-Service Attacks," Oct. 2021, acesso em: 15 ago. 2023. [Online]. Available: https://vuls.cert.org/confluence/display/historical/CERT+Advisory+CA-1998-01+Smurf+IP+Denial-of-Service+Attacks 29

[112] A. Bhardwaj, G. V. B. Subrahmanyam, V. Avasthi, H. Sastry, and S. Goundar, "DDoS attacks, new DDoS taxonomy and mitigation solutions — A survey," in *2016 International Conference on Signal Processing, Communication, Power and Embedded System (SCOPES)*, Oct. 2016, pp. 793–798. 29, 37, 39

[113] M. Masdari and M. Jalali, "A survey and taxonomy of DoS attacks in cloud computing," *Security and Communication Networks*, vol. 9, no. 16, pp. 3724–3751, Nov. 2016. [Online]. Available: https://onlinelibrary-wiley.ez54.periodicos.capes.gov.br/doi/10.1002/sec.1539 29, 37, 38, 39

[114] M. De Donno, A. Giaretta, N. Dragoni, and A. Spognardi, "A taxonomy of distributed denial of service attacks," in *2017 International Conference on Information Society (i-Society)*, Jul. 2017, pp. 100–107. 30, 37, 39

[115] I. Sharafaldin, A. H. Lashkari, S. Hakak, and A. A. Ghorbani, "Developing Realistic Distributed Denial of Service (DDoS) Attack Dataset and Taxonomy," in *2019 International Carnahan Conference on Security Technology (ICCST)*. IEEE, Oct. 2019, pp. 1–8. 30, 31, 37, 39

[116] BROADCOM, "Web Attack: MS SQL PacketResolution DoS," 2023, acesso em: 15 ago. 2023. [Online]. Available: https://www.broadcom.com/support/security-center/attacksignatures/detail?asid=20533 30

[117] M. Anagnostopoulos, S. Lagos, and G. Kambourakis, "Large-scale empirical evaluation of DNS and SSDP amplification attacks," *Journal of Information Security and Applications*, vol. 66, p. 103168, May 2022. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S2214212622000539 30

[118] O. Thorat, N. Parekh, and R. Mangrulkar, "TaxoDaCML: Taxonomy based Divide and Conquer using machine learning approach for DDoS attack classification," *International Journal of Information Management Data Insights*, vol. 1, no. 2, p. 100048, Nov. 2021. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S2667096821000410 31, 37, 39

[119] A. Aleroud and L. Zhou, "Phishing environments, techniques, and countermeasures: A survey," *Computers & Security*, vol. 68, pp. 160–196, Jul. 2017. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0167404817300810 31, 37, 39

[120] M. A. Basset, A. Abbas, and A. Shawky, "QRLJacking Attack," Jul. 2016, acesso em: 14 ago. 2023. [Online]. Available: https://github.com/OWASP/QRLJacking/wiki/QRLJacking-Attack 32

[121] B. B. Gupta, N. A. G. Arachchilage, and K. E. Psannis, "Defending against phishing attacks: Taxonomy of methods, current issues and future directions," *Telecommunication Systems*, vol. 67, no. 2, pp. 247–267, Feb. 2018. [Online]. Available: https://doi.org/10.1007/s11235-017-0334-z 32, 37, 39

[122] T. Alexenko, M. Jenne, S. D. Roy, and W. Zeng, "Cross-Site Request Forgery: Attack and Defense," in *2010 7th IEEE Consumer Communications and Networking Conference*, Jan. 2010, pp. 1–2. 32

[123] M. S. Siddiqui and D. Verma, "Cross site request forgery: A common web application weakness," in *2011 IEEE 3rd International Conference on Communication Software and Networks*. IEEE, May 2011, pp. 538–543. 32

[124] J. Rastenis, S. Ramanauskaitė, J. Janulevičius, A. Čenys, A. Slotkienė, and K. Pakrijauskas, "E-mail-Based Phishing Attack Taxonomy," *Applied Sciences*, vol. 10, no. 7, p. 2363, Jan. 2020. [Online]. Available: https://www.mdpi.com/2076-3417/10/7/2363 32, 37, 39

[125] S. Iqbal, M. L. Mat Kiah, B. Dhaghighi, M. Hussain, S. Khan, M. K. Khan, and K.-K. Raymond Choo, "On cloud security attacks: A taxonomy and intrusion detection and prevention as a service," *Journal of Network and Computer Applications*, vol. 74, pp. 98–120, Oct. 2016. [Online]. Available: https://linkinghub.elsevier.com/retrieve/pii/S1084804516301771 34, 35, 37, 39

[126] L. Savu, "Cloud Computing: Deployment Models, Delivery Models, Risks and Research Challenges," in *2011 International Conference on Computer and Management (CAMAN)*. IEEE, May 2011, pp. 1–4. 34

[127] S. Iqbal, M. L. M. Kiah, N. B. Anuar, B. Daghighi, A. W. A. Wahab, and S. Khan, "Service delivery models of cloud computing: Security issues and open challenges," *Security and Communication Networks*, vol. 9, no. 17, pp. 4726–4750, 2016. [Online]. Available: https://onlinelibrary.wiley.com/doi/abs/10.1002/sec.1585 34

[128] M. Swathy Akshaya and G. Padmavathi, "Taxonomy of Security Attacks and Risk Assessment of Cloud Computing," in *Advances in Big Data and Cloud Computing*, J. D. Peter, A. H. Alavi, and B. Javadi, Eds. Singapore: Springer, 2019, pp. 37–59. 34, 37, 39

[129] A. Alnaim, A. Alwakeel, and E. B. Fernandez, "A Misuse Pattern for Compromising VMs via Virtual Machine Escape in NFV," in *Proceedings of the 14th International Conference on Availability, Reliability and Security*, ser. ARES '19. New York, NY, USA: Association for Computing Machinery, Aug. 2019, pp. 1–6. [Online]. Available: https://doi.org/10.1145/3339252.3340530 35

[130] H. Abusaimeh, "Virtual Machine Escape in Cloud Computing Services," *International Journal of Advanced Computer Science and Applications*, vol. 11, no. 7, pp. 327–331, 2020. 35

[131] M. Prandini and M. Ramilli, "Return-Oriented Programming," *IEEE Security & Privacy*, vol. 10, no. 6, pp. 84–87, Nov. 2012. 35

[132] I. Hydara, A. B. M. Sultan, H. Zulzalil, and N. Admodisastro, "Current state of research on cross-site scripting (XSS) – A systematic literature review," *Information and Software Technology*, vol. 58, pp. 170–186, Feb. 2015. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0950584914001700 35

[133] S. Gupta and B. B. Gupta, "Cross-Site Scripting (XSS) attacks and defense mechanisms: Classification and state-of-the-art," *International Journal of System Assurance Engineering and Management*, vol. 8, no. 1, pp. 512–530, Jan. 2017. [Online]. Available: https://doi.org/10.1007/s13198-015-0376-0 35

[134] A. Singh, A. Sharma, N. Sharma, I. Kaushik, and B. Bhushan, "Taxonomy of Attacks on Web Based Applications," in *2019 2nd International Conference on Intelligent Computing, Instrumentation and Control Technologies (ICICICT)*, vol. 1. IEEE, Jul. 2019, pp. 1231–1235. 35, 37, 39

[135] C. Gupta, R. K. Singh, and A. K. Mohapatra, "A survey and classification of XML based attacks on web applications," *Information Security Journal: A Global Perspective*, vol. 29, no. 4, pp. 183–198, Jul. 2020. [Online]. Available: https://doi.org/10.1080/19393555.2020.1740839 35

[136] E. Koutrouli and A. Tsalgatidou, "Taxonomy of attacks and defense mechanisms in P2P reputation systems—Lessons for reputation system designers," *Computer Science Review*, vol. 6, no. 2-3, pp. 47–70, May 2012. [Online]. Available: https://linkinghub.elsevier.com/retrieve/pii/S1574013712000093 36, 37, 39

[137] M. Hollick, C. Nita-Rotaru, P. Papadimitratos, A. Perrig, and S. Schmid, "Toward a Taxonomy and Attacker Model for Secure Routing Protocols," *ACM SIGCOMM Computer Communication Review*, vol. 47, no. 1, pp. 43–48, Jan. 2017. [Online]. Available: http://doi.org/10.1145/3041027.3041033 36, 37, 38, 39

[138] S. Barnum, "Common Attack Pattern Enumeration and Classification (CAPEC) Schema Description," Department of Homeland Security, Tech. Rep., Jan. 2008. [Online]. Available: http://capec.mitre.org/documents/documentation/CAPEC_Schema_Description_v1.3.pdf 40

[139] C. von Clausewitz, *On War*. Princeton University Press, 1832. 42

[140] J. C. of Staff, "Joint publication 2-0: Joint intelligence," Department of Defense, Tech. Rep. 2-0, 2013. 42

[141] A. Bolla, "Threat Hunting Driven by Cyber Threat Intelligence," Ph.D. dissertation, POLITECNICO DI TORINO, 2022. 42

[142] CERT-UK, "An introduction to threat intelligence," CERT-UK, TLP White, 2015. [Online]. Available: https://www.ncsc.gov.uk/files/An-introduction-to-threat-intelligence.pdf 42, 43

[143] M. Kosinski, "What is Threat Intelligence?" Mar. 2025, acesso em: 22 maio 2025. [Online]. Available: https://www.ibm.com/think/topics/threat-intelligence 43

[144] R. McMillian, "Definition: Threat Intelligence," May 2013, acesso em: 23 maio 2022. [Online]. Available: https://www.gartner.com/en/documents/2487216 43

[145] S. Pearson and R. Watson, "Chapter 1 - New age of warfare: How digital forensics is reshaping today's military," in *Digital Triage Forensics*, S. Pearson and R. Watson, Eds. Boston: Syngress, 2010, pp. 1–11. [Online]. Available: https://www.sciencedirect.com/science/article/pii/B9781597495967000012 43

[146] M. R. Rahman, R. M. Hezaveh, and L. Williams, "What Are the Attackers Doing Now? Automating Cyberthreat Intelligence Extraction from Text on Pace with the Changing Threat Landscape: A Survey," *ACM Computing Surveys*, vol. 55, no. 12, pp. 241:1–241:36, Mar. 2023. [Online]. Available: https://doi.org/10.1145/3571726 43

[147] H. Griffioen, T. Booij, and C. Doerr, "Quality Evaluation of Cyber Threat Intelligence Feeds," in *Applied Cryptography and Network Security*, ser. Lecture Notes in Computer Science, M. Conti, J. Zhou, E. Casalicchio, and A. Spognardi, Eds. Cham: Springer International Publishing, 2020, pp. 277–296. 45

[148] Y. Zhou, Y. Tang, M. Yi, C. Xi, and H. Lu, "CTI View: APT Threat Intelligence Analysis System," *Security and Communication Networks*, vol. 2022, p. e9875199, Jan. 2022. [Online]. Available: https://www.hindawi.com/journals/scn/2022/9875199/ 45

[149] R. Daszczyszak, D. R. Ellis, S. Luke, and S. M. Whitley, "TTP-Based Hunting," MITRE, Tech. Rep. MTR180158, Mar. 2019. [Online]. Available: https://www.mitre.org/publications/technical-papers/ttp-based-hunting 45, 59, 60

[150] N. Lukova-Chuiko, A. Fesenko, H. Papirna, and S. Gnatyuk, "Threat Hunting as a Method of Protection Against Cyber Threats," in *Information Technology and Interactions (IT&I-2020)*, vol. 2833, Taras Shevchenko National University of Kyiv. Kyiv, Ukraine: CEUR-WS, Dec. 2020, pp. 103–113. [Online]. Available: http://ceur-ws.org/Vol-2833/ 47, 48

[151] E. Cole, "Threat Hunting: Open Season on the Adversary," *SANS Institute Information Reading Room*, Apr. 2016. 47

[152] Sqrrl, "A Framework for Cyber Threat Hunting," Sqrrl, White Paper, 2018. [Online]. Available: http://sqrrl.com/media/Framework-for-Threat-Hunting-Whitepaper.pdf 47

[153] B. Nour, M. Pourzandi, and M. Debbabi, "A Survey on Threat Hunting in Enterprise Networks," *IEEE Communications Surveys & Tutorials*, pp. 1–1, 2023. 49

[154] S. Lee, H. Cho, N. Kim, B. Kim, and J. Park, "Managing Cyber Threat Intelligence in a Graph Database: Methods of Analyzing Intrusion Sets, Threat Actors, and Campaigns," in *2018 International Conference on Platform Technology and Service (PlatCon)*, Jan. 2018, pp. 1–6. [Online]. Available: https://ieeexplore.ieee.org/abstract/document/8472752 49

[155] MITRE, "MITRE ATT&CK Frequently Asked Questions," 2023, acesso em: 21 dez. 2023. [Online]. Available: https://attack.mitre.org/resources/faq/ 51

[156] F. Maymí, R. Bixler, R. Jones, and S. Lathrop, "Towards a definition of cyberspace tactics, techniques and procedures," in *2017 IEEE International Conference on Big Data (Big Data)*, IEEE Computer Society. IEEE, Dec. 2017, pp. 4674–4679. 52, 56

[157] B. E. Strom, A. Applebaum, D. P. Miller, K. C. Nickels, A. G. Pennington, and C. B. Thomas, "Mitre att&ck: Design and philosophy," MITRE, Tech. Rep. MP180360R1, 2018. [Online]. Available: https://attack.mitre.org/docs/ATTACK_Design_and_Philosophy_March_2020.pdf 52, 60, 86

[158] MITRE, "Data Sources | MITRE ATT&CK®," 2024, acesso em: 17 dez. 2024. [Online]. Available: https://attack.mitre.org/datasources/ 52, 84, 111

[159] ——, "Data Components | MITRE ATT&CK®," 2025, acesso em: 3 nov. 2025. [Online]. Available: https://attack.mitre.org/datacomponents/ 52, 84, 87

[160] S. Flynn, "What are the challenges associated with the MITRE ATT&CK framework?" Dec. 2021, acesso em: 3 jun. 2025. [Online]. Available: https://www.computerweekly.com/news/252510650/What-are-the-challenges-associated-with-the-MITRE-ATTCK-framework 55

[161] D. Mcwhorter, "Mandiant Exposes APT1 – One of China's Cyber Espionage Units – and Releases 3,000 Indicators," Feb. 2013, acesso em: 9 mar. 2025. [Online]. Available: https://cloud.google.com/blog/topics/threat-intelligence/mandiant-exposes-apt1-chinas-cyber-espionage-units 56

[162] C. Peacock, "Summiting the Pyramid of Pain: The TTP Pyramid," Mar. 2022, acesso em: 9 mar. 2025. [Online]. Available: https://scythe.io/library/summiting-the-pyramid-of-pain-the-ttp-pyramid 56

[163] S. Sharma, "See the Evolution of the MITRE ATT&CK Framework from 2015 to Now," Apr. 2022, acesso em: 10 mar. 2025. [Online]. Available: https://d3security.com/blog/see-the-evolution-of-the-mitre-attack-framework-from-2015-to-now/ 56

[164] G. Husari, E. Al-Shaer, M. Ahmed, B. Chu, and X. Niu, "TTPDrill: Automatic and Accurate Extraction of Threat Actions from Unstructured Text of CTI Sources," in *Proceedings of the 33rd Annual Computer Security Applications Conference*, ser.

