TU WIEN Informatics

Advanced Persistent Threats: Attribution, Profiling, and Tracking

DISSERTATION

submitted in partial fulfillment of the requirements for the degree of

Doktorin der Technischen Wissenschaften

by

BSc. MSc. Aakanksha Saha
Registration Number 12123526

to the Faculty of Informatics

at the TU Wien

Advisor: Associate Prof. Martina Lindorfer

The dissertation has been reviewed by:

Simone Aonzo

Christian Wressnegger

Vienna, September 5, 2025

Aakanksha Saha

Erklärung zur Verfassung der Arbeit

BSc. MSc. Aakanksha Saha

Hiermit erkläre ich, dass ich diese Arbeit selbständig verfasst habe, dass ich die verwendeten Quellen und Hilfsmittel vollständig angegeben habe und dass ich die Stellen der Arbeit – einschließlich Tabellen, Karten und Abbildungen –, die anderen Werken oder dem Internet im Wortlaut oder dem Sinn nach entnommen sind, auf jeden Fall unter Angabe der Quelle als Entlehnung kenntlich gemacht habe.
Ich erkläre weiters, dass ich mich generativer KI-Tools lediglich als Hilfsmittel bedient habe und in der vorliegenden Arbeit mein gestalterischer Einfluss überwiegt. Im Anhang „Übersicht verwendeter Hilfsmittel“ habe ich alle generativen KI-Tools gelistet, die verwendet wurden, und angegeben, wo und wie sie verwendet wurden. Für Textpassagen, die ohne substantielle Änderungen übernommen wurden, haben ich jeweils die von mir formulierten Eingaben (Prompts) und die verwendete IT- Anwendung mit ihrem Produktnamen und Versionsnummer/Datum angegeben.

Wien, 5. September 2025

Aakanksha Saha

Kurzfassung

Advanced Persistent Threats (APTs) stellen eine der komplexesten und hartnäckigsten Herausforderungen im Bereich der Cybersicherheit dar und bergen erhebliche Risiken für Organisationen, Regierungen und die Gesellschaft insgesamt. In dieser Dissertation untersuchen wir die sich ständig entwickelnde Landschaft der APTs, indem wir ihre Verhaltensmuster und die Herausforderungen ihrer Zuschreibung analysieren. Erstens entwickeln wir durch Interviews mit Cybersicherheitsexpert_innen einen differenzierten dreistufigen Ansatz zur Analyse von APT-Kampagnen und heben die Bedeutung des Verständnisses der Taktiken, Techniken und Verfahren (TTPs) der Angreifer_innen gegenüber der direkten Gruppenzuschreibung hervor. Zudem beleuchten wir die operativen und kollaborativen Herausforderungen, denen Analyst_innen in realen Umgebungen gegenüberstehen. Zweitens stellen wir ADAPT vor, ein auf maschinellem Lernen basierendes Framework zur Automatisierung der APT-Zuschreibung. ADAPT wurde mit dem Schwerpunkt auf der Clustering heterogener Dateiartefakte – wie Dokumente und Binärdateien – entwickelt, die häufig in APT-Kampagnen verwendet werden. Drittens führen wir eine groß angelegte Messstudie zu dokumentbasierter Malware durch, einem häufig genutzten, aber wenig erforschten Vektor in APT-Kampagnen. Durch die Analyse von über 9.000 schädlichen Office-Dokumenten identifizieren wir gängige Angriffstaktiken und decken grundlegende Einschränkungen aktueller Dokumentenanalysetechniken auf. Diese Ergebnisse informieren sowohl das Design von ADAPT als auch die allgemeine Malware-Detektionsforschung. Abschließend untersucht diese Dissertation die komplementäre Dimension von Cyber Threat Intelligence (CTI) und ihre Rolle bei der APT-Zuschreibung. Wir führen eine umfassende Analyse von Verhaltensprofilen von Bedrohungsgruppen (TTPs und Tooling) durch, die von Plattformen wie MITRE ATT&CK und Malpedia abgeleitet wurden. Unsere Ergebnisse zeigen, dass den meisten Gruppen eindeutige Verhaltenssignaturen fehlen, was die Zuverlässigkeit der verhaltensbasierten Zuschreibung, die sich ausschließlich auf Bedrohungsinformationen stützt, in Frage stellt.

Insgesamt trägt diese Arbeit zum Verständnis von APT-Operationen bei, liefert nutzbare Werkzeuge für Anwender_innen und unterstreicht die Notwendigkeit von Zuschreibungsansätzen, die sowohl robust gegenüber unvollständiger Information als auch an realen Analysten-Workflows ausgerichtet sind.

Abstract

Advanced Persistent Threats (APTs) represent one of the most complex and persistent challenges in cybersecurity, posing significant risks to organizations, governments, and society at large. This dissertation investigates the evolving landscape of APTs by examining their behavioral patterns and the challenges associated with their attribution. First, through interviews with security practitioners, we identify a nuanced, three-layer approach to analyzing APT campaigns, emphasizing the importance of understanding attacker Tactics, Techniques, and Procedures (TTPs) over direct group attribution. We also highlight the operational and collaborative challenges analysts face in real-world environments. Second, we introduce ADAPT, a machine learning-based framework designed to automate APT attribution. ADAPT focuses on clustering heterogeneous file artifacts—documents and binaries—commonly used in APT campaigns. Third, given the limited research on document-based attack vectors, we conduct a large-scale measurement study of over 9,000 document malware samples from both targeted and widespread attacks. Our analysis identifies prevalent attacker tactics and exposes fundamental limitations in current document analysis techniques, informing both the design of ADAPT and broader malware detection research. Finally, we explore the complementary dimension of cyber threat intelligence and its role in APT attribution. We analyze threat group behavioral profiles (TTPs and tooling) derived from platforms such as MITRE ATT&CK and Malpedia, and find that most groups lack distinctive behavioral signatures, challenging the reliability of behavior-based attribution based solely on threat intelligence.

Collectively, this work advances understanding of APT operations, delivers actionable tools for practitioners, and highlights the need for attribution approaches that are resilient to incomplete intelligence and aligned with real-world analyst workflows.