ACSAC '17, Applied Computer Security Associates (ACSA). New York, NY, USA: Association for Computing Machinery, Dec. 2017, pp. 103–115. 56

[165] U. Noor, Z. Anwar, A. W. Malik, S. Khan, and S. Saleem, "A machine learning framework for investigating data breaches based on semantic analysis of adversary's attack patterns in threat intelligence repositories," *Future Generation Computer Systems*, vol. 95, pp. 467–487, Jun. 2019. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0167739X18306708 57

[166] V. Legoy, M. Caselli, C. Seifert, and A. Peter, "Automated Retrieval of ATT&CK Tactics and Techniques for Cyber Threat Reports," in *1st Cyber Threat Intelligence Symposium, CTI 2020*. Zurich, Switzerland: Forum of Incident Response and Security Teams (FIRST), 2020, p. 20. [Online]. Available: https://arxiv.org/abs/2004.14322 57

[167] L. Zongxun, L. Yujun, Z. Haojie, and L. Juan, "Construction of TTPs From APT Reports Using BERT," in *2021 18th International Computer Conference on Wavelet Active Media Technology and Information Processing (ICCWAMTIP)*, University of Electronic Science and Technology of China (UESTC). Chengdu, China: IEEE, Dec. 2021, pp. 260–263. 57

[168] C. Sauerwein and A. Pfohl, "Towards Automated Classification of Attackers' TTPs by combining NLP with ML Techniques," Jul. 2022. [Online]. Available: http://arxiv.org/abs/2207.08478 57

[169] N. Rani, B. Saha, V. Maurya, and S. K. Shukla, "TTPHunter: Automated Extraction of Actionable Intelligence as TTPs from Narrative Threat Reports," in *Proceedings of the 2023 Australasian Computer Science Week*, ser. ACSW '23, Computing Research and Education Association of Australasia (CORE). New York, NY, USA: Association for Computing Machinery, Mar. 2023, pp. 126–134. [Online]. Available: https://doi.org/10.1145/3579375.3579391 57

[170] ——, "TTPXHunter: Actionable Threat Intelligence Extraction as TTPs from Finished Cyber Threat Reports," *Digital Threats*, Sep. 2024. [Online]. Available: https://dl.acm.org/doi/10.1145/3696427 57

[171] MITRE, "THREAT REPORT ATT&CK MAPPER (TRAM)," Aug. 2023, acesso em: 19 dez. 2023. [Online]. Available: https://mitre-engenuity.org/cybersecurity/center-for-threat-informed-defense/our-work/threat-report-attck-mapper-tram/ 57

[172] S. M. Milajerdi, R. Gjomemo, B. Eshete, R. Sekar, and V. Venkatakrishnan, "HOLMES: Real-Time APT Detection through Correlation of Suspicious Information Flows," in *2019 IEEE Symposium on Security and Privacy (SP)*, IEEE Computer Society. IEEE, May 2019, pp. 1137–1152. 58

[173] S. Gianvecchio, C. Burkhalter, H. Lan, A. Sillers, and K. Smith, "Closing the Gap with APTs Through Semantic Clusters and Automated Cybergames," in *Security and Privacy in Communication Networks*, ser. Lecture Notes of the Institute

for Computer Sciences, Social Informatics and Telecommunications Engineering, S. Chen, K.-K. R. Choo, X. Fu, W. Lou, and A. Mohaisen, Eds., European Alliance for Innovation (EAI). Cham: Springer International Publishing, 2019, pp. 235–254. 58

[174] M. Arafune, S. Rajalakshmi, L. Jaldon, Z. Jadidi, S. Pal, E. Foo, and N. Venkatachalam, "Design and Development of Automated Threat Hunting in Industrial Control Systems," in *2022 IEEE International Conference on Pervasive Computing and Communications Workshops and Other Affiliated Events (PerCom Workshops)*, IEEE Computer Society. Pisa, Italy: IEEE Computer Society, Mar. 2022, pp. 618–623. [Online]. Available: https://www.computer.org/csdl/proceedings-article/percom-workshops/2022/09767375/1Df7Wjr1dqU 58

[175] K. Kurniawan, A. Ekelhart, E. Kiesling, G. Quirchmayr, and A. M. Tjoa, "KRYSTAL: Knowledge graph-based framework for tactical attack discovery in audit data," *Computers & Security*, vol. 121, p. 102828, Oct. 2022. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S016740482200222X 58

[176] S. Harris and A. Seaborne, "SPARQL 1.1 Query Language," Mar. 2013, acesso em: 23 set. 2025. [Online]. Available: https://www.w3.org/TR/sparql11-query/ 58

[177] Sigma, "Sigma Detection Format," 2025, acesso em: 18 mar. 2025. [Online]. Available: https://sigmahq.io/ 58

[178] E. Kiesling, A. Ekelhart, K. Kurniawan, and F. Ekaputra, "The SEPSES Knowledge Graph: An Integrated Resource for Cybersecurity," in *The Semantic Web – ISWC 2019*, ser. Lecture Notes in Computer Science, C. Ghidini, O. Hartig, M. Maleshkova, V. Svátek, I. Cruz, A. Hogan, J. Song, M. Lefrançois, and F. Gandon, Eds., Semantic Web Science Association (SWSA). Auckland, New Zealand: Springer, Cham, 2019, pp. 198–214. 58

[179] K. Kurniawan, A. Ekelhart, and E. Kiesling, "An ATT&CK-KG for linking cybersecurity attacks to adversary tactics and techniques," in *International Semantic Web Conference (ISWC)*, Semantic Web Science Association. Springer International Publishing, Oct. 2021, p. 5. [Online]. Available: https://research.wu.ac.at/en/publications/an-attampck-kg-for-linking-cybersecurity-attacks-to-adversary-tac-4 58

[180] B. E. Strom, J. A. Battaglia, M. S. Kemmerer, W. Kupersanin, D. P. Miller, C. Wampler, S. M. Whitley, and R. D. Wolf, "Finding Cyber Threats with ATT&CK-Based Analytics," MITRE, Tech. Rep. MTR170202, Jul. 2017. [Online]. Available: https://www.mitre.org/publications/technical-papers/finding-cyber-threats-with-attck-based-analytics 59, 60

[181] Cutover, "What is a runbook? Best practices & benefits for enterprises," Mar. 2024, acesso em: 7 abr. 2025. [Online]. Available: https://www.cutover.com/blog/what-is-a-runbook 61

[182] ——, “Runbooks vs Playbooks: A Comprehensive Overview,” Mar. 2024, acesso em: 7 abr. 2025. [Online]. Available: https://www.cutover.com/blog/runbooks-vs-playbooks-comprehensive-overview 61

[183] Sematext, “Definition: What Is a Runbook?” 2025, acesso em: 7 abr. 2025. [Online]. Available: https://sematext.com/glossary/runbook/ 61

[184] J. Manerick, “Hunting Playbooks,” 2019, acesso em: 1 set. 2025. [Online]. Available: https://github.com/w8mej/ThreatPlays 61, 63

[185] R. Rodriguez, “Threat Hunter Playbook,” 2022, acesso em: 28 ago. 2025. [Online]. Available: https://threathunterplaybook.com/intro.html 61, 63

[186] MITRE, “Cyber Analytics Repository,” 2023, acesso em: 1 set. 2025. [Online]. Available: https://github.com/mitre-attack/car/tree/master/analytics 62

[187] B. Jordan and A. Thomson, “CACAO Security Playbooks Version 2.0,” Nov. 2023, acesso em: 1 set. 2025. [Online]. Available: https://docs.oasis-open.org/cacao/security-playbooks/v2.0/security-playbooks-v2.0.html 62, 63

[188] B. Jordan, V. Mavroeidis, L. Morgese, and A. Thomson, “Standardized Security Orchestration with CACAO,” Dec. 2023, acesso em: 1 set. 2025. [Online]. Available: https://www.oasis-open.org/2023/12/06/cacao-security-playbooks-v2-blog/ 62

[189] V. Mavroeidis, P. Eis, M. Zadnik, M. Caselli, and B. Jordan, “On the Integration of Course of Action Playbooks into Shareable Cyber Threat Intelligence,” in *2021 IEEE International Conference on Big Data (Big Data)*, IEEE Computer Society. Orlando, FL, USA: IEEE, Dec. 2021, pp. 2104–2108. [Online]. Available: https://ieeexplore.ieee.org/document/9671893 62

[190] MITRE, “D3FEND - A knowledge graph of cybersecurity countermeasures,” 2025, acesso em: 2 set. 2025. [Online]. Available: https://d3fend.mitre.org/ 63, 154

[191] R. Rodriguez, “OSSEM,” 2021, acesso em: 18 dez. 2024. [Online]. Available: https://ossemproject.com/intro.html 85, 87

[192] R. Harwood, B. Drumm, K. Toliver, and E. Ross, “Wevtutil,” Mar. 2023, acesso em: 13 abr. 2025. [Online]. Available: https://learn.microsoft.com/en-us/windows-server/administration/windows-commands/wevtutil 86

[193] osquery, “Osquery,” 2024, acesso em: 20 dez. 2024. [Online]. Available: https://github.com/osquery/osquery 86

[194] Elastic, “Kibana Query Language,” 2025, acesso em: 24 mar. 2025. [Online]. Available: https://www.elastic.co/guide/en/kibana/current/kuery-query.html 91

[195] ——, “ES|QL,” 2025, acesso em: 24 mar. 2025. [Online]. Available: https://www.elastic.co/guide/en/kibana/current/esql.html 91

[196] Apache, “Spark SQL & DataFrames | Apache Spark,” 2025, acesso em: 24 mar. 2025. [Online]. Available: https://spark.apache.org/sql/ 91, 198

[197] Sigma, “Sigma Rules,” 2025, acesso em: 24 mar. 2025. [Online]. Available: https://sigmahq.io/ 91

[198] Splunk, “About the search language - Splunk Documentation,” Jun. 2017, acesso em: 24 mar. 2025. [Online]. Available: https://docs.splunk.com/Documentation/Splunk/9.4.1/Search/Aboutthesearchlanguage 91

[199] Center for Threat-Informed Defense, “Adversary Emulation Library,” 2024, acesso em: 21 dez. 2024. [Online]. Available: https://ctid.mitre.org/resources/adversary-emulation-library/ 96

[200] ——, “Adversary Emulation Library,” Dec. 2024, acesso em: 21 dez. 2024. [Online]. Available: https://github.com/center-for-threat-informed-defense/adversary_emulation_library 96

[201] S3N4T0R, “APT Attack Simulation,” 2024, acesso em: 21 dez. 2024. [Online]. Available: https://github.com/S3N4T0R-0X0/APT-Attack-Simulation 96

[202] Elastic, “Query languages,” 2025, acesso em: 31 out. 2025. [Online]. Available: https://www.elastic.co/docs/explore-analyze/query-filter/languages 96

[203] ——, “Elasticsearch: The Official Distributed Search & Analytics Engine,” 2025, acesso em: 24 jul. 2025. [Online]. Available: https://www.elastic.co/elasticsearch 96

[204] ——, “Kibana: Explore, Visualize, Discover Data,” 2025, acesso em: 23 jul. 2025. [Online]. Available: https://www.elastic.co/kibana 96

[205] ——, “Logstash: Centralize, transform and stash your data,” 2025, acesso em: 23 jul. 2025. [Online]. Available: https://www.elastic.co/logstash 96

[206] ——, “Beats: Data Shippers for Elasticsearch,” 2025, acesso em: 23 jul. 2025. [Online]. Available: https://www.elastic.co/beats 96

[207] Apache, “Apache Kafka,” 2024, acesso em: 23 jul. 2025. [Online]. Available: https://kafka.apache.org/ 96

[208] Jupyter, “The Jupyter Notebook,” 2025, acesso em: 23 jul. 2025. [Online]. Available: https://jupyter-notebook.readthedocs.io/en/latest/notebook.html 96, 98

[209] Apache, “Apache Spark™ - Unified Engine for large-scale data analytics,” 2025, acesso em: 20 jan. 2025. [Online]. Available: https://spark.apache.org/ 96, 98, 127, 142

[210] Elastic, “Elasticsearch for Hadoop,” 2025, acesso em: 23 jul. 2025. [Online]. Available: https://www.elastic.co/elasticsearch/hadoop 97

[211] Splunk, “Splunk Attack Range,” 2024, acesso em: 22 dez. 2024. [Online]. Available: https://github.com/splunk/attack_range 97

[212] R. Rodriguez, "The HELK," 2020, acesso em: 22 dez. 2024. [Online]. Available: https://thehelk.com/intro.html 97

[213] DetectionLab, "DetectionLab," Jan. 2023, acesso em: 22 dez. 2024. [Online]. Available: https://detectionlab.network/ 97

[214] R. Rodriguez, "Security Datasets," 2022, acesso em: 23 dez. 2024. [Online]. Available: https://securitydatasets.com 99

[215] Splunk, "Attack Data," 2023, acesso em: 23 dez. 2024. [Online]. Available: https://github.com/splunk/attack_data 99

[216] P. Santos, "Runbook Creator/Editor," 2025, acesso em: 27 maio 2025. [Online]. Available: https://github.com/pauloferraz80/Runbook_Creator_Editor 103, 106, 118, 130, 134, 144, 147, 151

[217] ATTCK. MITRE, "OS Credential Dumping: LSASS Memory, Sub-technique T1003.001 - Enterprise | MITRE ATT&CK®," 2020, acesso em: 28 dez. 2024. [Online]. Available: https://attack.mitre.org/techniques/T1003/001/ 118, 119, 120

[218] MITRE, "ATT&CK® Evaluations," 2025, acesso em: 16 out. 2025. [Online]. Available: https://evals.mitre.org/ 118, 135, 142

[219] T. I. Microsoft, "Detecting and preventing LSASS credential dumping attacks," Oct. 2022, acesso em: 1 jan. 2025. [Online]. Available: https://www.microsoft.com/en-us/security/blog/2022/10/05/detecting-and-preventing-lsass-credential-dumping-attacks/ 119, 120

[220] ATTCK. MITRE, "Credential Access, Tactic TA0006 - Enterprise | MITRE ATT&CK®," Jul. 2018, acesso em: 1 jan. 2025. [Online]. Available: https://attack.mitre.org/tactics/TA0006/ 120

[221] T. Upadhyay, "Credential Dumping: LSASS Memory Dump Detection," Nov. 2024, acesso em: 1 jan. 2025. [Online]. Available: https://dev.to/tilakupadhyay/credential-dumping-lsass-memory-dump-detection-3401 120

[222] V. Pamnani, "4663(S) An attempt was made to access an object," Sep. 2021, acesso em: 3 jan. 2025. [Online]. Available: https://learn.microsoft.com/en-us/previous-versions/windows/it-pro/windows-10/security/threat-protection/auditing/event-4663 121, 122

[223] ——, "4656(S, F) A handle to an object was requested," Sep. 2021, acesso em: 4 jan. 2025. [Online]. Available: https://learn.microsoft.com/en-us/previous-versions/windows/it-pro/windows-10/security/threat-protection/auditing/event-4656 121

[224] A. Jeyashankar, "OS Credential Dumping- LSASS Memory vs Windows Logs - Security Investigation," Sep. 2022, acesso em: 5 jan. 2025. [Online]. Available: https://www.socinvestigation.com/os-credential-dumping-lsass-memory-vs-windows-logs/ 123

[225] Wardog, "Hunting for In-Memory Mimikatz with Sysmon and ELK," Mar. 2017, acesso em: 15 jan. 2025. [Online]. Available: https://rstforums.com/forum/topic/106292-hunting-for-in-memory-mimikatz-with-sysmon-and-elk/ 124

[226] R. Rodriguez, "Empire Mimikatz LogonPasswords — Security Datasets," Sep. 2020, acesso em: 19 jan. 2025. [Online]. Available: https://securitydatasets.com/notebooks/atomic/windows/credential_access/SDWIN-190518202151.html 126

[227] C. Ross and W. Schroeder, "GitHub - EmpireProject/Empire: Empire is a PowerShell and Python post-exploitation agent." 2020, acesso em: 17 out. 2025. [Online]. Available: https://github.com/EmpireProject/Empire 126

[228] F. Mudjialim, "Empire: A Powerful Post – Exploitation Tool," Jan. 2023, acesso em: 19 jan. 2025. [Online]. Available: https://www.ciso.inc/blog-posts/empire-powerful-post-exploitation-tool/ 126

[229] S. Sharma, "Examining the Activities of the Turla APT Group," Sep. 2023, acesso em: 26 jan. 2025. [Online]. Available: https://www.trendmicro.com/en_us/research/23/i/examining-the-activities-of-the-turla-group.html 130

[230] MITRE, "Turla | MITRE ATT&CK®," Jun. 2024, acesso em: 26 jan. 2025. [Online]. Available: https://attack.mitre.org/groups/G0010/ 130

[231] spol. s r.o. ESET, "Diplomats in Eastern Europebitten by a Turla mosquito," ESET, Tech. Rep., Jan. 2018. [Online]. Available: https://web-assets.esetstatic.com/wls/2018/01/ESET_Turla_Mosquito.pdf 131