List of Publications

This dissertation includes several research papers, which have been previously published in the following conference and workshop proceedings, and the chapters closely mirror these publications.

[228] **Aakanksha Saha**, James Mattei, Jorge Blasco, Lorenzo Cavallaro, Daniel Votipka, and Martina Lindorfer. Expert Insights into Advanced Persistent Threats: Analysis, Attribution, and Challenges. In Proceedings of the 34th USENIX Security Symposium (USENIX Security), 2025.
(Appears in Chapter 2)

[225] **Aakanksha Saha**, Jorge Blasco, Lorenzo Cavallaro, and Martina Lindorfer. ADAPT it! Automating APT Campaign and Group Attribution by Leveraging and Linking Heterogeneous Files. In Proceedings of the 27th International Symposium on Research in Attacks, Intrusions and Defenses (RAID), 2024.
(Appears in Chapter 3)

[226] **Aakanksha Saha**, Jorge Blasco, and Martina Lindorfer. Exploring the Malicious Document Threat Landscape: Towards a Systematic Approach to Detection and Analysis. In Proceedings of the 3rd Workshop on Rethinking Malware Analysis (WoRMA), 2024
(Appears in Chapter 4)

This dissertation also includes unpublished results from:

[227] **Aakanksha Saha**, Martina Lindorfer, and Juan Caballero. From IOCs to Group Profiles: On the Specificity of Threat Group Behaviors in CTI Knowledge Bases. *arXiv:2506.10645* (Under submission at AsiaCCS, 2026)
(Appears in Chapter 5)

Related publications that are not included in the dissertation.

[105] Gerhard Jungwirth, **Aakanksha Saha**, Michael Schröder, Tobias Fiebig, Martina Lindorfer, and Jürgen Cito. Connecting the .dotfiles: Checked-In Secret Exposure with Extra (Lateral Movement) Steps. In Proceedings of the 20th International Conference on Mining Software Repositories (MSR), 2023.

Contents

CHAPTER 1

Introduction

Over the past two decades, threats have evolved from isolated, opportunistic malware attacks into sophisticated, long-term espionage campaigns orchestrated by nation-states and well-resourced adversaries. These campaigns, commonly referred to as Advanced Persistent Threats (APTs), are aimed at infiltrating targeted organizations, maintaining access over extended periods, and pursuing strategic objectives such as data exfiltration, disruption, or surveillance. Unlike conventional malware threats that rely on rapid infection and commoditized tools, APTs are often tailored to the victim environment, using custom malware, novel techniques, and deception tactics to evade detection and attribution [171].

APT campaigns have increasingly become powerful tools for geopolitical influence, enabling adversaries to stealthily compromise government agencies, critical infrastructure, and private sector organizations. Notable examples such as Stuxnet [113], SolarWinds [143], and Hafnium [162] demonstrate the persistent, multi-phase nature of these advanced threats. These operations show how adversaries achieve long-term access through carefully sequenced stages, including initial compromise, privilege escalation, lateral movement, and data exfiltration. For instance, Stuxnet used multiple zero-day vulnerabilities and tailored payloads to silently disrupt industrial control systems, while the SolarWinds campaign involved the covert insertion of malware into legitimate software updates, allowing attackers to infiltrate thousands of organizations over several months. Similarly, Hafnium exploited Microsoft Exchange zero-days, deployed web shells, and maintained persistent access even after patching. These campaigns employed diverse tooling and persistent stealth to evade detection across extended periods.

Why attribution matters. Attribution is the process of identifying the threat actor or group behind a cyber threat operation. While technically complex and often uncertain, accurate attribution is essential for critical activities such as incident response, risk assessments, threat intelligence, and strategic policy decisions, including sanctions or diplomatic responses. Attribution helps organizations and governments connect specific threat campaigns to known threat actors, including cybercriminal groups, nation-states, or insiders. This knowledge guides decisions on how to respond, whether through technical countermeasures, public exposure, or diplomatic engagement.

Attribution is complex. In contrast to malware detection, which primarily determines whether software is benign or malicious, APT attribution aims to identify the adversary behind a malicious operation and to understand their objectives. Further, unlike binary authorship attribution, which focuses on identifying the individual who authored a specific malware sample, APT attribution is broader and more complex. It must account for coordinated, multi-phase campaigns that often involve multiple actors, infrastructure providers, and evolving toolchains.

For example, Google's Threat Analysis Group has documented multiple campaigns attributed to North Korean actors that used social engineering techniques to target security researchers [123, 265]. These operations spanned several years, employed novel exploits, and reused distinctive behavioral elements across otherwise separate campaigns. Attribution in such cases goes beyond linking binaries to a single author and requires correlating diverse artifacts to identify recurring patterns of behavior. In practice, this correlation is often labor-intensive and requires manual analysis by experts.

Expert insights into attribution. While significant progress has been made in developing automated solutions, such as machine learning-based malware clustering [79, 137, 217, 263] and provenance graph-based APT detection approaches [72, 80, 100, 125, 211, 223], these technical solutions still fall short of supporting the complex, real-world workflows of security practitioners. Expert insights are crucial for understanding how APT attribution is conducted in practice, bridging the gap between theoretical models and operational requirements.

To address this disconnect, we conducted semi-structured interviews with 15 professionals involved in APT investigations, including malware analysts, threat intelligence researchers, incident responders, and security consultants. Our findings reveal that attribution should not be viewed as a monolithic task, but rather as a layered process involving: (1) APT classification, which distinguishes targeted attacks from commodity threats; (2) TTP attribution, where analysts correlate incidents through malicious artifacts analysis and threat intelligence; and (3) country-level attribution, which, while often desirable, is deprioritized given geopolitical complexity, as it seeks to link activity to a specific nation-state.

Practitioners emphasized that attribution is valued not only for identifying adversaries but also for anticipating their future tactics, understanding their capabilities, and informing defensive strategies. However, they face challenges, as existing tools struggle to handle heterogeneous data and suffer from a lack of standardization in naming and labeling conventions. These limitations result in a continued reliance on manual effort to analyze and correlate APT artifacts, making timely and actionable remediation difficult to achieve.

Approach: ADAPT. The gap between analyst needs and existing tooling directly motivates the development of ADAPT (Attributing Diverse APT Samples), a system designed to automate campaign-level and group-level attribution. ADAPT addresses the operational challenges by aligning attribution tasks with real-world investigative workflows and reducing the manual burden practitioners currently face. ADAPT is a machine learning-based approach that uses clustering to group heterogeneous file types, such as documents and binaries, commonly used in APT campaigns.

To support this, we built a first-of-its-kind APT dataset consisting of 6,134 samples spanning heterogeneous file types. Through manual (re)-labeling, we identified 92 unique APT groups within the dataset. The majority of samples comprise executable binaries (58.73%) and document files (26.26%). Drawing on practitioner insights, we developed a methodology that clusters executable and document samples independently to facilitate TTP-level attribution. Beyond extracting features tailored to the dominant file types, we also compiled a set of linking features, commonly used in practice, to support heterogeneous sample correlation across campaigns and help attribute them to the same threat group origin. Together, ADAPT and its dataset advance reproducible research and aim to serve as an exploratory tool to help practitioners classify threat events into relevant campaigns or groups, supporting real-world threat investigations.

Document-based threats. As mentioned, documents ranked the second most prevalent file type in the APT dataset underlying ADAPT. However, feature extraction and clustering of document files proved challenging, as deriving robust and discriminative features for these formats is non-trivial. Moreover, limited research on the malicious document ecosystem, combined with the lack of a large and reliable dataset, motivated our subsequent study to collect and analyze 9,086 malicious documents used in both targeted and non-targeted attacks. Using a systematic analysis pipeline, we investigate threats in Microsoft Office (Word and Excel) and Rich Text Format (RTF) documents.

Our analysis highlights that basic file identification is surprisingly unreliable, as common tools such as libmagic, ExifTool, and Magika often disagree, requiring a majority-vote approach to stabilize downstream parsing and malicious content analysis. Malicious documents often embed weaponized macros, deliver obfuscated payloads, or exploit format-specific vulnerabilities. To increase their effectiveness, adversaries frequently pair these technical tactics with social engineering, such as deceptive prompts to "enable editing," that trick users into triggering malware execution. Along with the dataset and the identification of prevalent document-based threats, our work provides practical guidance for practitioners performing post-mortem analysis and automating the extraction of malicious content (IOCs) across different formats (e.g., OLE vs. OOXML). It further informs future research on techniques to address obfuscation strategies, including VBA and Excel 4.0 macros.

Threat intelligence-driven profiles. While ADAPT primarily focuses on malware artifacts, a complementary perspective for studying APTs and their attribution is cyber threat intelligence (CTI). Insights from our expert study with threat intelligence researchers highlight the importance of developing threat group profiles to enable proactive tracking and facilitate APT incident correlation. Motivated by this, we examine how attacker behaviors, including Tactics, Techniques, and Procedures (TTPs) and tooling, are represented in CTI knowledge bases and how they can support attribution tasks.

We assess the distinctiveness and completeness of behavioral indicators in open-source threat intelligence by evaluating two widely used CTI platforms, MITRE ATT&CK, and Malpedia. Our findings show that only a small fraction of threat groups have behaviors that are unique enough to support reliable attribution, with most groups lacking group-specific techniques, tools, or vulnerabilities. Even after combining both knowledge bases and augmenting profiles with additional behavioral indicators extracted from threat reports, 64% of groups still lack any distinctive behavior. These results highlight a key limitation of current CTI-driven analysis, where group profiles are often sparse, biased toward high-profile actors, and dependent on manually curated information.

Contributions and Structure of this Work. This thesis advances the state of practice by combining an automated approach to attribution, grounded in practical insights from analysts, with an assessment of the capabilities and limitations of current tooling and behavioral intelligence. In summary, we make the following contributions:

- First, we perform a qualitative study involving 15 practitioners across roles such as malware analysis, threat intelligence, and incident response. Our findings reveal a layered attribution process and expose critical misalignments between analyst needs and existing tools, motivating the development of more usable, workflow-aligned attribution systems.
- Second, we develop ADAPT, an unsupervised learning system that performs clustering of heterogeneous file types, including documents and binaries. ADAPT streamlines the attribution process by clustering samples at two levels. At the campaign level, it groups samples according to their specific tactics and techniques, enabling exploratory analysis and supporting threat investigations. At the group level, it incorporates infrastructure and operational traits to provide insights into the broader context in which campaigns operate and to associate samples with the corresponding threat group.
- Third, recognizing both the prevalence of document files among targeted threats and the challenges of extracting robust features from these formats, we carry out a large-scale measurement study of 9,086 malicious documents involved in both targeted and widespread attacks. The analysis exposes blind spots in current tooling caused by diverse document formats and obfuscation techniques, and provides actionable insights for improving document malware research.
- Finally, we evaluate the reliability of threat group profiles in open-source CTI platforms, focusing on MITRE ATT&CK and Malpedia. Our study quantifies the lack of distinctive behavioral indicators for most groups and identifies structural gaps that limit the effectiveness of CTI profiles for reliable attribution.

Together, these contributions offer both methodological and practical advances in APT attribution. The remainder of the thesis is organized as follows: Chapter 2 presents the findings of our practitioner study on APT attribution, highlighting analyst workflows, operational challenges, and the importance of understanding attacker Tactics, Techniques, and Procedures (TTPs). This chapter is mainly based on the paper published at USENIX Security [228]. Chapter 3 introduces ADAPT, an unsupervised learning-based framework for clustering heterogeneous file types, including documents and binaries. We detail feature extraction, clustering methodology, and evaluation, demonstrating how insights from the practitioner study inform automated attribution. This chapter is primarily based on the work published at RAID [225]. Chapter 4 presents a large-scale measurement study of document-based malware, analyzing over 9,000 samples to identify prevalent attacker tactics and limitations in existing detection approaches. These results were presented at the European Symposium on Security and Privacy Workshop (WORMA/EuroS&PW) [226]. Chapter 5 evaluates CTI knowledge bases and attacker behavioral distinctiveness. The corresponding technical report is currently under submission at AsiaCCS [227]. Finally, Chapter 6 summarizes the results and outlines future research directions.