[232] MITRE, "APT29," May 2017, acesso em: 26 out. 2025. [Online]. Available: https://attack.mitre.org/groups/G0016/ 134

[233] F-Secure, "The Dukes 7 Years of Russian Cyberespionage," F-Secure, Whitepaper, Sep. 2015. [Online]. Available: https://blog-assets.f-secure.com/wp-content/uploads/2020/03/18122307/F-Secure_Dukes_Whitepaper.pdf 135

[234] MITRE, "APT29 / Cozy Bear / The Dukes Emulation Plan," MITRE Corporation, Tech. Rep. 19-03607-2, Jan. 2021. [Online]. Available: https://github.com/mitre-attack/attack-arsenal/blob/master/adversary_emulation/APT29/Emulation_Plan/APT29_EmuPlan.pdf 135

[235] ——, "APT29 Enterprise Evaluation 2020," 2020, acesso em: 17 out. 2025. [Online]. Available: https://evals.mitre.org/enterprise/apt29 135

[236] ——, "Scenario 1," 2020, acesso em: 17 out. 2025. [Online]. Available: https://attackevals.github.io/ael/enterprise/apt29/emulation_plan/scenario_1/ 136, 142

[237] R. Rodriguez, "APT29 Evals - Open Datasets," 2020, acesso em: 25 out. 2025. [Online]. Available: https://github.com/OTRF/detection-hackathon-apt29/blob/master/datasets/day1/README.md 142

[238] ——, “Mordor Labs — Part 1: Deploying ATT&CK APT29 Evals Environments via ARM Templates to Create Detection Research Opportunities,” Jun. 2020, acesso em: 25 out. 2025. [Online]. Available: https://medium.com/threat-hunters-forge/mordor-labs-part-1-deploying-att-ck-apt29-evals-environments-via-arm-templates-to-create-1c6c4bc32c9a 142

[239] MITRE, “MITRE Engage™ | An Adversary Engagement Framework from MITRE,” 2025, acesso em: 20 dez. 2025. [Online]. Available: https://engage.mitre.org/ 155

Apêndice A

TTPs mapeadas do Mosquito Malware

A lista a seguir apresenta as 29 TTPs identificadas na Seção 10.2, no contexto do *Mosquito Malware*, organizadas conforme o modelo de dados definido nesta pesquisa. Para otimizar a apresentação das informações, campos vazios foram propositalmente omitidos, assim como os campos `notes` e `references`. Este último foi suprimido, pois todas as TTPs referenciam o mesmo relatório de CTI, identificado pelo código *REF-0017-2739-5256*. Além disso, o conteúdo do campo `procedure` pode ser exibido parcialmente, utilizando-se o símbolo "(···)" para indicar a omissão de trechos extensos.

- TTP ID: TTP-0017-2739-5258
  Tactic: TA0002 (Execution)
  Technique: T1204.002 (User Execution: Malicious File)
  Procedure: `Once the user has downloaded and launched the fake Flash ins↵ taller, the compromise process starts.`
  TTP Chain: TTP-0017-2739-5978
- TTP ID: TTP-0017-2739-5978
  Tactic: TA0005 (Defense Evasion)
  Technique: T1027 (Obfuscated Files or Information)
  Procedure: `In recent versions, the installer is always obfuscated wit↵ h what seems to be a custom crypter. Firstly, the crypter makes heavy use of opaque predicates along with arithmetic operations.` (···)
  TTP Chain: TTP-0017-2740-3192
- TTP ID: TTP-0017-2740-3192
  Tactic: TA0005 (Defense Evasion)
  Technique: T1497.001 (Virtualization/Sandbox Evasion: System Checks)

Procedure: `Secondly, after the first layer is de-obfuscated, a call to the Win32 API SetupDiGetClassDevs (0,0,0,0xFFFFFFFF) is performed, and the crypter then checks whether the return value equals 0xE000021A.` (···)
Secondary Techniques: T1106 (Execution through API)
TTP Chain: TTP-0017-2740-3422

- TTP ID: TTP-0017-2740-3422
Tactic: TA0005 (Defense Evasion)
Technique: T1620 (Reflective Code Loading)
Procedure: `Thirdly, the real code is divided into several chunks that are decrypted, using a custom function, and re-ordered at run time to build a PE in memory. It is then executed in-place by the crypter's PE loader function.`
TTP Chain: TTP-0017-2740-3554; TTP-0017-2740-3996; TTP-0017-2744-5189
- TTP ID: TTP-0017-2740-3554
Tactic: TA0005 (Defense Evasion)
Technique: T1036.005 (Masquerading: Match Legitimate Name or Location)
Procedure: `Once decrypted, the installer searches the %APPDATA% subtree and drops two files in the deepest folder it finds. When searching for this folder, it avoids any folder that contains AVAST in its name. It then uses the filename of one of the non-hidden files in this folder, truncated at the extension, as the base filename for the files it will drop.` (···)
- TTP ID: TTP-0017-2740-3996
Tactic: TA0007 (Discovery)
Technique: T1518.001 (Software Discovery: Security Software Discovery)
Procedure: `If the antivirus display name, retrieved using Windows Management Instrumentation (WMI), is "Total Security".`
Secondary Techniques: T1047 (Windows Management Instrumentation)
TTP Chain: TTP-0017-2740-4665; TTP-0017-2740-5434
- TTP ID: TTP-0017-2740-4665
Tactic: TA0003 (Persistence)
Technique: T1547.001 (Boot or Logon Autostart Execution: Registry Run Keys / Startup Folder)
Procedure: `It establishes persistence by using a Run registry key. If the antivirus display name is "Total Security", it adds rundll32.exe [backdoor_path], StartRoutine in HKCU\Software\Run\auto_update.`

Secondary Techniques: T1218.011 (System Binary Proxy Execution: Rundll32)
TTP Chain: TTP-0017-2765-8882

- TTP ID: TTP-0017-2740-5434
  Tactic: TA0003 (Persistence)
  Technique: T1546.015 (Event Triggered Execution: Component Object Model Hijacking)
  Procedure: `It will replace the registry entry under HKCR\CLSID\{d914 4dcd-e998-4eca-ab6a-dcd83ccba16d}\InprocServer32 or HKCR\CLSID\{0824 4ee6-92f0-47f2-9fc9-929baa2e7235}\InprocServer32 with the path to the loader.` (···)
  Secondary Techniques: T1112 (Modify Registry)
  TTP Chain: TTP-0017-2744-6446
- TTP ID: TTP-0017-2744-5189
  Tactic: TA0003 (Persistence)
  Technique: T1136.001 (Create Account: Local Account)
  Procedure: `The installer creates an administrative account HelpAssistan t (or HelpAsistant in some samples) with the password sysQ!123.`
  TTP Chain: TTP-0017-2744-5342
- TTP ID: TTP-0017-2744-5342
  Tactic: TA0004 (Privilege Escalation)
  Technique: T1134.001 (Access Token Manipulation: Token Impersonation/Theft)
  Procedure: `The LocalAccountToken FilterPolicy is set to 1, allowing rem ote administrative actions.`
- TTP ID: TTP-0017-2744-6446
  Tactic: TA0005 (Defense Evasion)
  Technique: T1574.001 (Hijack Execution Flow: DLL Search Order Hijacking)
  Procedure: `The launcher, named DebugParser.dll internally, is called w hen the hijacked COM object is loaded. It is responsible for launching the main backdoor and for loading the hijacked COM object.` (···)
  Secondary Techniques: T1218.011 (System Binary Proxy Execution: Rundll32)
  TTP Chain: TTP-0017-2765-8882
- TTP ID: TTP-0017-2765-8882
  Tactic: TA0005 (Defense Evasion)
  Technique: T1070.004 (Indicator Removal: File Deletion)
  Procedure: `Firstly, the CommanderDLL module deletes the dropper (the fa ke Flash installer) file.` (···)
  TTP Chain: TTP-0017-2765-9210

- TTP ID: TTP-0017-2765-9210
  Tactic: TA0005 (Defense Evasion)
  Technique: T1027.011 (Obfuscated Files or Information: Fileless Storage)
  Procedure: `Secondly, it sets up some internal structures and stores con⌋figuration values in the registry.` (···)
  Secondary Techniques: T1027.013 (Obfuscated Files or Information: Encrypted/Encoded File)
  Related TTPs: TTP-0017-2769-9435
  TTP Chain: TTP-0017-2769-9896
- TTP ID: TTP-0017-2769-9435
  Tactic: TA0011 (Command and Control)
  Technique: T1573.001 (Encrypted Channel: Symmetric Cryptography)
  Procedure: `This backdoor relies on a custom encryption algorithm. Eac⌋h byte of the plaintext is XORed with a stream generated by a function that looks similar to the Blum Blum Shub algorithm` (···)
  Related TTPs: TTP-0017-2765-9210; TTP-0017-2769-9896; TTP-0017-2770-0651
- TTP ID: TTP-0017-2769-9896
  Tactic: TA0011 (Command and Control)
  Technique: T1102.003 (Web Service: One-Way Communication)
  Procedure: `Third, an additional C&C server address is downloaded from a⌋document hosted on Google Docs` (···)
  Secondary Techniques: T1573.001 (Encrypted Channel: Symmetric Cryptography); T1071.001 (Application Layer Protocol: Web Protocols)
  Related TTPs: TTP-0017-2769-9435
  TTP Chain: TTP-0017-2770-0651
- TTP ID: TTP-0017-2770-0651
  Tactic: TA0011 (Command and Control)
  Technique: T1071.001 (Application Layer Protocol: Web Protocols)
  Procedure: `The requests to the C&C server always use the same URL schem⌋e: https://[C&C server domain]/scripts/m/query.php?id=[base64(encrypte⌋d data)].` (···)
  Secondary Techniques: T1132.001 (Data Encoding: Standard Encoding); T1573.001 (Encrypted Channel: Symmetric Cryptography)
  Related TTPs: TTP-0017-2769-9435
  TTP Chain: TTP-0017-2770-1458; TTP-0017-2770-1785; TTP-0017-2770-1793; TTP-0017-2770-1875
- TTP ID: TTP-0017-2770-1458

Tactic: TA0007 (Discovery)
Technique: T1016 (System Network Configuration Discovery)
Procedure: `The initial request contains general information about the c⌋ ompromised machine, such as the result of the command 'ipconfig'.`
References: REF-0017-2739-5256
TTP Chain: TTP-0017-2770-1958

- TTP ID: TTP-0017-2770-1785
  Tactic: TA0007 (Discovery)
  Technique: T1082 (System Information Discovery)
  Procedure: `The initial request contains general information about the c⌋ ompromised machine, such as the result of the command 'set'.`
  TTP Chain: TTP-0017-2770-1958
- TTP ID: TTP-0017-2770-1793
  Tactic: TA0007 (Discovery)
  Technique: T1033 (System Owner/User Discovery)
  Procedure: `The initial request contains general information about the c⌋ ompromised machine, such as the result of the command 'whoami'.`
  TTP Chain: TTP-0017-2770-1958
- TTP ID: TTP-0017-2770-1875
  Tactic: TA0007 (Discovery)
  Technique: T1057 (Process Discovery)
  Procedure: `The initial request contains general information about the c⌋ ompromised machine, such as the result of the command 'tasklist'.`
  TTP Chain: TTP-0017-2770-1958
- TTP ID: TTP-0017-2770-1958
  Tactic: TA0011 (Command and Control)
  Technique: T1071.001 (Application Layer Protocol: Web Protocols)
  Procedure: `Then, the C&C server replies with one of several batches of ⌋ instructions with commands.` (···)
  Related TTPs: TTP-0017-2770-4412
  TTP Chain: TTP-0017-2770-5684; TTP-0017-2771-8918; TTP-0017-2771-9069; TTP-0017-2771-9252; TTP-0017-2771-9368; TTP-0017-2791-9294
- TTP ID: TTP-0017-2770-4412
  Tactic: TA0011 (Command and Control)
  Technique: T1573.001 (Encrypted Channel: Symmetric Cryptography)
  Procedure: `The packet is fully encrypted (except the first four bytes),⌋ with the same algorithm, derived from Blum Blum Shub,` (···)

Related TTPs: TTP-0017-2770-1958

- TTP ID: TTP-0017-2770-5684
  Tactic: TA0011 (Command and Control)
  Technique: T1105 (Ingress Tool Transfer)
  Procedure: `Command ID: 0x3001. Description: Download file to the compro⌋ mised machine.`
  TTP Chain: TTP-0017-2771-8516
- TTP ID: TTP-0017-2771-8516
  Tactic: TA0002 (Execution)
  Technique: T1106 (Native API)
  Procedure: `Command ID: 0x3001. If there is .dll or .exe in the filename⌋ , run it using LoadLibrary or CreateProcess.`
- TTP ID: TTP-0017-2771-8918
  Tactic: TA0005 (Defense Evasion)
  Technique: T1070.004 (Indicator Removal: File Deletion)
  Procedure: `Command ID: 0x3003. Description: Delete a file using DeleteF⌋ ileW.`
- TTP ID: TTP-0017-2771-9069
  Tactic: TA0010 (Exfiltration)
  Technique: T1041 (Exfiltration Over C2 Channel)
  Procedure: `Command ID: 0x3004. Description: Exfiltrate a file (max size⌋ sent = 104,857,600 bytes).`
  Secondary Techniques: T1030 (Data Transfer Size Limits)
- TTP ID: TTP-0017-2771-9252
  Tactic: TA0005 (Defense Evasion)
  Technique: T1027.011 (Obfuscated Files or Information: Fileless Storage)
  Procedure: `Command ID: 0x3005 or 0x0007 or 0x0008. Description: Store d⌋ ata to the registry Flags. The size of data should be` $\leq$ `240 bytes.`
- TTP ID: TTP-0017-2771-9368
  Tactic: TA0011 (Command and Control)
  Technique: T1059.003 (Command and Scripting Interpreter: Windows Command Shell)
  Procedure: `Command ID: 0x3006. Description: Execute cmd.exe /c [command⌋ ]. The result is read using a pipe and sent back to the C&C.`
  Secondary Techniques: T1559 (Inter-Process Communication); T1041 (Exfiltration Over C2 Channel)
- TTP ID: TTP-0017-2791-9294

Tactic: TA0002 (Execution)
Technique: T1059.001 (Command and Scripting Interpreter: PowerShell)
Procedure: `In some of the samples the backdoor is also able to launch PowerShell scripts.`

Apêndice B

Dados da ameaça e referências do APT29

A lista a seguir apresenta as informações da ameaça analisada na Seção 10.3, bem como as referências utilizadas, organizadas de acordo com o modelo de dados proposto. Campos vazios foram propositalmente omitidos para otimizar o uso de espaço.

Title: APT29 cenário 1
Threat ID: THR-0017-3066-2211
Creation Date: 2024-11-03
Update Date: 2025-10-17
Type: APT
Domain: Enterprise
Platforms: Windows
Description: Esta ameaça é um ataque cibernético sofisticado inspirada no modus operandi do grupo APT29. Ela começa com um usuário legítimo executando um arquivo malicioso mascarado como um documento do Word. O payload executado cria uma conexão C2 criptografada, permitindo ao invasor interagir com o sistema, coletar e exfiltrar arquivos. O atacante faz upload de um novo payload com um script PowerShell oculto, elevando privilégios e estabelecendo uma nova conexão C2 via HTTPS. Ferramentas adicionais são para enumeração, limpeza dos rastros do acesso inicial, reconhecimento e persistência, que é garantida através da criação de um serviço e de um payload na pasta de inicialização. Credenciais e dados sensíveis são coletados, incluindo chaves privadas, hashes de senha, capturas de tela, dados da área de transferência do usuário e pressionamentos de teclas. Consultas LDAP são usadas para descobrir vítimas secundárias no domínio, para as quais são realizados uploads de novos payloads via sessão remota do PowerShell, sendo estes executados por meio do utilitário PSExec usando as credenciais roubadas anteriormente. Uma eventual reinicialização do

sistema aciona mecanismos de persistência, permitindo ao atacante manter o controle e continuar a coleta de dados sensíveis, enquanto se disfarça como um utilitário legítimo e exfiltra informações compactadas e criptografadas para um compartilhamento WebDAV controlado pelo invasor.

References:

- ID: REF-0017-3068-4751
  Type: Web page
  Title: MITRE Engenuity ATT&CK Evaluations - APT29
  Authors:
  Date: 2020
  Link: `https://attackevals.mitre-engenuity.org/enterprise/apt29/`
- ID: REF-0017-3068-4862
  Type: Web page
  Title: APT29
  Authors: Daniyal Naeem, Matt Brenton, Katie Nickels, Joe Gumke, Liran Ravich
  Date: 2017-05-31
  Link: `https://attack.mitre.org/groups/G0016/`
- ID: REF-0017-3068-5073
  Type: APT29 Cenário 1
  Title: Scenario 1
  Authors: Michael Butt
  Date: 2020
  Link: `https://github.com/attackevals/ael/tree/main/Enterprise/apt29/Emulation_Plan/Scenario_1`
- ID: REF-0017-3103-4789
  Type: Dataset
  Title: APT29 Evals - Open Datasets
  Authors: Roberto Rodriguez
  Date: 2020
  Link: `https://github.com/OTRF/detection-hackathon-apt29/tree/master/datasets/day1`
  Notes: Download do arquivo em `https://github.com/OTRF/detection-hackathon-apt29/raw/refs/heads/master/datasets/day1/apt29_evals_day1_manual.zip`
- ID: REF-0017-6044-5046
  Type: Emulation Plan
  Title: APT29 Day 1 (Steps 1 through 10)

Authors: Connor Magee
Date: 2020
Link: https://github.com/mitre-attack/attack-arsenal/tree/master/adversary_emulation/APT29/Emulation_Plan/Day%201

Apêndice C

TTPs mapeadas do APT29

A lista a seguir apresenta as 45 TTPs identificadas no primeiro estágio da metodologia (Mapeamento Ameaça–TTP), durante o estudo de caso descrito na Seção 10.3, referente ao ataque APT29. Os resultados estão organizados conforme o modelo de dados proposto, observando-se as seguintes considerações: (i) campos vazios foram propositalmente omitidos, a fim de otimizar o uso do espaço; e (ii) o campo `references` foi suprimido, uma vez que todas as TTPs se baseiam na mesma fonte, identificada pelo código *REF-0017-3068-5073*.

- TTP ID: TTP-0017-3068-5003
  Tactic: TA0002 (Execution)
  Technique: T1204.002 (User Execution: Malicious File)
  Procedure: `Um usuário legítimo clica em um payload executável (tipo screensaver executável) mascarado de documento Word benigno por meio da técnica "Right-to-Left Override" (RTLO) que usa o caractere U+202E no nome do arquivo para invertê-lo.`
  Secondary Techniques: T1036.002 (Masquerading: Right-to-Left Override)
  TTP Chain: TTP-0017-6044-2942
- TTP ID: TTP-0017-6044-2942
  Tactic: TA0011 (Command and Control)
  Technique: T1573.001 (Encrypted Channel: Symmetric Cryptography)
  Procedure: `Uma vez executado, o payload cria uma ligação C2 através da porta 1234 usando a cifra criptográfica RC4.`
  Secondary Techniques: T1571 (Non-Standard Port)
  TTP Chain: TTP-0017-6044-6572
- TTP ID: TTP-0017-6044-6572
  Tactic: TA0002 (Execution)
  Technique: T1059.001 (Command and Scripting Interpreter: PowerShell)

Procedure: `O invasor usa a conexão C2 ativa para lançar os programas interativos cmd.exe e powershell.exe.`
Secondary Techniques: T1059.003 (Command and Scripting Interpreter: Windows Command Shell)
TTP Chain: TTP-0017-6044-6947

- TTP ID: TTP-0017-6044-6947
Tactic: TA0007 (Discovery)
Technique: T1083 (File and Directory Discovery)
Procedure: `O invasor executa um comando de uma linha para pesquisar o sistema de arquivos em busca de documentos e arquivos de mídia, coletando-os.`
Secondary Techniques: T1005 (Data from Local System)
TTP Chain: TTP-0017-6044-7344
- TTP ID: TTP-0017-6044-7344
Tactic: TA0009 (Collection)
Technique: T1560.001 (Archive Collected Data: Archive via Utility)
Procedure: `O invasor comprime o conteúdo coletado em arquivo único.`
Secondary Techniques: T1074.001 (Data Staged: Local Data Staging)
TTP Chain: TTP-0017-6044-7520
- TTP ID: TTP-0017-6044-7520
Tactic: TA0010 (Exfiltration)
Technique: T1041 (Exfiltration Over C2 Channel)
Procedure: `O arquivo é então exfiltrado pela conexão C2 existente.`
TTP Chain: TTP-0017-6044-7700
- TTP ID: TTP-0017-6044-7700
Tactic: TA0011 (Command and Control)
Technique: T1105 (Ingress Tool Transfer)
Procedure: `O invasor carrega um novo payload para a vítima. O payload é um arquivo de imagem legitimamente formado (formato png) com um script PowerShell oculto.`
Secondary Techniques: T1027.003 (Obfuscated Files or Information: Steganography)
TTP Chain: TTP-0017-6045-0689
- TTP ID: TTP-0017-6045-0689
Tactic: TA0004 (Privilege Escalation)
Technique: T1548.002 (Abuse Elevation Control Mechanism: Bypass User Account Control)

Procedure: O invasor eleva os privilégios por meio de um bypass do controle de conta de usuário (UAC), que executa o payload recém-adicionado. Esse processo é realizado da seguinte forma: O invasor cria uma nova chave de registro e uma entrada `DelegateExecute` que aponta para seu payload malicioso, como um prompt de comando ou um script. Em seguida, o invasor executa o `sdclt.exe`. Devido à sua configuração `autoElevate`, o processo `sdclt.exe` inicia um novo processo com privilégios (administrador). Esse processo `sdclt.exe` acessa o registro e, devido à modificação do invasor, é induzido a executar a carga maliciosa com privilégios elevados, ignorando o prompt de segurança do UAC.
Secondary Techniques: T1546.015 (Event Triggered Execution: Component Object Model Hijacking)
TTP Chain: TTP-0017-6045-2068

- TTP ID: TTP-0017-6045-2068
  Tactic: TA0011 (Command and Control)
  Technique: T1071.001 (Application Layer Protocol: Web Protocols)
  Procedure: Uma nova conexão C2 é estabelecida pela porta 443 usando o protocolo HTTPS.
  Secondary Techniques: T1573 (Encrypted Channel); T1043 (Commonly Used Port (deprecated))
  TTP Chain: TTP-0017-6045-2305
- TTP ID: TTP-0017-6045-2305
  Tactic: TA0005 (Defense Evasion)
  Technique: T1112 (Modify Registry)
  Procedure: O invasor remove os artefatos da escalação de privilégios do Registro.
  TTP Chain: TTP-0017-6045-2548
- TTP ID: TTP-0017-6045-2548
  Tactic: TA0011 (Command and Control)
  Technique: T1105 (Ingress Tool Transfer)
  Procedure: O invasor carrega ferramentas adicionais (arquivo zip) através do novo acesso elevado.
  TTP Chain: TTP-0017-6045-8979
- TTP ID: TTP-0017-6045-8979
  Tactic: TA0005 (Defense Evasion)
  Technique: T1140 (Deobfuscate/Decode Files or Information)

Procedure: `Descompacta a ferramenta usando um shell interativo com acesso elevado (powershell.exe) para posicioná-la no destino para uso.`
Secondary Techniques: T1059.001 (Command and Scripting Interpreter: PowerShell)
TTP Chain: TTP-0017-6045-9266

- TTP ID: TTP-0017-6045-9266
  Tactic: TA0007 (Discovery)
  Technique: T1057 (Process Discovery)
  Procedure: `O invasor então enumera os processos em execução para descobrir/encerrar o acesso inicial da Etapa 1.`
  TTP Chain: TTP-0017-6046-1439
- TTP ID: TTP-0017-6046-1439
  Tactic: TA0005 (Defense Evasion)
  Technique: T1070.004 (Indicator Removal: File Deletion)
  Procedure: `Exclui os arquivos associados ao acesso inicial da Etapa 1.`
  TTP Chain: TTP-0017-6046-1657; TTP-0017-6046-2409; TTP-0017-6046-2437; TTP-0017-6046-2449; TTP-0017-6046-2450; TTP-0017-6046-2453; TTP-0017-6046-2456
- TTP ID: TTP-0017-6046-1657
  Tactic: TA0007 (Discovery)
  Technique: T1083 (File and Directory Discovery)
  Procedure: `O invasor lança um script PowerShell que executa uma ampla variedade de comandos de reconhecimento, incluindo enumerar o caminho do diretório temporário do usuário.`
  Secondary Techniques: T1059.001 (Command and Scripting Interpreter: PowerShell)
  TTP Chain: TTP-0017-6046-4431; TTP-0017-6046-4432
- TTP ID: TTP-0017-6046-2409
  Tactic: TA0007 (Discovery)
  Technique: T1033 (System Owner/User Discovery)
  Procedure: `O invasor lança um script PowerShell que executa uma ampla variedade de comandos de reconhecimento, incluindo a enumeração do nome de usuário atual.`
  Secondary Techniques: T1059.001 (Command and Scripting Interpreter: PowerShell)
  TTP Chain: TTP-0017-6046-4431; TTP-0017-6046-4432
- TTP ID: TTP-0017-6046-2437
  Tactic: TA0007 (Discovery)
  Technique: T1082 (System Information Discovery)
  Procedure: `O invasor lança um script PowerShell que executa uma ampla variedade de comandos de reconhecimento, incluindo a enumeração do nome`

do host do computador e da versão do sistema operacional.
Secondary Techniques: T1059.001 (Command and Scripting Interpreter: PowerShell)
TTP Chain: TTP-0017-6046-4431; TTP-0017-6046-4432

- TTP ID: TTP-0017-6046-2449
  Tactic: TA0007 (Discovery)
  Technique: T1016 (System Network Configuration Discovery)
  Procedure: O invasor lança um script PowerShell que executa uma ampla variedade de comandos de reconhecimento, incluindo a enumeração do nome de domínio atual.
  Secondary Techniques: T1059.001 (Command and Scripting Interpreter: PowerShell)
  TTP Chain: TTP-0017-6046-4431; TTP-0017-6046-4432
- TTP ID: TTP-0017-6046-2450
  Tactic: TA0007 (Discovery)
  Technique: T1057 (Process Discovery)
  Procedure: O invasor lança um script PowerShell que executa uma ampla variedade de comandos de reconhecimento, incluindo a enumeração do ID do processo atual.
  Secondary Techniques: T1059.001 (Command and Scripting Interpreter: PowerShell)
  TTP Chain: TTP-0017-6046-4431; TTP-0017-6046-4432
- TTP ID: TTP-0017-6046-2453
  Tactic: TA0007 (Discovery)
  Technique: T1518.001 (Software Discovery: Security Software Discovery)
  Procedure: O invasor lança um script PowerShell que executa uma ampla variedade de comandos de reconhecimento, incluindo a enumeração de softwares antivírus e firewall.
  Secondary Techniques: T1059.001 (Command and Scripting Interpreter: PowerShell)
  TTP Chain: TTP-0017-6046-4431; TTP-0017-6046-4432
- TTP ID: TTP-0017-6046-2456
  Tactic: TA0007 (Discovery)
  Technique: T1069 (Permission Groups Discovery)
  Procedure: O invasor lança um script PowerShell que executa uma ampla variedade de comandos de reconhecimento, incluindo enumerar grupos de usuário no sistema por meio da API NetUserGetGroups do Netapi32.dll.
  Secondary Techniques: T1106 (Native API); T1059.001 (Command and Scripting Interpreter: PowerShell)
  TTP Chain: TTP-0017-6046-4431; TTP-0017-6046-4432
- TTP ID: TTP-0017-6046-4431

Tactic: TA0003 (Persistence)
Technique: T1543.003 (Create or Modify System Process: Windows Service)
Procedure: `Cria um novo serviço (javamtsup) que execute um binário de serviço (javamtsup.exe) na inicialização do sistema.`
TTP Chain: TTP-0017-6046-5407

- TTP ID: TTP-0017-6046-4432
  Tactic: TA0003 (Persistence)
  Technique: T1547.001 (Boot or Logon Autostart Execution: Registry Run Keys / Startup Folder)
  Procedure: `Cria uma carga maliciosa (hostui.lnk) na pasta Inicialização do Windows que seja executada no login.`
  TTP Chain: TTP-0017-6046-5407
- TTP ID: TTP-0017-6046-5407
  Tactic: TA0006 (Credential Access)
  Technique: T1555.003 (Credentials from Password Stores: Credentials from Web Browsers)
  Procedure: `O invasor acessa as credenciais armazenadas em um navegador da web local usando uma ferramenta renomeada para se passar por um aplicativo legítimo.`
  Secondary Techniques: T1552.001 (Unsecured Credentials: Credentials In Files); T1036.005 (Masquerading: Match Legitimate Name or Location)
  Related TTPs: TTP-0017-6046-5816
  TTP Chain: TTP-0017-6046-6464; TTP-0017-6046-7568
- TTP ID: TTP-0017-6046-5816
  Tactic: TA0005 (Defense Evasion)
  Technique: T1036.005 (Masquerading: Match Legitimate Name or Location)
  Procedure: `Ferramenta de dump de senhas do Chrome mascarada como accesscheck.exe, uma ferramenta legítima da Sysinternals.`
  Related TTPs: TTP-0017-6046-5407
- TTP ID: TTP-0017-6046-6464
  Tactic: TA0006 (Credential Access)
  Technique: T1552.004 (Unsecured Credentials: Private Keys)
  Procedure: `O invasor coleta chaves privadas.`
  TTP Chain: TTP-0017-6049-5413; TTP-0017-6049-7247; TTP-0017-6049-7350; TTP-0017-6049-7604
- TTP ID: TTP-0017-6046-7568
  Tactic: TA0006 (Credential Access)

Technique: T1003.002 (OS Credential Dumping: Security Account Manager)
Procedure: `O invasor coleta hashes de senhas.`
TTP Chain: TTP-0017-6049-5413; TTP-0017-6049-7247; TTP-0017-6049-7350; TTP-0017-6049-7604

- TTP ID: TTP-0017-6049-5413
  Tactic: TA0009 (Collection)
  Technique: T1113 (Screen Capture)
  Procedure: `O invasor coleta capturas de tela.`
  TTP Chain: TTP-0017-6049-8712
- TTP ID: TTP-0017-6049-7247
  Tactic: TA0009 (Collection)
  Technique: T1115 (Clipboard Data)
  Procedure: `O invasor coleta dados da área de transferência do usuário.`
  TTP Chain: TTP-0017-6049-8712
- TTP ID: TTP-0017-6049-7350
  Tactic: TA0009 (Collection)
  Technique: T1056.001 (Input Capture: Keylogging)
  Procedure: `O invasor coleta as teclas digitadas.`
  TTP Chain: TTP-0017-6049-8712
- TTP ID: TTP-0017-6049-7604
  Tactic: TA0009 (Collection)
  Technique: T1005 (Data from Local System)
  Procedure: `O invasor então coleta os arquivos que são compactados e criptografados.`
  Secondary Techniques: T1560 (Archive Collected Data)
  Related TTPs: TTP-0017-6049-8462
  TTP Chain: TTP-0017-6049-8712
- TTP ID: TTP-0017-6049-8462
  Tactic: TA0009 (Collection)
  Technique: T1560 (Archive Collected Data)
  Procedure: `Os arquivos coletados são compactados e criptografados antes de serem exfiltrados.`
  Related TTPs: TTP-0017-6049-7604
- TTP ID: TTP-0017-6049-8712
  Tactic: TA0010 (Exfiltration)
  Technique: T1048 (Exfiltration Over Alternative Protocol)

Procedure: `Os arquivos coletados são exfiltrados para um compartilhamento WebDAV controlado pelo invasor.`
TTP Chain: TTP-0017-6049-9015