CHAPTER 2

Expert Insights into Advanced Persistent Threats: Analysis, Attribution, and Challenges

Advanced Persistent Threats (APTs) are sophisticated and targeted threats that demand significant effort from analysts for detection and attribution. Researchers have developed various techniques to support these efforts. However, security practitioners' perceptions and challenges in analyzing APT-level threats are not yet well understood. To address this gap, we conducted semi-structured interviews with 15 security practitioners across diverse roles and expertise. From the interview responses, we identify a three-layer approach to APT attribution, each having its own goals and challenges. We find that practitioners typically prioritize understanding the adversary's tactics, techniques, procedures (TTPs), and motivations over identifying the specific entity behind an attack. We also find challenges in existing tools and processes mostly stemming from their inability to handle diverse and complex data and issues with both internal and external collaboration. Based on these findings, we provide four recommendations for improving attribution approaches and discuss how these improvements can address the identified challenges.

2.1 Introduction

Advanced Persistent Threats (APTs) have become a critical instrument of modern geopolitical warfare, allowing nation-states to conduct sophisticated cyber espionage and strategic intelligence. Cyber threat analysts regularly uncover APT campaigns targeting government agencies and private sector companies [32, 144, 257]. Attribution of these campaigns has exposed evolving and sophisticated adversaries that engage in espionage, theft of information, and disruption of services. In response, researchers and industry practitioners have advanced APT detection [72, 79, 80, 100, 125, 137] and attribution [211, 217, 223, 225, 263] emphasizing its critical role in informing defensive strategies and understanding adversarial behaviors. Despite these advancements, the majority of existing research is concentrated on developing technical solutions for robust and accurate attribution. This includes automating malware clustering and leveraging machine learning for detection and attribution. However, technical solutions often do not fully engage with the professionals who actively use, manage, or attribute malware in real-world scenarios.

Understanding real-world practices is crucial for bridging the gap between theoretical models and practical applications. By examining how APT investigations are conducted in the field, we can ensure that tools and techniques are designed to meet the actual needs of analysts, align with their processes, and identify key assumptions that might simplify tool development. While previous studies have focused on reverse engineering [261] and malware analysis [273], and recent research has explored broader threat-hunting practices [17, 154], our research uniquely identifies the nuanced challenges specific to APT incidents and attribution, examining how practitioners navigate these complex scenarios.

In the area of generic, i.e., non-targeted, malware, Wong et al. [267] recently identified a significant misalignment between the practical challenges faced by malware experts and the focus of existing research solutions. Building on their observations, we explore the complexities of APTs and their attribution to better understand the disconnect between research and practical applications. Our study provides insights into the relevance and effectiveness of APT attack attribution tools and methodologies, aiming to offer a deeper understanding of *'why attribution is important'* and *'how attribution is performed'* in real-world scenarios. With this in mind, we seek to answer the following three main research questions:

RQ1 Why is attribution important, and what objectives does it serve in the context of security incidents?

RQ2 What are the key steps and processes involved in investigating and attributing APT incidents?

RQ3 What challenges and obstacles do practitioners face when investigating and attributing APT activities?

To address these questions, we conducted semi-structured interviews with a diverse group of 15 security practitioners including malware analysts, threat intelligence researchers, security consultants, and incident responders. Each of these roles plays a crucial part in the attribution process at different levels, highlighting the collaborative nature of effective attack investigation and attribution. Incident responders often initiate investigations and provide critical insights into active threats, while malware analysts and threat intelligence researchers offer detailed analyses of attack artifacts. Security Operation Center (SOC) and incident response team leads coordinate investigations, while upper management integrates findings into broader organizational strategies.

In our interviews, we explored how security practitioners investigate and attribute APTs, focusing on the tools and techniques they use for analyzing malicious samples, threat hunting, and addressing challenges in the attribution process. We also explored the use of internal and external intelligence for tracking and correlating threats, as well as broader organizational and policy-related aspects, such as collaboration between teams or agencies. The interviews provided a comprehensive overview of the strategies employed by security professionals across various roles.

We found that attribution can generally be modeled across three distinct decision levels. (1) *APT Classification*, (2) *Tactics, Techniques, and Procedures (TTP) Attribution*, and (3) *Country Attribution*. Victim organizations progress through these levels depending on the nature of the incident and their specific needs. APT classification helps differentiate between generic threats and advanced threats, thus prioritizing resources for mitigation efforts. TTP attribution involves a comprehensive and collaborative investigation to identify the specific TTPs used by the threat actor, aiding precise and effective response strategies. In contrast, country attribution—aiming to identify the exact entity (nation or country) behind the attack—is often challenging and less emphasized. Participants prioritize identifying *what* threat actors are likely to do rather than focusing on *who* they are, as the latter is often not critical for immediate response efforts.

In addition to the identified goals and decision process for APT attribution, we observed several challenges in practice. These primarily stem from existing tools' inability to handle the diverse and complex data required for accurate attribution and difficulties in attributing APTs that use standard system tools, shared infrastructure, and overlapping malware. Further lack of standardization in naming conventions affects the merging and correlation of threat information from disparate sources. We also noted issues with internal and external collaboration, which is essential for the more advanced levels of attribution.

In summary, we make the following contributions:

- We offer a comprehensive understanding of security practitioners' processes for investigating APTs, identifying attribution as a layered process that balances the accurate identification of threat actors with the practical considerations of incident mitigation.
- We highlight the various tools and processes used to investigate the three distinct levels, i.e., (1) APT Classification, (2) TTP Attribution and (3) Country Attribution.
- We provide insights into the challenges encountered with these tools and processes, offering recommendations for improving attribution-based research.
- Based on our results, we provide four recommendations for improving attribution tool development and threat intelligence sharing.

Artifacts. To support transparency, replication, future research, and compliance with the open science policy, we include relevant research materials as part of our artifact, namely (1) interview questions, (2) survey questions, and (3) a codebook, at https://osf.io/hjdk2/.

To comply with data protection requirements and maintain ethical research practices, we do not include raw interview data, such as audio recordings or transcripts, in our replication package. This decision reflects our strong commitment to safeguarding participant privacy and ensuring their right to data protection. By excluding this data, we mitigate the risk of inadvertently disclosing information that could potentially identify our participants or their roles. Instead, we present our findings using thematic analysis and anonymized interview quotes, ensuring our research insights are shared without compromising participant confidentiality.