- TTP ID: TTP-0017-6049-9015
  Tactic: TA0007 (Discovery)
  Technique: T1018 (Remote System Discovery)
  Procedure: `O invasor usa consultas LDAP (Lightweight Directory Access Protocol) para enumerar outros hosts no domínio.`
  TTP Chain: TTP-0017-6049-9628
- TTP ID: TTP-0017-6049-9628
  Tactic: TA0008 (Lateral Movement)
  Technique: T1021.006 (Remote Services: Windows Remote Management)
  Procedure: `O invasor cria uma sessão remota do PowerShell para uma vítima secundária via Windows Remote Management.`
  TTP Chain: TTP-0017-6049-9772
- TTP ID: TTP-0017-6049-9772
  Tactic: TA0007 (Discovery)
  Technique: T1057 (Process Discovery)
  Procedure: `Por meio dessa conexão, o invasor enumera os processos em execução.`
  TTP Chain: TTP-0017-6049-9887
- TTP ID: TTP-0017-6049-9887
  Tactic: TA0011 (Command and Control)
  Technique: T1105 (Ingress Tool Transfer)
  Procedure: `O invasor carrega um novo payload compactado com UPX para a vítima secundária.`
  Secondary Techniques: T1027.002 (Obfuscated Files or Information: Software Packing)
  TTP Chain: TTP-0017-6050-0048
- TTP ID: TTP-0017-6050-0048
  Tactic: TA0002 (Execution)
  Technique: T1569.002 (System Services: Service Execution)
  Procedure: `Esse novo payload é executado na vítima secundária por meio do utilitário PSExec, usando as credenciais roubadas anteriormente.`
  Secondary Techniques: T1021.002 (Remote Services: SMB/Windows Admin Shares); T1078.002 (Valid Accounts: Domain Accounts)
  TTP Chain: TTP-0017-6050-0265
- TTP ID: TTP-0017-6050-0265

Tactic: TA0011 (Command and Control)
Technique: T1105 (Ingress Tool Transfer)
Procedure: `O invasor carrega utilitários adicionais para a vítima secundária.`
TTP Chain: TTP-0017-6053-3303

- TTP ID: TTP-0017-6053-3303
Tactic: TA0007 (Discovery)
Technique: T1083 (File and Directory Discovery)
Procedure: `Executa um comando de linha única do PowerShell para procurar arquivos de documentos e mídia no sistema de arquivos.`
Secondary Techniques: T1059.001 (Command and Scripting Interpreter: PowerShell)
TTP Chain: TTP-0017-6053-5026
- TTP ID: TTP-0017-6053-5026
Tactic: TA0009 (Collection)
Technique: T1560.001 (Archive Collected Data: Archive via Utility)
Procedure: `Os arquivos de interesse são coletados, criptografados e compactados em um único arquivo.`
Secondary Techniques: T1005 (Data from Local System); T1074.001 (Data Staged: Local Data Staging)
TTP Chain: TTP-0017-6053-5295
- TTP ID: TTP-0017-6053-5295
Tactic: TA0010 (Exfiltration)
Technique: T1041 (Exfiltration Over C2 Channel)
Procedure: `O arquivo é exfiltrado pela conexão C2 existente.`
TTP Chain: TTP-0017-6053-5447
- TTP ID: TTP-0017-6053-5447
Tactic: TA0005 (Defense Evasion)
Technique: T1070.004 (Indicator Removal: File Deletion)
Procedure: `O invasor exclui vários arquivos associados a esse acesso.`
- TTP ID: TTP-0017-6053-5618
Tactic: TA0002 (Execution)
Technique: T1569.002 (System Services: Service Execution)
Procedure: `A vítima original é reiniciada e o usuário legítimo faz login, acionando os mecanismos de persistência previamente estabelecidos, dentre eles a execução do serviço criado "javamtsup".`
TTP Chain: TTP-0017-6046-5407
- TTP ID: TTP-0017-6053-5786

Tactic: TA0003 (Persistence)
Technique: T1547.002 (Boot or Logon Autostart Execution: Authentication Package)
Procedure: `A vítima original é reiniciada e o usuário legítimo faz log⌋ in, acionando os mecanismos de persistência previamente estabelecidos, dentre eles a execução o payload na pasta Inicialização do Windows, qu⌋ e executa um payload subsequente usando um token roubado.`
Secondary Techniques: T1134.002 (Access Token Manipulation: Create Process with Token)
TTP Chain: TTP-0017-6046-5407

Apêndice D

Regras de Detecção do APT29

A lista a seguir apresenta as 38 Regras de Detecção produzidas no segundo estágio da metodologia (Mapeamento TTP–Dados), durante o estudo de caso descrito na Seção 10.3, referente ao ataque APT29. Os resultados estão organizados conforme o modelo de dados proposto, observando-se as seguintes considerações: (i) campos vazios, bem como os campos `creation_date` e `update_date`, foram propositalmente omitidos a fim de otimizar o uso do espaço; e (ii) o campo `language` foi suprimido, uma vez que todas as regras foram desenvolvidas na linguagem SparkSQL [196].

- Rule ID: DTR-0017-3103-4080
  Reference TTP: TTP-0017-3068-5003
  Description: `Detecta a execução de um arquivo suspeito com extensão '.scr' e nome contendo 'cod.', indicando uma possível tentativa de utilização da técnica RTLO (Right-to-Left Override) para disfarçar um arquivo executável como um '.doc'.`
  Covered Techniques: T1204.002 (User Execution: Malicious File); T1036.002 (Masquerading: Right-to-Left Override)
  Sources: Sysmon
  Query:
  ```
  SELECT `@timestamp`,UtcTime,Message
  FROM apt29
  WHERE Channel = "Microsoft-Windows-Sysmon/Operational"
          AND EventID = 1
          AND LOWER(ParentImage) LIKE "%explorer.exe"
          AND LOWER(Image) LIKE '%cod.%'
          AND LOWER(Image) LIKE '%.scr'
  ```

- Rule ID: DTR-0017-6092-4557
  Reference TTP: TTP-0017-6044-2942
  Description: `Detecta quando um executável suspeito (que contém 'cod.' no nome e possui extensão '.scr') estabelece uma conexão de rede na por`

ta 1234, indicando uma possível atividade de comunicação de comando e controle (C2).

Covered Techniques: T1571 (Non-Standard Port)

Sources: Sysmon

Query:
```
SELECT `@timestamp`,UtcTime,EventTime,Message
FROM apt29
WHERE Channel = "Microsoft-Windows-Sysmon/Operational"
        AND EventID = 3
        AND LOWER(DestinationPort) LIKE '1234'
        AND LOWER(Image) LIKE '%cod.%'
        AND LOWER(Image) LIKE '%.scr'
```

- Rule ID: DTR-0017-6092-7552

  Reference TTP: TTP-0017-6044-6572

  Description: Detecta quando um processo PowerShell é criado a partir de um processo 'cmd.exe' iniciado por um executável suspeito (que contém 'cod.' no nome e possui extensão '.scr'), indicando uma possível cadeia de execução maliciosa para o uso de scripts ou comandos PowerShell.

  Covered Techniques: T1059.001 (Command and Scripting Interpreter: PowerShell)

  Sources: Sysmon

  Query:
```
SELECT `@timestamp`,UtcTime,EventTime,Message
FROM apt29 a
INNER JOIN (
    SELECT ProcessGuid
    FROM apt29
    WHERE Channel = "Microsoft-Windows-Sysmon/Operational"
        AND EventID = 1
        AND LOWER(ParentImage) LIKE '%cod.%'
        AND LOWER(ParentImage) LIKE '%.scr'
        AND LOWER(Image) LIKE '%cmd.exe'
) b
ON a.ParentProcessGuid = b.ProcessGuid
WHERE Channel = "Microsoft-Windows-Sysmon/Operational"
    AND EventID = 1
    AND LOWER(Image) LIKE '%powershell.exe'
```

- Rule ID: DTR-0017-6099-4535

  Reference TTP: TTP-0017-6044-6947

  Description: Detecta quando um processo criado por um executável suspeito (que contém 'cod.' no nome e possui extensão '.scr') executa um script PowerShell que utiliza o cmdlet 'Get-ChildItem', indicando uma possível tentativa de enumeração de diretórios ou coleta de arquivos para exfiltração.

  Covered Techniques: T1083 (File and Directory Discovery); T1059.001 (Command and Scripting Interpreter: PowerShell)

  Sources: Sysmon; PowerShell

  Query:

```
SELECT b.`@timestamp`,b.EventTime,b.ScriptBlockText
FROM apt29 a
INNER JOIN (
  SELECT d.ParentProcessGuid, d.ProcessId, c.ScriptBlockText, c.EventTime,
  ↪ c.`@timestamp`
  FROM apt29 c
  INNER JOIN (
      SELECT ParentProcessGuid, ProcessGuid, ProcessId
      FROM apt29
      WHERE Channel = "Microsoft-Windows-Sysmon/Operational"
          AND EventID = 1
      ) d
  ON c.ExecutionProcessID = d.ProcessId
  WHERE c.Channel = "Microsoft-Windows-PowerShell/Operational"
          AND c.EventID = 4104
          AND LOWER(c.ScriptBlockText) LIKE "%childitem%"
) b
ON a.ProcessGuid = b.ParentProcessGuid
WHERE a.Channel = "Microsoft-Windows-Sysmon/Operational"
          AND a.EventID = 1
          AND LOWER(a.ParentImage) LIKE '%cod.%'
          AND LOWER(a.ParentImage) LIKE '%.scr'
```

- Rule ID: DTR-0017-6099-9496
  Reference TTP: TTP-0017-6044-7344
  Description: Detecta quando um processo criado por um executável malicioso (que contém 'cod.' no nome e possui extensão '.scr') executa um script PowerShell que utiliza o cmdlet 'Compress-Archive', indicando uma possível tentativa de compactação de arquivos para exfiltração.
  Covered Techniques: T1560.001 (Archive Collected Data: Archive via Utility); T1059.001 (Command and Scripting Interpreter: PowerShell)
  Sources: Sysmon; PowerShell
  Query:

```
SELECT b.`@timestamp`,b.EventTime,b.ScriptBlockText
FROM apt29 a
INNER JOIN (
  SELECT d.ParentProcessGuid, d.ProcessId, c.ScriptBlockText, c.EventTime,
  ↪ c.`@timestamp`
  FROM apt29 c
  INNER JOIN (
      SELECT ParentProcessGuid, ProcessGuid, ProcessId
      FROM apt29
      WHERE Channel = "Microsoft-Windows-Sysmon/Operational"
          AND EventID = 1
      ) d
  ON c.ExecutionProcessID = d.ProcessId
  WHERE c.Channel = "Microsoft-Windows-PowerShell/Operational"
          AND c.EventID = 4104
          AND LOWER(c.ScriptBlockText) LIKE "%compress-archive%"
) b
ON a.ProcessGuid = b.ParentProcessGuid
WHERE a.Channel = "Microsoft-Windows-Sysmon/Operational"
```

```
    AND a.EventID = 1
    AND LOWER(a.ParentImage) LIKE '%cod.%'
    AND LOWER(a.ParentImage) LIKE '%.scr'
```

- Rule ID: DTR-0017-6124-7922
  Reference TTP: TTP-0017-6044-7344
  Description: suspeito (que contém 'cod.' no nome e possui extensão '.scr'), indicando uma possível atividade de compactação ou descompactação de arquivos.
  Covered Techniques: T1560.001 (Archive Collected Data: Archive via Utility); T1074.001 (Data Staged: Local Data Staging); T1560 (Archive Collected Data)
  Sources: Sysmon
  Query:

```
SELECT `@timestamp`,UtcTime,EventTime,Message
FROM apt29 d
INNER JOIN (
    SELECT b.ProcessGuid
    FROM apt29 b
    INNER JOIN (
        SELECT ProcessGuid
        FROM apt29
        WHERE Channel = "Microsoft-Windows-Sysmon/Operational"
            AND EventID = 1
            AND LOWER(ParentImage) LIKE '%cod.%'
            AND LOWER(ParentImage) LIKE '%.scr'
    ) a
    ON b.ParentProcessGuid = a.ProcessGuid
    WHERE b.Channel = "Microsoft-Windows-Sysmon/Operational"
        AND b.EventID = 1
) c
ON d.ProcessGuid = c.ProcessGuid
WHERE d.Channel = "Microsoft-Windows-Sysmon/Operational"
  AND d.EventID = 11
  AND LOWER(d.TargetFilename) LIKE '%.zip'
```

- Rule ID: DTR-0017-6100-1386
  Reference TTP: TTP-0017-6044-7700
  Description: Detecta quando um processo criado pelo executável malicioso (que contém 'cod.' no nome e extensão '.scr') carrega um arquivo com extensão '.png'.
  Covered Techniques: T1105 (Ingress Tool Transfer)
  Sources: Sysmon
  Query:

```
SELECT b.`@timestamp`,b.UtcTime,b.EventTime,b.Message
FROM apt29 a
INNER JOIN (
    SELECT ProcessGuid, Message, UtcTime, EventTime, `@timestamp`
    FROM apt29
    WHERE Channel = "Microsoft-Windows-Sysmon/Operational"
        AND EventID = 11
        AND LOWER(TargetFilename) LIKE '%.png'
```

```
) b
ON a.ProcessGuid = b.ProcessGuid
WHERE a.Channel = "Microsoft-Windows-Sysmon/Operational"
    AND a.EventID = 1
    AND LOWER(a.Image) LIKE '%cod.%'
    AND LOWER(a.Image) LIKE '%.scr'
```

- Rule ID: DTR-0017-6100-4576
  Reference TTP: TTP-0017-6045-0689
  Description: Detecta alterações suspeitas no registro relacionadas à chave 'DelegateExecute' em caminhos de comando do Shell, indicando uma possível tentativa de execução por meio de técnicas de COM hijacking.
  Covered Techniques: T1546.015 (Event Triggered Execution: Component Object Model Hijacking)
  Sources: Sysmon
  Query:
```
SELECT `@timestamp`,UtcTime,EventTime,Message
FROM apt29
WHERE Channel = "Microsoft-Windows-Sysmon/Operational"
    AND EventID = 13
    AND LOWER(TargetObject) RLIKE '.*\\\\\\\\folder\\\\\\\\shell\\\\\\\\
    ↪  open\\\\\\\\command\\\\\\\\delegateexecute.*'
```
- Rule ID: DTR-0017-6100-6059
  Reference TTP: TTP-0017-6045-0689
  Description: Detecta a execução do processo 'sdclt.exe' com nível de integridade elevado (High), indicando uma possível tentativa de abuso de ferramenta legítima para escalação de privilégios via bypass do UAC.
  Covered Techniques: T1548.002 (Abuse Elevation Control Mechanism: Bypass User Account Control)
  Sources: Sysmon
  Query:
```
SELECT `@timestamp`,UtcTime,EventTime,Message
FROM apt29
WHERE Channel = "Microsoft-Windows-Sysmon/Operational"
    AND EventID = 1
    AND LOWER(Image) LIKE "%sdclt.exe"
    AND IntegrityLevel = "High"
```
- Rule ID: DTR-0017-6104-7163
  Reference TTP: TTP-0017-6045-2305
  Description: Detecta a exclusão de uma chave do registro relacionada a caminhos de comando do Shell ('...\folder\shell\open\command') realizada por um processo descendente de um executável suspeito (que contém 'cod.' no nome e possui extensão '.scr'), indicando tentativa de remoção de artefato usado na escalação de privilégio.
  Covered Techniques: T1112 (Modify Registry)

Sources: Sysmon

Query:
```
SELECT `@timestamp`,UtcTime,EventTime,Message
FROM apt29 d
INNER JOIN (
    SELECT b.ProcessGuid
    FROM apt29 b
    INNER JOIN (
        SELECT ProcessGuid
        FROM apt29
        WHERE Channel = "Microsoft-Windows-Sysmon/Operational"
            AND EventID = 1
            AND LOWER(ParentImage) LIKE '%cod.%'
            AND LOWER(ParentImage) LIKE '%.scr'
    ) a
    ON b.ParentProcessGuid = a.ProcessGuid
    WHERE b.Channel = "Microsoft-Windows-Sysmon/Operational"
        AND b.EventID = 1
) c
ON d.ProcessGuid = c.ProcessGuid
WHERE d.Channel = "Microsoft-Windows-Sysmon/Operational"
  AND d.EventID = 12
  AND LOWER(d.TargetObject) RLIKE
  ↪  '.*\\\\\\\\folder\\\\\\\\shell\\\\\\\\open\\\\\\\\command.*'
  AND d.Message RLIKE '.*EventType: DeleteKey.*'
```

- Rule ID: DTR-0017-6104-8503

  Reference TTP: TTP-0017-6045-2548

  Description: Detecta a criação de arquivos .zip por um processo executado com nível de integridade elevado (High), iniciado a partir de control.exe, o qual foi lançado por sdclt.exe.