2.2 Background & Related Work

APT incident response involves a focused set of activities within security operations, including the detection of sophisticated malicious activities, in-depth analysis of attack artifacts, and the attribution of these activities to specific threat groups.

APT Detection and Attribution. Existing APT detection research primarily uses alert correlation to identify anomalous behaviors or APT footprints. For instance, Ghafir et al. [72] developed MLAPT, a machine-learning based system for detecting APTs via network traffic data, while Sachinananda et al. [223] correlated alerts from Intrusion Detection System (IDS), Endpoint Detection and Response (EDR), and Security Information and Event Management (SIEM) systems to cluster those related to the same APT attack scenario. Provenance graphs have also emerged as a state-of-the-art approach in APT detection with tools like ANUBIS [10], APTHunter [137], Unicorn [80], NODLINK [125], and MAGIC [100] using audit logs for anomaly detection.

Prior work on APT attribution has utilized various methods to link malware to specific threat groups. Marquis-Boire et al. [152] manually extracted static features like command and control (C&C) infrastructure to associate executables with their authors. Rosenberg et al. [217] and Wang et al. [263] applied machine learning to classify APT groups using features from sandbox reports and string and code features, respectively, while Han et al. [79] used dynamic API sequences for detection and attribution. Mirzaei et al. [169] identified unknown APT samples through code reuse analysis. Ren et al. [211] proposed a knowledge graph model for attribution using Open-Source Cyber Threat Intelligence (OSCTI). Most recently, ADAPT [225] utilized static features extracted from heterogeneous file types for attribution by clustering executables and documents in threat groups and campaigns.

Expert Studies in Security Operations. In addition to developing technical solutions, human-centered studies have been conducted to understand the cognitive process of software reverse engineering [12, 150, 261, 273]. Votipka et al. [261] performed a study with reverse engineers in 2020. They developed workflows that represent the necessary process reverse engineers follow and suggest guidelines for designing future reverse engineering tools. This early research has informed subsequent studies that further explore this area [150], and investigate other related fields such as malware analysis. In 2021, Yong et al. [273] conducted a user study specifically to understand the objectives and workflows of malware analysts in practice. Aonzo et al. [12] compared the procedures followed by humans and machines to classify unknown programs as benign or malicious, aiming to understand how data from malware analysis reports is used to reach a decision. They accomplish this by designing an online game that requests participants to classify suspicious files based on their sandbox reports. Prior work has also investigated the usability of tools used by reverse engineers [153, 269] and compared the vulnerability discovering process of software testers and ethical hackers [262]. More recently, Maxam et al. [154] and Badva et al. [17] have focused on the broader practice of threat hunting and the associated challenges, with their studies examining the overall process of detecting and responding to threats.

There is a rich line of research on SOC workflows, exploring the general challenges analysts face. Studies have interviewed SOC analysts to investigate their views on security misconfigurations [53], strategies for analyzing sophisticated malware attacks [4], burnout among SOC personnel [245], collaboration between people and tools [73], and the problem of excessive and false security alerts [5, 109]. Oesch et al. [189] examined the usability of two machine-learning based network security tools, identifying issues such as poor documentation and inconsistent UI design, based on surveys of six US Naval SOC analysts. Mink et al. [168] expanded on this by exploring the unique challenges of machine-learning based tools in SOCs, particularly focusing on their explainability and how they differ from traditional tools.

There is also a substantial body of work examining the properties of open threat intelligence (OTI), also known as abuse feeds and blocklists [112, 157]. These studies consistently highlight issues with coverage, timeliness, and accuracy. Recent efforts to measure threat intelligence (TI) quality have primarily focused on OTI, as seen in studies by Li et al. [126] and Griffioen et al. [77]. David Bianco [22] found that contextualized, high-level TI can help address false positives; however, a 2019 SANS survey [31] revealed that respondents still value low-level indicators of compromise more than high-level TTPs. Bouwman et al. [28] explored paid threat intelligence (PTI) and found that experts favor PTI due to its curated and aggregated information. Tounsi et al. [253] demonstrated that (1) fast sharing of TI alone is insufficient to prevent targeted attacks, (2) trust is crucial for effective TI sharing between organizations, (3) standardized TI formats enhance data quality and support better-automated analytics, and (4) the best TI tool depends on an organization's goals, balancing standardization, automation, and speed requirements.

While previous studies have primarily focused on general malware analysis and reverse engineering workflows, user experiences with various security tools, and broader SOC and threat-hunting workflows, our research is specifically concentrated on analyzing the specific workflows, processes, and challenges involved with APT incidents.

2.3 Methodology

We employ a semi-structured interview protocol designed to get detailed insights into experts' processes. We conducted the interviews with key stakeholders within the SOC [5], including threat intelligence researchers, incident response specialists, security consultants, and malware analysts across various levels of seniority. This approach ensured a comprehensive understanding from multiple perspectives. The interviews were designed to gain deeper insights into how security practitioners investigate and attribute APTs, with a focus on the tools and techniques they employ. Our research is specifically centered on examining the workflows, processes, and challenges associated with managing real-life APT incidents.

Our study consists of two parts (see Figure 2.1): a screening survey to select qualified participants, and a one-hour semi-structured interview, during which we recorded video and audio, which we later transcribed. Our study was reviewed and approved by our institutions' ethics review boards (details provided in our ethics and open science statement).

In the following, we describe our recruitment, screening survey, interview, and data analysis procedures, as well as our survey's limitations.

Recruitment. We recruited participants over a six-month period (November 2023 – April 2024) through multiple social media platforms (e.g., Twitter/X, LinkedIn), and by distributing flyers at targeted in-person industry security conferences. We also reached out to our personal contacts in various organizations who then shared the study information with their security teams. We recruited participants from security companies, managed security service providers, and the security research domain, all of whom have extensive experience in SOC roles and handling APT security incidents, which was verified through the screening survey. In total, 15 qualified participants completed the interview. Our sample size is sufficient to provide strong guidance for future quantitative work and develop generalizable recommendations for design based on qualitative best practice [78].

We stopped recruitment when we observed that no new concepts or themes appeared from the interviews (i.e., thematic saturation [45]).

Eligibility. We invited participants, who were older than 18 with at least one year of professional experience in dealing with APT incidents and attribution. We assess the experience criterion through the screening survey, where participants self-reported their relevant industry experience and the primary goals of their threat or malware analysis work.

Screening Survey (Figure 2.1.A-C). Participants began by completing a brief screening survey in which they self-report their job titles, roles, and industry sectors. They further report their expertise in specific areas—such as malware analysis, APT tracking, threat intelligence, and incident response—using Likert scales with options ranging between "Novice," "Intermediate," "Advanced" and "Expert" to capture self-reported proficiency (see Appendix 6 and our artifact).

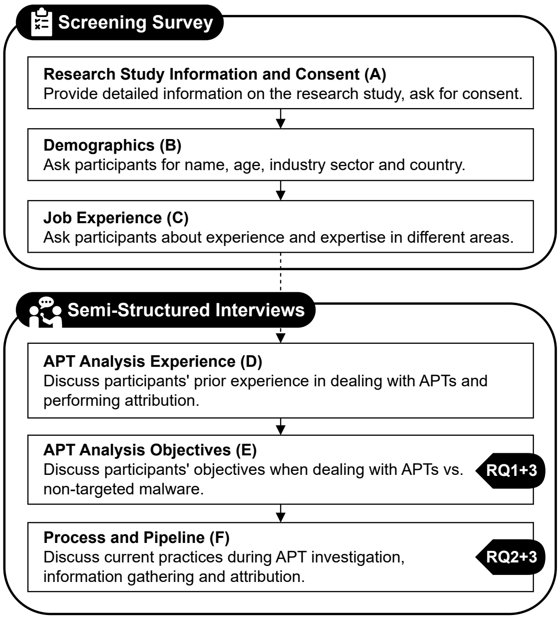
Screening Survey
Research Study Information and Consent (A)
Provide detailed information on the research study, ask for consent.
Demographics (B)
Ask participants for name, age, industry sector and country.
Job Experience (C)
Ask participants about experience and expertise in different areas.
Semi-Structured Interviews
APT Analysis Experience (D)
Discuss participants' prior experience in dealing with APTs and performing attribution.
APT Analysis Objectives (E)
Discuss participants' objectives when dealing with APTs vs. non-targeted malware.
RQ1+3
Process and Pipeline (F)
Discuss current practices during APT investigation, information gathering and attribution.
RQ2+3

Figure 2.1: Study protocol diagram outlining key stages: (A) Research Study Information and Consent, (B) Demographics, (C) Job Experience, (D) APT Analysis Experience, (E) APT Analysis Objectives, and (F) Processes and Pipeline.

Semi-Structured Interview (Figure 2.1.D-F). We invited eligible participants to a 60-minute online interview, conducted in English. The interview procedure was designed to provide a comprehensive understanding of participants' approaches to handling APTs. Initially, we explored participants' understanding of APTs and the significance of attribution. We focused on their knowledge and experience with APT incidents, including the processes and pipelines used for investigation. To capture a broad range of perspectives, we did not impose a strict definition of attribution; instead, we utilized participant-provided definitions throughout the interviews. This approach facilitated discussions on the varying levels of attribution, including steps taken to identify the type of attack, the attackers, and their tactics (modus operandi). Subsequently, we examined current practices by inquiring about the specific tools and processes employed during APT investigations. This included methods for threat correlation, usage of machine learning, and the integration of Cyber Threat Intelligence (CTI) into investigations and attribution. Finally, we discussed the challenges participants face when managing APT incidents and performing attribution, aiming to identify common obstacles and areas for potential improvement in current practices. Our interview questions are listed in Appendix 6 and available as part of our artifact.

To maintain consistency between interviews, the interviewer followed a detailed guide on best practices, including how to begin and end the interview, ask questions in a non-leading manner, re-obtain consent, and allow time for participant questions (adapted from Rader et al. [200]). To ensure the clarity of questions and the use of appropriate terminology, we co-designed the interview questions with a usability security expert with nearly a decade of experience and a security threat analyst from the authors' personal contacts. To ensure the questions were easily understandable, we conducted three pilot interviews. The pilot participants were selected to reflect the professional experience and expertise of our target population. The first participant is a security analyst with over 9 years of experience in nation-state threats and AI/ML in security tools. The second participant is a geopolitical intelligence analyst with 12 years of experience in ransomware, cryptocurrency, and the dark web. The third participant is an Associate Professor and founder of a startup, with over 10 years in advanced malware research and cybersecurity education. Following the pilot interviews, we made minor adjustments to the questionnaire and the major themes. Hence, we do not include the pilot interviews in the final data. Our changes included consolidating questions about machine learning usage into the broader theme of tools and processes and added questions about gathering intelligence. Additionally, we incorporated prompting examples drawn from our pilots' responses to help participants better understand the context of the questions if the participant appeared confused.

Data Analysis. We transcribed all interviews using the GDPR-compliant transcription service MAXQDA [155]. We then analyzed these transcripts following an inductive thematic coding approach [45]. To establish an initial codebook, two authors collaboratively analyzed two interviews, allowing codes to emerge from the data and then discussing the initial codebook with the full research team. The two authors then independently coded interviews in rounds of two. After each round, inter-rater reliability (IRR) was calculated using Krippendorff's Alpha (α) to account for chance agreement during coding [85]. Then, coders met to resolve disagreements, change the codebook as necessary, and apply changes to previously coded interviews. The full research team met to discuss the results after each round and to review proposed changes to the codebook. These changes included identifying and merging overlapping codes, as well as adding new codes to reflect emerging themes, such as future improvements that our interviewees suggested as the field of APT attribution progresses. After four rounds of independent coding (eight interviews), an IRR of $\alpha > 0.8$ was reached for all subjective codes, indicating high agreement [85, 134]. We did not calculate alpha for objective variables like tools mentioned by participants, as these can be inferred directly from the transcript. The five remaining interviews were coded independently by one author. The final codebook and α values for each variable are available as part of our artifact.