  Covered Techniques: T1105 (Ingress Tool Transfer)

  Sources: Sysmon

  Query:
```
SELECT `@timestamp`,UtcTime,EventTime,Message
FROM apt29 d
INNER JOIN (
  SELECT a.ProcessGuid, a.ParentProcessGuid
  FROM apt29 a
  INNER JOIN (
    SELECT ProcessGuid
    FROM apt29
    WHERE Channel = "Microsoft-Windows-Sysmon/Operational"
        AND EventID = 1
        AND LOWER(Image) LIKE "%control.exe"
        AND LOWER(ParentImage) LIKE "%sdclt.exe"
  ) b
  ON a.ParentProcessGuid = b.ProcessGuid
  WHERE a.Channel = "Microsoft-Windows-Sysmon/Operational"
    AND a.EventID = 1
    AND a.IntegrityLevel = "High"
) c
ON d.ProcessGuid = c.ProcessGuid
WHERE d.Channel = "Microsoft-Windows-Sysmon/Operational"
```

```
    AND d.EventID = 11
    AND LOWER(d.TargetFilename) LIKE '%.zip'
```

- Rule ID: DTR-0017-6104-9721
  Reference TTP: TTP-0017-6045-8979
  Description: Detecta a execução de um script PowerShell que utiliza o cmdlet 'Expand-Archive', executado a partir de um processo 'control.exe' iniciado por 'sdclt.exe' com nível de integridade elevado (High), indicando uma possível tentativa de descompactação de arquivos após escalação de privilégios.
  Covered Techniques: T1140 (Deobfuscate/Decode Files or Information); T1059.001 (Command and Scripting Interpreter: PowerShell)
  Sources: Sysmon; PowerShell
  Query:

```
SELECT `@timestamp`,EventTime,Payload
FROM apt29 f
INNER JOIN (
    SELECT d.ProcessId
    FROM apt29 d
    INNER JOIN (
      SELECT a.ProcessGuid, a.ParentProcessGuid
      FROM apt29 a
      INNER JOIN (
        SELECT ProcessGuid
        FROM apt29
        WHERE Channel = "Microsoft-Windows-Sysmon/Operational"
            AND EventID = 1
            AND LOWER(Image) LIKE "%control.exe"
            AND LOWER(ParentImage) LIKE "%sdclt.exe"
      ) b
      ON a.ParentProcessGuid = b.ProcessGuid
      WHERE a.Channel = "Microsoft-Windows-Sysmon/Operational"
        AND a.EventID = 1
        AND a.IntegrityLevel = "High"
    ) c
    ON d.ParentProcessGuid= c.ProcessGuid
    WHERE d.Channel = "Microsoft-Windows-Sysmon/Operational"
      AND d.EventID = 1
      AND d.Image LIKE '%powershell.exe'
) e
ON f.ExecutionProcessID = e.ProcessId
WHERE f.Channel = "Microsoft-Windows-PowerShell/Operational"
    AND f.EventID = 4103
    AND LOWER(f.Payload) LIKE "%expand-archive%"
```

- Rule ID: DTR-0017-6105-0296
  Reference TTP: TTP-0017-6045-9266
  Description: Detecta a execução de um script PowerShell que utiliza o cmdlet 'Get-Process', executado a partir de um processo 'control.exe' iniciado por 'sdclt.exe' com nível de integridade elevado (High), indi

cando uma possível tentativa de enumeração de processos após escalação de privilégios.

Covered Techniques: T1057 (Process Discovery); T1059.001 (Command and Scripting Interpreter: PowerShell)

Sources: Sysmon; PowerShell

Query:
```
SELECT `@timestamp`,EventTime,ScriptBlockText
FROM apt29 f
INNER JOIN (
    SELECT d.ProcessId
    FROM apt29 d
    INNER JOIN (
      SELECT a.ProcessGuid, a.ParentProcessGuid
      FROM apt29 a
      INNER JOIN (
        SELECT ProcessGuid
        FROM apt29
        WHERE Channel = "Microsoft-Windows-Sysmon/Operational"
            AND EventID = 1
            AND LOWER(Image) LIKE "%control.exe"
            AND LOWER(ParentImage) LIKE "%sdclt.exe"
      ) b
      ON a.ParentProcessGuid = b.ProcessGuid
      WHERE a.Channel = "Microsoft-Windows-Sysmon/Operational"
        AND a.EventID = 1
        AND a.IntegrityLevel = "High"
    ) c
    ON d.ParentProcessGuid= c.ProcessGuid
    WHERE d.Channel = "Microsoft-Windows-Sysmon/Operational"
      AND d.EventID = 1
      AND d.Image LIKE '%powershell.exe'
) e
ON f.ExecutionProcessID = e.ProcessId
WHERE f.Channel = "Microsoft-Windows-PowerShell/Operational"
    AND f.EventID = 4104
    AND LOWER(f.ScriptBlockText) LIKE "%get-process%"
```

- Rule ID: DTR-0017-6105-0810

  Reference TTP: TTP-0017-6046-1439

  Description: Detecta a execução do utilitário 'sdelete' para exclusão de arquivos 'draft.zip', 'sysinternalssuite.zip' ou que contenham 'cod.' no nome extensão '.scr', iniciada por um processo PowerShell originado da cadeia de execução envolvendo 'sdclt.exe' e 'control.exe' com nível de integridade elevado (High), indicando uma possível tentativa de exclusão segura de arquivos maliciosos para ocultação de rastros.

  Covered Techniques: T1070.004 (Indicator Removal: File Deletion)

  Sources: Sysmon

  Query:
```
SELECT `@timestamp`,UtcTime,EventTime,Message, g.CommandLine
FROM apt29 h
INNER JOIN (
```

```
SELECT f.ProcessGuid, f.CommandLine
FROM apt29 f
INNER JOIN (
  SELECT d.ProcessId, d.ProcessGuid
  FROM apt29 d
  INNER JOIN (
    SELECT a.ProcessGuid, a.ParentProcessGuid
    FROM apt29 a
    INNER JOIN (
      SELECT ProcessGuid
      FROM apt29
      WHERE Channel = "Microsoft-Windows-Sysmon/Operational"
          AND EventID = 1
          AND LOWER(Image) LIKE "%control.exe"
          AND LOWER(ParentImage) LIKE "%sdclt.exe"
    ) b
    ON a.ParentProcessGuid = b.ProcessGuid
    WHERE a.Channel = "Microsoft-Windows-Sysmon/Operational"
      AND a.EventID = 1
      AND a.IntegrityLevel = "High"
  ) c
  ON d.ParentProcessGuid = c.ProcessGuid
  WHERE d.Channel = "Microsoft-Windows-Sysmon/Operational"
    AND d.EventID = 1
    AND d.Image LIKE '%powershell.exe'
) e
ON f.ParentProcessGuid = e.ProcessGuid
WHERE f.Channel = "Microsoft-Windows-Sysmon/Operational"
  AND f.EventID = 1
  AND LOWER(f.Image) LIKE '%sdelete%'
  AND (
      LOWER(f.CommandLine) LIKE '%draft.zip%'
      OR LOWER(f.CommandLine) LIKE '%sysinternalssuite.zip%'
      OR (
          LOWER(f.CommandLine) LIKE '%cod.%'
          AND LOWER(f.CommandLine) LIKE '%.scr'
      )
  )
) g
ON h.ProcessGuid = g.ProcessGuid
WHERE h.Channel = "Microsoft-Windows-Sysmon/Operational"
  AND h.EventID = 23
```

- Rule ID: DTR-0017-6105-2497
  Reference TTP: TTP-0017-6046-1657
  Description: `Detecta a execução de um script PowerShell que faz referência ao diretório '%TEMP%', executado a partir de um processo 'control.exe' iniciado por 'sdclt.exe' com nível de integridade elevado (High), indicando possível enumeração do caminho do diretório temporário do usuário.`
  Covered Techniques: T1083 (File and Directory Discovery); T1059.001 (Command and Scripting Interpreter: PowerShell)

Sources: Sysmon; PowerShell

Query:
```
SELECT `@timestamp`,EventTime,ScriptBlockText
FROM apt29 f
INNER JOIN (
  SELECT d.ProcessId
  FROM apt29 d
  INNER JOIN (
    SELECT a.ProcessGuid, a.ParentProcessGuid
    FROM apt29 a
    INNER JOIN (
      SELECT ProcessGuid
      FROM apt29
      WHERE Channel = "Microsoft-Windows-Sysmon/Operational"
          AND EventID = 1
          AND LOWER(Image) LIKE "%control.exe"
          AND LOWER(ParentImage) LIKE "%sdclt.exe"
    ) b
    ON a.ParentProcessGuid = b.ProcessGuid
    WHERE a.Channel = "Microsoft-Windows-Sysmon/Operational"
      AND a.EventID = 1
      AND a.IntegrityLevel = "High"
  ) c
  ON d.ParentProcessGuid= c.ProcessGuid
  WHERE d.Channel = "Microsoft-Windows-Sysmon/Operational"
    AND d.EventID = 1
    AND d.Image LIKE '%powershell.exe'
) e
ON f.ExecutionProcessID = e.ProcessId
WHERE f.Channel = "Microsoft-Windows-PowerShell/Operational"
  AND f.EventID = 4104
  AND LOWER(f.ScriptBlockText) LIKE "%$env:temp%"
```

- Rule ID: DTR-0017-6105-3592

  Reference TTP: TTP-0017-6046-2409

  Description: Detecta a execução de um script PowerShell que faz referência à variável de ambiente '$env:USERNAME', executado a partir de um processo 'control.exe' iniciado por 'sdclt.exe' com nível de integridade elevado (High), indicando possível enumeração do nome de usuário.

  Covered Techniques: T1033 (System Owner/User Discovery); T1059.001 (Command and Scripting Interpreter: PowerShell)

  Sources: Sysmon; PowerShell

  Query:
```
SELECT `@timestamp`,EventTime,ScriptBlockText
FROM apt29 f
INNER JOIN (
  SELECT d.ProcessId
  FROM apt29 d
  INNER JOIN (
    SELECT a.ProcessGuid, a.ParentProcessGuid
    FROM apt29 a
    INNER JOIN (
      SELECT ProcessGuid
```

```
      FROM apt29
      WHERE Channel = "Microsoft-Windows-Sysmon/Operational"
          AND EventID = 1
          AND LOWER(Image) LIKE "%control.exe"
          AND LOWER(ParentImage) LIKE "%sdclt.exe"
    ) b
    ON a.ParentProcessGuid = b.ProcessGuid
    WHERE a.Channel = "Microsoft-Windows-Sysmon/Operational"
      AND a.EventID = 1
      AND a.IntegrityLevel = "High"
  ) c
  ON d.ParentProcessGuid= c.ProcessGuid
  WHERE d.Channel = "Microsoft-Windows-Sysmon/Operational"
    AND d.EventID = 1
    AND d.Image LIKE '%powershell.exe'
) e
ON f.ExecutionProcessID = e.ProcessId
WHERE f.Channel = "Microsoft-Windows-PowerShell/Operational"
  AND f.EventID = 4104
  AND LOWER(f.ScriptBlockText) LIKE "%$env:username%"
```

- Rule ID: DTR-0017-6105-5552
  Reference TTP: TTP-0017-6046-2437
  Description: Detecta a execução de um script PowerShell que faz referência à variável '$env:COMPUTERNAME' ou utiliza o cmdlet 'gwmi Win32_OperatingSystem', executado a partir de um processo 'control.exe' iniciado por 'sdclt.exe' com nível de integridade elevado (High), indicando possível enumeração do nome do computador ou da versão do Sistema Operacional.
  Covered Techniques: T1082 (System Information Discovery); T1059.001 (Command and Scripting Interpreter: PowerShell)
  Sources: Sysmon; PowerShell
  Query:

```
SELECT `@timestamp`,EventTime,ScriptBlockText
FROM apt29 f
INNER JOIN (
  SELECT d.ProcessId
  FROM apt29 d
  INNER JOIN (
    SELECT a.ProcessGuid, a.ParentProcessGuid
    FROM apt29 a
    INNER JOIN (
      SELECT ProcessGuid
      FROM apt29
      WHERE Channel = "Microsoft-Windows-Sysmon/Operational"
          AND EventID = 1
          AND LOWER(Image) LIKE "%control.exe"
          AND LOWER(ParentImage) LIKE "%sdclt.exe"
    ) b
    ON a.ParentProcessGuid = b.ProcessGuid
    WHERE a.Channel = "Microsoft-Windows-Sysmon/Operational"
```

```
      AND a.EventID = 1
      AND a.IntegrityLevel = "High"
  ) c
  ON d.ParentProcessGuid= c.ProcessGuid
  WHERE d.Channel = "Microsoft-Windows-Sysmon/Operational"
    AND d.EventID = 1
    AND d.Image LIKE '%powershell.exe'
) e
ON f.ExecutionProcessID = e.ProcessId
WHERE f.Channel = "Microsoft-Windows-PowerShell/Operational"
  AND f.EventID = 4104
  AND (
    LOWER(f.ScriptBlockText) LIKE "%$env:computername%"
    OR LOWER(f.ScriptBlockText) LIKE "%gwmi win32_operatingsystem%"
    )
```

- Rule ID: DTR-0017-6105-6366
  Reference TTP: TTP-0017-6046-2449
  Description: Detecta a execução de um script PowerShell que faz referência à variável '$env:USERDOMAIN', executado a partir de um processo 'control.exe' iniciado por 'sdclt.exe' com nível de integridade elevado (High), indicando possível enumeração do nome do domínio.
  Covered Techniques: T1016 (System Network Configuration Discovery); T1059.001 (Command and Scripting Interpreter: PowerShell)
  Sources: Sysmon; PowerShell
  Query:

```
SELECT `@timestamp`,EventTime,ScriptBlockText
FROM apt29 f
INNER JOIN (
  SELECT d.ProcessId
  FROM apt29 d
  INNER JOIN (
    SELECT a.ProcessGuid, a.ParentProcessGuid
    FROM apt29 a
    INNER JOIN (
      SELECT ProcessGuid
      FROM apt29
      WHERE Channel = "Microsoft-Windows-Sysmon/Operational"
          AND EventID = 1
          AND LOWER(Image) LIKE "%control.exe"
          AND LOWER(ParentImage) LIKE "%sdclt.exe"
    ) b
    ON a.ParentProcessGuid = b.ProcessGuid
    WHERE a.Channel = "Microsoft-Windows-Sysmon/Operational"
      AND a.EventID = 1
      AND a.IntegrityLevel = "High"
  ) c
  ON d.ParentProcessGuid= c.ProcessGuid
  WHERE d.Channel = "Microsoft-Windows-Sysmon/Operational"
    AND d.EventID = 1
    AND d.Image LIKE '%powershell.exe'
) e
```

```
ON f.ExecutionProcessID = e.ProcessId
WHERE f.Channel = "Microsoft-Windows-PowerShell/Operational"
  AND f.EventID = 4104
  AND LOWER(f.ScriptBlockText) LIKE "%$env:userdomain%"
```

- Rule ID: DTR-0017-6105-6754
  Reference TTP: TTP-0017-6046-2450
  Description: Detecta a execução de um script PowerShell que faz referência à variável '$PID', executado a partir de um processo 'control.exe' iniciado por 'sdclt.exe' com nível de integridade elevado (High), indicando possível enumeração de processos ou auto-inspeção do identificador de processo.
  Covered Techniques: T1057 (Process Discovery); T1059.001 (Command and Scripting Interpreter: PowerShell)
  Sources: Sysmon; PowerShell
  Query:
```
SELECT `@timestamp`,EventTime,ScriptBlockText
FROM apt29 f
INNER JOIN (
  SELECT d.ProcessId
  FROM apt29 d
  INNER JOIN (
    SELECT a.ProcessGuid, a.ParentProcessGuid
    FROM apt29 a
    INNER JOIN (
      SELECT ProcessGuid
      FROM apt29
      WHERE Channel = "Microsoft-Windows-Sysmon/Operational"
          AND EventID = 1
          AND LOWER(Image) LIKE "%control.exe"
          AND LOWER(ParentImage) LIKE "%sdclt.exe"
    ) b
    ON a.ParentProcessGuid = b.ProcessGuid
    WHERE a.Channel = "Microsoft-Windows-Sysmon/Operational"
      AND a.EventID = 1
      AND a.IntegrityLevel = "High"
  ) c
  ON d.ParentProcessGuid= c.ProcessGuid
  WHERE d.Channel = "Microsoft-Windows-Sysmon/Operational"
    AND d.EventID = 1
    AND d.Image LIKE '%powershell.exe'
) e
ON f.ExecutionProcessID = e.ProcessId
WHERE f.Channel = "Microsoft-Windows-PowerShell/Operational"
  AND f.EventID = 4104
  AND LOWER(f.ScriptBlockText) LIKE "%$pid%"
```