In the next phase, we performed axial coding to explore the relationships between and within these categories [45]. We aimed to develop a theoretical model by extracting and organizing themes from the coded data. We identified three primary categories related to the handling of APT incidents i.e., attribution value, attack analysis (including processes and tools), and the challenges encountered within each process. We further linked these challenges to participants' future recommendations and suggestions. From these connections and relationships, we derived a theory that identifies the high-level processes and specific technical approaches used by analysts.

Limitations. As a semi-structured interview, some follow-up questions may not have been asked in every session, and participants' responses might not cover all topics with the same depth. This is especially common for expert tasks [11]. We produced a thorough script, and a single interviewer conducted each interview to improve consistency. None of the participants spoke off the record. It is important to acknowledge that due to the limited amount of time per interview, certain themes might not be covered in participants' responses. This further adds to the motivation not to generalize the findings based on the frequency of specific responses.

Second, there may exist concerns that the participants do not fully represent the entire population of security professionals who deal with APT-related incidents, such as government officials. Although we recruited participants from diverse professional and demographic backgrounds (see Table 2.1), our sample is predominantly centered on US and EU participants. Moreover, it does not comprehensively represent all possible roles, industries, or demographics. To address this limitation, we ensure careful interpretation of our qualitative results and do not attempt to generalize our findings. Instead, we focus on capturing a diverse range of perspectives from various stakeholders within the community. These findings can serve as a foundation for hypotheses in large-scale surveys or targeted studies focused on specific professional or demographic factors in future research. Finally, biases such as social desirability and confirmation bias may influence some participants' responses. We mitigated these by framing questions in a neutral manner and encouraging participants to consider and discuss opposing viewpoints.

Ethics considerations. This study was performed in collaboration between institutions in Europe and the US and was approved by TU Wien's Research Ethics Committee (REC) in Europe and the Institutional Review Board (IRB) at Tufts University in the US. Informed consent was obtained from all participants during the screening survey, and they were provided with detailed information about the research objectives and interview protocol. While we collected email addresses for interview scheduling purposes, this was the only personally identifiable information (PII) gathered. These email addresses were deleted once no longer needed, and they were not stored with the interview data. During transcription, all names of individuals, countries, and organizations mentioned were anonymized using unique identifiers to protect participant identities. The consent form clearly stated the use of the transcription service, ensuring compliance with GDPR rules involving third-party access to recorded data [155].

Participants were given the option to conduct interviews without video if it made them more comfortable, and they could choose not to show their faces during video calls. Additionally, if participants shared their screens to demonstrate workflows, we ensured this information was kept secure and not shared further. The data analysis was conducted on the first author's institution premises, and only aggregated results and anonymized transcripts were shared among the research team. To prevent the potential misuse of research data and address risks associated with disclosing information about threat actors, we encouraged participants to withhold or omit sensitive details they were uncomfortable sharing, particularly if they felt such information could be exploited by malicious parties. Participants were also allowed to speak "off-the-record" by pausing the recording at any

Table 2.1: Participants in our study along with their roles, organizations, and years of experience. *Role key:* MA = Malware Analyst, IR = Incident Response, TI = Threat Intelligence. *Organization (Org) key:* MSS = Managed Security Services, SC = Security Company, SR = Security Research, IntSec = Internal Security Team (Tech, Financial, Healthcare). *Size key:* S = Small, M = Medium, L = Large.

ID	Job Title (Experience in Years)	Sector	Org Type	Org Size
P1IR/MA	Research Director (22)	Industry	MSS	M
P2TI	Managed Defense Head (13)	Industry	MSS	M
P3IR/TI	Security Consultant (6)	Industry	SC	S
P4IR	Research Scientist (12)	Non Profit	SR	S
P5MA	Senior Malware Analyst (18)	Industry	IntSec	L
P6TI	Threat Intelligence Researcher (5)	Industry	SC	L
P7IR	Security Analyst (10)	Government	SR	M
P8IR/MA	Security Researcher (10)	Industry	SC	M
P9IR	Security Operation Lead (14)	Industry	MSS	M
P10TI	Manager CTI (15)	Industry	MSS	L
P11IR	Security Engineer (17)	Industry	IntSec	L
P12IR	Sec. Operations Director (15)	Industry	SC	L
P13IR	Senior Threat Hunt Analyst (9)	Industry	SC	L
P14IR/TI	Threat Operations Lead (16)	Government	IntSec	L
P15MA/TI	Sec. Consult. Manager (10)	Industry	MSS	L

time. Participants were given the opportunity to review any quotes attributed to them before publication and the context for specific quotes was provided by describing the respondent's role and sector.

2.4 Participants

We had 15 participants who completed the interview. All participants had more than five years of experience, with most having over ten years, specifically in SOC operations. By interviewing participants with extensive experience working at well-established security groups in large tech companies, leading security industry organizations, and government agencies, we were able to identify a wide range of perspectives and gain a comprehensive understanding of how APT incidents are managed in practice.

Table 2.1 shows the list of all participants, their job title, sector, type of organization, size of organization, and years of experience. Professionally, our participants consisted of a variety of roles, including first-level responders to active security alerts, upper management in their organizations, senior malware analysts, research scientists, and threat intelligence researchers. Five participants worked for managed security services, five for security companies, two in security research and advocacy, and three on tech, financial, or healthcare institutions' security teams. Eight participants were from large organizations with more than 5,000 employees (several exceeding 100,000), five from medium-scale organizations with 50-5,000 employees, and two from small organizations with fewer than 50 employees. We provide this information about participants' roles and organizations only to add context and demonstrate the sample's diversity.

Our participants reside in a variety of countries, such as the UK, USA, Austria, Canada, Israel, and Finland. On average, participants spent about 70 minutes completing both the survey and the interview (60 minutes of which were the interview). The majority of study participants identified as men; two identified as women. Our participants were educated (i.e., all had a Bachelor's degree and eight had a graduate degree). Additionally, our participants reported having an advanced level of skill in at least one area relevant to attribution, i.e., malware analysis, APT attribution, threat intelligence research, or incident response.