- Rule ID: DTR-0017-6105-7294
  Reference TTP: TTP-0017-6046-2453
  Description: Detecta a execução de um script PowerShell que consulta classes WMI '-Class AntivirusProduct' ou '-Class FirewallProduct', execut

ado a partir de um processo 'control.exe' iniciado por 'sdclt.exe' com nível de integridade elevado (High), indicando possível reconhecimento de produtos de segurança instalados no sistema.
Covered Techniques: T1518.001 (Software Discovery: Security Software Discovery); T1059.001 (Command and Scripting Interpreter: PowerShell)
Sources: Sysmon; PowerShell
Query:
```
SELECT `@timestamp`,EventTime,ScriptBlockText
FROM apt29 f
INNER JOIN (
  SELECT d.ProcessId
  FROM apt29 d
  INNER JOIN (
    SELECT a.ProcessGuid, a.ParentProcessGuid
    FROM apt29 a
    INNER JOIN (
      SELECT ProcessGuid
      FROM apt29
      WHERE Channel = "Microsoft-Windows-Sysmon/Operational"
          AND EventID = 1
          AND LOWER(Image) LIKE "%control.exe"
          AND LOWER(ParentImage) LIKE "%sdclt.exe"
    ) b
    ON a.ParentProcessGuid = b.ProcessGuid
    WHERE a.Channel = "Microsoft-Windows-Sysmon/Operational"
      AND a.EventID = 1
      AND a.IntegrityLevel = "High"
  ) c
  ON d.ParentProcessGuid= c.ProcessGuid
  WHERE d.Channel = "Microsoft-Windows-Sysmon/Operational"
    AND d.EventID = 1
    AND d.Image LIKE '%powershell.exe'
) e
ON f.ExecutionProcessID = e.ProcessId
WHERE f.Channel = "Microsoft-Windows-PowerShell/Operational"
  AND f.EventID = 4104
  AND (
    LOWER(f.ScriptBlockText) LIKE "%-class antivirusproduct%"
    OR LOWER(f.ScriptBlockText) LIKE "%-class firewallproduct%"
    )
```

- Rule ID: DTR-0017-6106-2801
  Reference TTP: TTP-0017-6046-2456
  Description: Detecta a execução de um script PowerShell que utiliza os cmdlets 'NetUserGetGroups' ou 'NetUserGetLocalGroups', executado a partir de um processo 'control.exe' iniciado por 'sdclt.exe' com nível de integridade elevado (High), indicando possível enumeração de grupos de usuários no sistema.
  Covered Techniques: T1069 (Permission Groups Discovery); T1059.001 (Command and Scripting Interpreter: PowerShell); T1106 (Native API)

Sources: Sysmon; PowerShell

Query:
```
SELECT `@timestamp`,EventTime,ScriptBlockText
FROM apt29 f
INNER JOIN (
  SELECT d.ProcessId
  FROM apt29 d
  INNER JOIN (
    SELECT a.ProcessGuid, a.ParentProcessGuid
    FROM apt29 a
    INNER JOIN (
      SELECT ProcessGuid
      FROM apt29
      WHERE Channel = "Microsoft-Windows-Sysmon/Operational"
          AND EventID = 1
          AND LOWER(Image) LIKE "%control.exe"
          AND LOWER(ParentImage) LIKE "%sdclt.exe"
    ) b
    ON a.ParentProcessGuid = b.ProcessGuid
    WHERE a.Channel = "Microsoft-Windows-Sysmon/Operational"
      AND a.EventID = 1
      AND a.IntegrityLevel = "High"
  ) c
  ON d.ParentProcessGuid= c.ProcessGuid
  WHERE d.Channel = "Microsoft-Windows-Sysmon/Operational"
    AND d.EventID = 1
    AND d.Image LIKE '%powershell.exe'
) e
ON f.ExecutionProcessID = e.ProcessId
WHERE f.Channel = "Microsoft-Windows-PowerShell/Operational"
  AND f.EventID = 4104
  AND (
    LOWER(f.ScriptBlockText) LIKE "%netusergetgroups%"
    OR LOWER(f.ScriptBlockText) LIKE "%netusergetlocalgroups%"
    )
```

- Rule ID: DTR-0017-6106-4244

  Reference TTP: TTP-0017-6046-4431

  Description: Detecta a execução de um script PowerShell que utiliza o cmdlet 'New-Service', executado a partir de um processo 'control.exe' iniciado por 'sdclt.exe' com nível de integridade elevado (High), indicando possível tentativa de criação de serviço para persistência.

  Covered Techniques: T1543.003 (Create or Modify System Process: Windows Service); T1059.001 (Command and Scripting Interpreter: PowerShell)

  Sources: Sysmon; PowerShell

  Query:
```
SELECT `@timestamp`,EventTime,Payload
FROM apt29 f
INNER JOIN (
  SELECT d.ProcessId, d.ParentProcessId
  FROM apt29 d
  INNER JOIN (
    SELECT a.ProcessGuid, a.ParentProcessGuid
```

```
    FROM apt29 a
    INNER JOIN (
      SELECT ProcessGuid
      FROM apt29
      WHERE Channel = "Microsoft-Windows-Sysmon/Operational"
          AND EventID = 1
          AND LOWER(Image) LIKE "%control.exe"
          AND LOWER(ParentImage) LIKE "%sdclt.exe"
    ) b
    ON a.ParentProcessGuid = b.ProcessGuid
    WHERE a.Channel = "Microsoft-Windows-Sysmon/Operational"
      AND a.EventID = 1
      AND a.IntegrityLevel = "High"
  ) c
  ON d.ParentProcessGuid= c.ProcessGuid
  WHERE d.Channel = "Microsoft-Windows-Sysmon/Operational"
    AND d.EventID = 1
    AND d.Image LIKE '%powershell.exe'
) e
ON f.ExecutionProcessID = e.ProcessId
WHERE f.Channel = "Microsoft-Windows-PowerShell/Operational"
  AND f.EventID = 4103
  AND LOWER(f.Payload) LIKE "%new-service%"
```

- Rule ID: DTR-0017-6106-5518

  Reference TTP: TTP-0017-6046-4432

  Description: Detecta a criação de arquivo em diretório de inicialização (`StartUp`) do Windows por um processo 'control.exe' iniciado por 'sdclt.exe' com nível de integridade elevado (High), indicando uma possível tentativa de estabelecer persistência.

  Covered Techniques: T1547.001 (Boot or Logon Autostart Execution: Registry Run Keys / Startup Folder)

  Sources: Sysmon

  Query:
```
SELECT `@timestamp`,UtcTime,EventTime,Message
FROM apt29 f
INNER JOIN (
    SELECT d.ProcessGuid
    FROM apt29 d
    INNER JOIN (
      SELECT a.ProcessGuid, a.ParentProcessGuid
      FROM apt29 a
      INNER JOIN (
        SELECT ProcessGuid
        FROM apt29
        WHERE Channel = "Microsoft-Windows-Sysmon/Operational"
            AND EventID = 1
            AND LOWER(Image) LIKE "%control.exe"
            AND LOWER(ParentImage) LIKE "%sdclt.exe"
      ) b
      ON a.ParentProcessGuid = b.ProcessGuid
      WHERE a.Channel = "Microsoft-Windows-Sysmon/Operational"
```

```
        AND a.EventID = 1
        AND a.IntegrityLevel = "High"
    ) c
    ON d.ParentProcessGuid= c.ProcessGuid
    WHERE d.Channel = "Microsoft-Windows-Sysmon/Operational"
      AND d.EventID = 1
      AND d.Image LIKE '%powershell.exe'
) e
ON f.ProcessGuid = e.ProcessGuid
WHERE f.Channel = "Microsoft-Windows-Sysmon/Operational"
    AND f.EventID = 11
    AND f.TargetFilename RLIKE
    ↪ '.*\\\\\\\\ProgramData\\\\\\\\Microsoft\\\\\\\\Windows\\\\\\\\Start
    ↪ Menu\\\\\\\\Programs\\\\\\\\StartUp.*'
```

- Rule ID: DTR-0017-6106-5995
  Reference TTP: TTP-0017-6046-5816
  Description: Detecta o carregamento do utilitário de nome 'accesschk' (mesmo nome de ferramenta legítma do Sysinternals) em um processo PowerShell iniciado por 'control.exe', que por sua vez foi executado a partir de 'sdclt.exe' com nível de integridade elevado (High).
  Covered Techniques: T1036.005 (Masquerading: Match Legitimate Name or Location)
  Sources: Sysmon
  Query:

```
SELECT `@timestamp`,UtcTime,EventTime,Message
FROM apt29 h
INNER JOIN (
    SELECT f.ProcessGuid
    FROM apt29 f
    INNER JOIN (
      SELECT d.ProcessGuid, d.ParentProcessGuid
      FROM apt29 d
      INNER JOIN (
        SELECT a.ProcessGuid, a.ParentProcessGuid
        FROM apt29 a
        INNER JOIN (
          SELECT ProcessGuid
          FROM apt29
          WHERE Channel = "Microsoft-Windows-Sysmon/Operational"
              AND EventID = 1
              AND LOWER(Image) LIKE "%control.exe"
              AND LOWER(ParentImage) LIKE "%sdclt.exe"
        ) b
        ON a.ParentProcessGuid = b.ProcessGuid
        WHERE a.Channel = "Microsoft-Windows-Sysmon/Operational"
          AND a.EventID = 1
          AND a.IntegrityLevel = "High"
      ) c
      ON d.ParentProcessGuid= c.ProcessGuid
      WHERE d.Channel = "Microsoft-Windows-Sysmon/Operational"
        AND d.EventID = 1
        AND d.Image LIKE '%powershell.exe'
    ) e
```

```
    ON f.ParentProcessGuid = e.ProcessGuid
    WHERE f.Channel = "Microsoft-Windows-Sysmon/Operational"
      AND f.EventID = 1
      AND LOWER(f.Image) LIKE '%accesschk%'
) g
ON h.ProcessGuid = g.ProcessGuid
WHERE h.Channel = "Microsoft-Windows-Sysmon/Operational"
    AND EventID = 7
    AND LOWER(ImageLoaded) LIKE '%accesschk%'
```

- Rule ID: DTR-0017-6106-6816
  Reference TTP: TTP-0017-6049-5413
  Description: Detecta o carregamento da biblioteca 'System.Drawing.ni.dll' por um processo PowerShell iniciado por 'control.exe', o qual foi executado a partir de 'sdclt.exe' com nível de integridade elevado (High), indicando possível uso de componentes gráficos .NET para captura de tela.
  Covered Techniques: T1113 (Screen Capture); T1106 (Native API)
  Sources: Sysmon
  Query:

```
SELECT `@timestamp`,UtcTime,EventTime,Message
FROM apt29 f
INNER JOIN (
    SELECT d.ProcessGuid, d.ParentProcessGuid
    FROM apt29 d
    INNER JOIN (
      SELECT a.ProcessGuid, a.ParentProcessGuid
      FROM apt29 a
      INNER JOIN (
        SELECT ProcessGuid
        FROM apt29
        WHERE Channel = "Microsoft-Windows-Sysmon/Operational"
            AND EventID = 1
            AND LOWER(Image) LIKE "%control.exe"
            AND LOWER(ParentImage) LIKE "%sdclt.exe"
      ) b
      ON a.ParentProcessGuid = b.ProcessGuid
      WHERE a.Channel = "Microsoft-Windows-Sysmon/Operational"
        AND a.EventID = 1
        AND a.IntegrityLevel = "High"
    ) c
    ON d.ParentProcessGuid= c.ProcessGuid
    WHERE d.Channel = "Microsoft-Windows-Sysmon/Operational"
      AND d.EventID = 1
      AND d.Image LIKE '%powershell.exe'
) e
ON f.ProcessGuid = e.ProcessGuid
WHERE f.Channel = "Microsoft-Windows-Sysmon/Operational"
    AND f.EventID = 7
    AND LOWER(f.ImageLoaded) LIKE "%system.drawing.ni.dll"
```

- Rule ID: DTR-0017-6106-7870

Reference TTP: TTP-0017-6049-5413

Description: Detecta a execução de comando PowerShell contendo a função `CopyFromScreen`, executado por um processo `powershell.exe` iniciado por `control.exe`, o qual, por sua vez, foi executado a partir de `sdclt.exe` com nível de integridade elevado (High), indicando possível captura de tela do sistema.

Covered Techniques: T1113 (Screen Capture); T1059.001 (Command and Scripting Interpreter: PowerShell)

Sources: Sysmon; PowerShell

Query:
```
SELECT `@timestamp`,EventTime,Payload
FROM apt29 f
INNER JOIN (
    SELECT d.ProcessId, d.ParentProcessId
    FROM apt29 d
    INNER JOIN (
      SELECT a.ProcessGuid, a.ParentProcessGuid
      FROM apt29 a
      INNER JOIN (
        SELECT ProcessGuid
        FROM apt29
        WHERE Channel = "Microsoft-Windows-Sysmon/Operational"
            AND EventID = 1
            AND LOWER(Image) LIKE "%control.exe"
            AND LOWER(ParentImage) LIKE "%sdclt.exe"
      ) b
      ON a.ParentProcessGuid = b.ProcessGuid
      WHERE a.Channel = "Microsoft-Windows-Sysmon/Operational"
        AND a.EventID = 1
        AND a.IntegrityLevel = "High"
    ) c
    ON d.ParentProcessGuid= c.ProcessGuid
    WHERE d.Channel = "Microsoft-Windows-Sysmon/Operational"
      AND d.EventID = 1
      AND d.Image LIKE '%powershell.exe'
) e
ON f.ExecutionProcessID = e.ProcessId
WHERE f.Channel = "Microsoft-Windows-PowerShell/Operational"
    AND f.EventID = 4103
    AND LOWER(f.Payload) LIKE "%copyfromscreen%"
```

- Rule ID: DTR-0017-6106-7534

  Reference TTP: TTP-0017-6049-7247

  Description: Detecta a execução de um comando PowerShell que utiliza 'Get-Clipboard', executado a partir de um processo 'control.exe' iniciado por 'sdclt.exe' com nível de integridade elevado (High), indicando possível captura de dados do clipboard do sistema.

  Covered Techniques: T1115 (Clipboard Data); T1059.001 (Command and Scripting Interpreter: PowerShell)

Sources: Sysmon; PowerShell

Query:
```
SELECT `@timestamp`,EventTime,Payload
FROM apt29 f
INNER JOIN (
  SELECT d.ProcessId, d.ParentProcessId
  FROM apt29 d
  INNER JOIN (
    SELECT a.ProcessGuid, a.ParentProcessGuid
    FROM apt29 a
    INNER JOIN (
      SELECT ProcessGuid
      FROM apt29
      WHERE Channel = "Microsoft-Windows-Sysmon/Operational"
          AND EventID = 1
          AND LOWER(Image) LIKE "%control.exe"
          AND LOWER(ParentImage) LIKE "%sdclt.exe"
    ) b
    ON a.ParentProcessGuid = b.ProcessGuid
    WHERE a.Channel = "Microsoft-Windows-Sysmon/Operational"
      AND a.EventID = 1
      AND a.IntegrityLevel = "High"
  ) c
  ON d.ParentProcessGuid= c.ProcessGuid
  WHERE d.Channel = "Microsoft-Windows-Sysmon/Operational"
    AND d.EventID = 1
    AND d.Image LIKE '%powershell.exe'
) e
ON f.ExecutionProcessID = e.ProcessId
WHERE f.Channel = "Microsoft-Windows-PowerShell/Operational"
    AND f.EventID = 4103
    AND LOWER(f.Payload) LIKE "%get-clipboard%"
```

- Rule ID: DTR-0017-6106-8933

  Reference TTP: TTP-0017-6049-8462

  Description: Detecta a execução de um script PowerShell que utiliza a função 'compress-7zip', executado a partir de um processo 'control.exe' iniciado por 'sdclt.exe' com nível de integridade elevado (High), indicando uma possível tentativa de compactação de arquivos com 7-Zip para preparação de exfiltração.

  Covered Techniques: T1560 (Archive Collected Data); T1059.001 (Command and Scripting Interpreter: PowerShell)

  Sources: Sysmon; PowerShell

  Query:
```
SELECT `@timestamp`,EventTime,f.ScriptBlockText
FROM apt29 f
INNER JOIN (
  SELECT d.ProcessId, d.ParentProcessId
  FROM apt29 d
  INNER JOIN (
    SELECT a.ProcessGuid, a.ParentProcessGuid
    FROM apt29 a
    INNER JOIN (
```

```
      SELECT ProcessGuid
      FROM apt29
      WHERE Channel = "Microsoft-Windows-Sysmon/Operational"
          AND EventID = 1
          AND LOWER(Image) LIKE "%control.exe"
          AND LOWER(ParentImage) LIKE "%sdclt.exe"
    ) b
    ON a.ParentProcessGuid = b.ProcessGuid
    WHERE a.Channel = "Microsoft-Windows-Sysmon/Operational"
      AND a.EventID = 1
      AND a.IntegrityLevel = "High"
  ) c
  ON d.ParentProcessGuid= c.ProcessGuid
  WHERE d.Channel = "Microsoft-Windows-Sysmon/Operational"
    AND d.EventID = 1
    AND d.Image LIKE '%powershell.exe'
) e
ON f.ExecutionProcessID = e.ProcessId
WHERE f.Channel = "Microsoft-Windows-PowerShell/Operational"
    AND f.EventID = 4104
    AND LOWER(f.ScriptBlockText) LIKE "%compress-7zip%"
```

- Rule ID: DTR-0017-6106-9592
  Reference TTP: TTP-0017-6049-8712
  Description: Detecta a invocação de `rundll32.exe` chamando `davclnt.dll::DavSetCookie`, indicando possível uso do cliente WebDAV. Essa atividade pode ser usada para manipulação de sessões WebDAV ou para exfiltração via WebDAV.
  Covered Techniques: T1048 (Exfiltration Over Alternative Protocol); T1218.011 (System Binary Proxy Execution: Rundll32)
  Sources: Sysmon
  Query:
```
SELECT `@timestamp`,UtcTime,EventTime,Message
FROM apt29
WHERE Channel = "Microsoft-Windows-Sysmon/Operational"
    AND EventID = 1
    AND CommandLine RLIKE '.*rundll32.exe.*\\\\\\\\windows\\\\\\\\
    ↪  system32\\\\\\\\davclnt.dll.*DavSetCookie.*'
```

- Rule ID: DTR-0017-6107-0164
  Reference TTP: TTP-0017-6049-9015
  Description: Detecta conexões de rede na porta 389 (LDAP) realizadas por um processo PowerShell iniciado por 'control.exe', o qual foi executado a partir de 'sdclt.exe' com nível de integridade elevado (High).
  Covered Techniques: T1018 (Remote System Discovery); T1482 (Domain Trust Discovery)
  Sources: Sysmon
  Query:
```
SELECT `@timestamp`,UtcTime,EventTime,Message
FROM apt29 f
```

```
INNER JOIN (
SELECT d.ProcessId, d.ParentProcessId
FROM apt29 d
INNER JOIN (
  SELECT a.ProcessGuid, a.ParentProcessGuid
  FROM apt29 a
  INNER JOIN (
    SELECT ProcessGuid
    FROM apt29
    WHERE Channel = "Microsoft-Windows-Sysmon/Operational"
        AND EventID = 1
        AND LOWER(Image) LIKE "%control.exe"
        AND LOWER(ParentImage) LIKE "%sdclt.exe"
  ) b
  ON a.ParentProcessGuid = b.ProcessGuid
  WHERE a.Channel = "Microsoft-Windows-Sysmon/Operational"
    AND a.EventID = 1
    AND a.IntegrityLevel = "High"
) c
ON d.ParentProcessGuid= c.ProcessGuid
WHERE d.Channel = "Microsoft-Windows-Sysmon/Operational"
  AND d.EventID = 1
  AND d.Image LIKE '%powershell.exe'
) e
ON f.ProcessId = e.ProcessId
WHERE f.Channel = "Microsoft-Windows-Sysmon/Operational"
  AND f.EventID = 3
  AND f.DestinationPort = 389
```

- Rule ID: DTR-0017-6107-0843
  Reference TTP: TTP-0017-6049-9628
  Description: Detecta conexões de rede na porta 5985 (WinRM HTTP) realizadas por um processo PowerShell iniciado por 'control.exe', o qual foi executado a partir de 'sdclt.exe' com nível de integridade elevado (High). Essa atividade pode indicar execução remota de comandos, movimentação lateral ou exfiltração de dados via Windows Remote Management.
  Covered Techniques: T1021.006 (Remote Services: Windows Remote Management)
  Sources: Sysmon
  Query:
```
SELECT `@timestamp`,UtcTime,EventTime,Message
FROM apt29 f
INNER JOIN (
SELECT d.ProcessId, d.ParentProcessId
FROM apt29 d
INNER JOIN (
  SELECT a.ProcessGuid, a.ParentProcessGuid
  FROM apt29 a
  INNER JOIN (
    SELECT ProcessGuid
    FROM apt29
    WHERE Channel = "Microsoft-Windows-Sysmon/Operational"
        AND EventID = 1
```

```
        AND LOWER(Image) LIKE "%control.exe"
        AND LOWER(ParentImage) LIKE "%sdclt.exe"
  ) b
  ON a.ParentProcessGuid = b.ProcessGuid
  WHERE a.Channel = "Microsoft-Windows-Sysmon/Operational"
    AND a.EventID = 1
    AND a.IntegrityLevel = "High"
) c
ON d.ParentProcessGuid= c.ProcessGuid
WHERE d.Channel = "Microsoft-Windows-Sysmon/Operational"
  AND d.EventID = 1
  AND d.Image LIKE '%powershell.exe'
) e
ON f.ProcessId = e.ProcessId
WHERE f.Channel = "Microsoft-Windows-Sysmon/Operational"
  AND f.EventID = 3
  AND f.DestinationPort = 5985
```

- Rule ID: DTR-0017-6107-2091
  Reference TTP: TTP-0017-6049-9772
  Description: Detecta a execução do cmdlet 'Get-Process' em um contexto de sessão remota PowerShell, realizada pelo processo 'wsmprovhost.exe' (Windows Remote Management Provider Host). Esse processo é responsável por hospedar comandos executados remotamente via WinRM/PowerShell Remoting. A presença de 'Get-Process' indica possível atividade de reconhecimento de processos em um host remoto durante movimentação lateral.
  Covered Techniques: T1057 (Process Discovery); T1059.001 (Command and Scripting Interpreter: PowerShell)
  Sources: Sysmon; PowerShell
  Query:
```
SELECT `@timestamp`,EventTime,ScriptBlockText
FROM apt29 b
INNER JOIN (
    SELECT ProcessGuid, ProcessId
    FROM apt29
    WHERE Channel = "Microsoft-Windows-Sysmon/Operational"
    AND EventID = 1
    AND LOWER(Image) LIKE '%wsmprovhost.exe'
) a
ON b.ExecutionProcessID = a.ProcessId
WHERE b.Channel = "Microsoft-Windows-PowerShell/Operational"
  AND b.EventID = 4104
  AND LOWER(b.ScriptBlockText) LIKE "%get-process%"
```

- Rule ID: DTR-0017-6107-2726
  Reference TTP: TTP-0017-6050-0048
  Description: Detecta acesso ao compartilhamento IPC$ com referência a 'PSEXESVC', indicando possível tentativa de execução remota por meio do PsExec.

Covered Techniques: T1569.002 (System Services: Service Execution)

Sources: Windows Security Auditing

Query:
```
SELECT `@timestamp`,EventTime, Hostname, ShareName, RelativeTargetName,
    ↪ SubjectUserName
FROM apt29
WHERE LOWER(Channel) = "security"
  AND EventID = 5145
  AND ShareName LIKE '%IPC%'
  AND RelativeTargetName LIKE '%PSEXESVC%'
```

- Rule ID: DTR-0017-6107-3395

  Reference TTP: TTP-0017-6050-0265

  Description: Detecta operações de criação/abertura de arquivos realizadas por um processo filho de 'python.exe' cuja cadeia de execução remonta a 'services.exe'.

  Covered Techniques: T1105 (Ingress Tool Transfer)

  Sources: Sysmon

  Query:
```
SELECT `@timestamp`,UtcTime,EventTime,Message
FROM apt29 f
INNER JOIN (
    SELECT d.ProcessGuid
    FROM apt29 d
    INNER JOIN (
        SELECT b.ProcessGuid
        FROM apt29 b
        INNER JOIN (
          SELECT ProcessGuid
          FROM apt29
          WHERE Channel = "Microsoft-Windows-Sysmon/Operational"
              AND EventID = 1
              AND ParentImage LIKE '%services.exe'
        ) a
        ON b.ParentProcessGuid = a.ProcessGuid
        WHERE Channel = "Microsoft-Windows-Sysmon/Operational"
          AND Image LIKE '%python.exe'
    ) c
    ON d.ParentProcessGuid = c.ProcessGuid
    WHERE Channel = "Microsoft-Windows-Sysmon/Operational"
        AND EventID = 1
) e
ON f.ProcessGuid = e.ProcessGuid
WHERE Channel = "Microsoft-Windows-Sysmon/Operational"
    AND EventID = 11
```

- Rule ID: DTR-0017-6107-4117

  Reference TTP: TTP-0017-6053-3303

  Description: Detecta a execução de comandos PowerShell contendo "Get-ChildItem" por um processo PowerShell iniciado por 'python.exe', cuja

cadeia de execução remonta a 'services.exe', indicando possível atividade de enumeração de diretórios ou arquivos.

Covered Techniques: T1083 (File and Directory Discovery); T1059.001 (Command and Scripting Interpreter: PowerShell)

Sources: Sysmon; PowerShell

Query:
```
SELECT `@timestamp`,EventTime,ScriptBlockText
FROM apt29 h
INNER JOIN (
    SELECT f.ProcessId
    FROM apt29 f
    INNER JOIN (
      SELECT d.ProcessGuid
      FROM apt29 d
      INNER JOIN (
          SELECT b.ProcessGuid
          FROM apt29 b
          INNER JOIN (
            SELECT ProcessGuid
            FROM apt29
            WHERE Channel = "Microsoft-Windows-Sysmon/Operational"
                AND EventID = 1
                AND ParentImage LIKE '%services.exe'
          ) a
          ON b.ParentProcessGuid = a.ProcessGuid
          WHERE Channel = "Microsoft-Windows-Sysmon/Operational"
            AND Image LIKE '%python.exe'
      ) c
      ON d.ParentProcessGuid = c.ProcessGuid
      WHERE Channel = "Microsoft-Windows-Sysmon/Operational"
          AND EventID = 1
    ) e
    ON f.ParentProcessGuid = e.ProcessGuid
    WHERE Channel = "Microsoft-Windows-Sysmon/Operational"
      AND EventID = 1
      AND Image LIKE '%powershell.exe'
) g
ON h.ExecutionProcessID = g.ProcessId
WHERE h.Channel = "Microsoft-Windows-PowerShell/Operational"
    AND h.EventID = 4104
    AND LOWER(h.ScriptBlockText) LIKE "%childitem%"
```

- Rule ID: DTR-0017-6107-4769

  Reference TTP: TTP-0017-6053-5026

  Description: Detecta a execução de 'rar.exe', iniciado por um processo PowerShell que, por sua vez, foi iniciado por 'python.exe' cuja cadeia de execução remonta a 'services.exe'. Essa sequência pode indicar compactação de arquivos usando RAR.

  Covered Techniques: T1560.001 (Archive Collected Data: Archive via Utility)

  Sources: Sysmon

Query:
```
SELECT `@timestamp`,UtcTime,EventTime,Message
FROM apt29 h
INNER JOIN (
    SELECT f.ProcessGuid
    FROM apt29 f
    INNER JOIN (
      SELECT d.ProcessGuid
      FROM apt29 d
      INNER JOIN (
          SELECT b.ProcessGuid
          FROM apt29 b
          INNER JOIN (
            SELECT ProcessGuid
            FROM apt29
            WHERE Channel = "Microsoft-Windows-Sysmon/Operational"
                AND EventID = 1
                AND ParentImage LIKE '%services.exe'
          ) a
          ON b.ParentProcessGuid = a.ProcessGuid
          WHERE Channel = "Microsoft-Windows-Sysmon/Operational"
            AND Image LIKE '%python.exe'
      ) c
      ON d.ParentProcessGuid = c.ProcessGuid
      WHERE Channel = "Microsoft-Windows-Sysmon/Operational"
          AND EventID = 1
    ) e
    ON f.ParentProcessGuid = e.ProcessGuid
    WHERE Channel = "Microsoft-Windows-Sysmon/Operational"
      AND EventID = 1
      AND Image LIKE '%powershell.exe'
) g
ON h.ParentProcessGuid = g.ProcessGuid
WHERE Channel = "Microsoft-Windows-Sysmon/Operational"
    AND h.EventID = 1
    AND LOWER(h.CommandLine) LIKE "%rar.exe%"
```

- Rule ID: DTR-0017-6107-5368

  Reference TTP: TTP-0017-6053-5447

  Description: Detecta a exclusão de arquivos via 'cmd.exe', executado a p⌋ artir de `python.exe`, cuja origem remonta a `services.exe`.

  Covered Techniques: T1070.004 (Indicator Removal: File Deletion)

  Sources: Sysmon

  Query:
```
SELECT `@timestamp`,UtcTime,EventTime,Message
FROM apt29 j
INNER JOIN (
    SELECT h.ProcessGuid
    FROM apt29 h
    INNER JOIN (
        SELECT f.ProcessGuid
        FROM apt29 f
        INNER JOIN (
          SELECT d.ProcessGuid
          FROM apt29 d
```

```
            INNER JOIN (
                SELECT b.ProcessGuid
                FROM apt29 b
                INNER JOIN (
                  SELECT ProcessGuid
                  FROM apt29
                  WHERE Channel = "Microsoft-Windows-Sysmon/Operational"
                      AND EventID = 1
                      AND ParentImage LIKE '%services.exe'
                ) a
                ON b.ParentProcessGuid = a.ProcessGuid
                WHERE Channel = "Microsoft-Windows-Sysmon/Operational"
                  AND Image LIKE '%python.exe'
            ) c
            ON d.ParentProcessGuid = c.ProcessGuid
            WHERE Channel = "Microsoft-Windows-Sysmon/Operational"
                AND EventID = 1
        ) e
        ON f.ParentProcessGuid = e.ProcessGuid
        WHERE Channel = "Microsoft-Windows-Sysmon/Operational"
          AND EventID = 1
          AND Image LIKE '%cmd.exe'
    ) g
    ON h.ParentProcessGuid = g.ProcessGuid
    WHERE Channel = "Microsoft-Windows-Sysmon/Operational"
        AND h.EventID = 1
) i
ON j.ProcessGuid = i.ProcessGuid
WHERE j.Channel = "Microsoft-Windows-Sysmon/Operational"
    AND j.EventID = 23
```

- Rule ID: DTR-0017-6107-5740

  Reference TTP: TTP-0017-6053-5618

  Description: Detecta a criação do processo javamtsup.exe cujo processo p⌋ai é services.exe.

  Covered Techniques: T1569.002 (System Services: Service Execution)

  Sources: Sysmon

  Query:
```
SELECT `@timestamp`,UtcTime,EventTime,Message
FROM apt29
WHERE Channel = "Microsoft-Windows-Sysmon/Operational"
  AND EventID = 1
  AND ParentImage LIKE '%services.exe'
  AND Image LIKE '%javamtsup.exe'
```