2.5 Result: Goals of Attribution (RQ1)

In this section, we discuss the importance of attribution and explore scenarios in which participants expressed interest and reasons for attributing an incident. Note, through our interviews, we do not attempt to generalize the prevalence of specific practices across all APT investigations as certain practices may not be applicable in all contexts or roles. Instead, we enumerate the range of practices and tools present generally in APT analysis to support future quantitative investigation. To enrich our findings, we include incident and organizational detail whenever participants provided them. However, it is important to note that participants shared experiences from incidents they had handled throughout their careers, sometimes referencing past organizations they worked for, without offering detailed descriptions of specific incidents or organizations. Additionally, we categorized roles based on participants' job titles and responsibilities. We observed from our interviewed samples that APT attribution often involves multiple roles, such as malware analysts, incident responders, and threat intelligence analysts, sometimes filled by the same person. However, we found that the tasks associated with these roles were not always consistent across different organizations. There are similar indications in prior research [26] regarding the correlation between job titles and the tasks performed, suggesting that while job titles may guide expectations, the actual tasks can vary based on other contextual factors. In our study, we observed that in-depth reverse engineering of malware was exclusively handled by senior malware analysts. Apart from this, we did not observe clear differences in themes across the reported roles, suggesting that our findings apply broadly across different aspects of the work.

TTP attribution informs investigation and effective threat prioritization. Several participants (N=8) pointed out that some level of threat actor attribution is useful for understanding the threat and guiding incident response. P3IR/TI noted (TTP) attribution helps in, "understanding what TTPs to look for in their network." P11IR further explained being able to "attribute a binary" to a "specific threat actor" helps in pulling other "indicators or TTPs" that are known for that specific threat actor and use that as a way to perform a deep dive investigation, further adding that TTP-level attribution "provides a lot of pivot points to be able to search for other things" in the environment and being comprehensive in the investigation. Participants emphasized that even partial attribution, such as classifying a threat as generic versus an APT, streamlines the incident response protocol (N=3). This classification allows organizations to effectively deploy specialized forensics teams, trigger adherence to specific protocols or engagement with law enforcement, and expedite remediation when necessary. Understanding whether an attack is specifically targeted or a “spray-and-pray” approach helps organizations to prioritize resources on addressing high-risk, targeted threats while de-prioritizing more generic, less impactful attacks. P6TI explained “If we don’t know what the threat is, we don’t know how severe or how to prioritize the attack. So [attribution] is quite a good way to know whether you’re being targeted or whether it’s sort of spray and pray.” P10TI further highlights the importance of accurate attribution in ransomware attacks. They added understanding the adversary and “what assets you have that are so attractive” allows for well-informed remediation when you “need to negotiate” with the attacker.

Balancing incident mitigation with full attribution for strategic decision-making. Participants emphasized (N=7) that victim organizations’ primary initial concerns are not identifying the specific perpetrator—what we will refer to as country attribution going forward—, but rather understanding an incident’s full scope. The question of who carried out the attack is often secondary, unless the target is politically sensitive, where country attribution carries more significance. For example, P1IR/MA noted “This was China, or this was Russia or this hacking group; that aspect comes right at the end, if at all. We’re not really too bothered if we get to that point or not because. . . [the client] just want a warm, fuzzy feeling that [the attackers] are out of the network.” However, participants (N=4) mentioned cases where full country attribution becomes valuable particularly when understanding the origin of an attack can significantly impact response strategies. For example, P6TI described their thinking assuming a scenario where they were operating a Ukrainian network and detected lateral movement saying “If it’s Russia, they’re going to go straight for the domain controller and then wipe everything. If it’s China, they might try and persist in there a little bit longer. . . you would want to try and stop the Russians first because they’re going to destroy the whole network.”

In such scenarios, prioritizing responses based on the likely actions of the threat actor, such as stopping Russian actors who may destroy the network versus Chinese actors who might engage in prolonged data exfiltration, can be critical for effective incident management. While participants highlighted the importance of rapid and accurate full-country attribution to deploy strategic and timely countermeasures, our study does not delve into the specific details of the mitigation process, as this was not the central focus of our discussions. Instead, we analyzed the goals and technical aspects of attribution and how it informs incident response.

Further, country attribution is necessary in cases when an incident might have geopolitical, legal, or strategic business consequences. As P4IR explained, precise country attribution can be used “to make policy, to respond, you know, and different entities can respond in different ways, right? Like a company might want to understand attribution, like, you know, Google, like, oh, our our servers are being hacked by a Chinese group. We're going to withdraw Google from China, right? Like they did in 2009, 2010." When participants described this level of attribution, they were quick to point out that a high level of caution and confidence is necessary as misattribution can cause geopolitical tensions or legal challenges and risks escalating conflicts. P3IR/TI remarked "You have to be extremely cautious when you're saying something like that. Right. And that level of attribution, because there are greater implications."

Country attribution helps with long-term remediation and proactive measures. While they might not care about full country attribution during the incident response, a few participants (N=3) recognized its importance for informed, long-term organizational strategy. P6TI explained "If you know a specific country, countries, APT groups are coming after you, then you know you should focus your research on the capabilities of those APT groups. . . you know, trying to be kind of predictive in a way."

2.6 Result: Processes and Tools (RQ2)

In accordance with the responses described in Section 2.5, we observed a decision tree describing how participants moved through increasing levels of attribution specificity. We begin this section by summarizing the decision tree and providing a visual depiction in Figure 2.2. Progression through the decision tree depends on the characteristics of the identified threat and the participants' organizations' motivation. As participants described moving through the decision tree, additional features were considered, and analysis processes were conducted to provide more detail. We describe the specific features and processes for each level in turn in this section.

Attribution Decision Tree. When handling a potential APT-level incident, the first step is to determine whether it qualifies as an APT. Participants reported certain characteristics that help distinguish an APT-level attack, as discussed in Section 2.6.1. If the incident does not meet these criteria, standard triage procedures are followed to contain and mitigate the threat. However, if the incident is confirmed as an APT and the organization has processes for TTP attribution, it triggers an advanced response involving different teams and an in-depth forensic process, as discussed in Section 2.6.2. The investigation then focuses on the dedicated remediation process by identifying various attributes and TTPs associated with the incident mentioned in Section 2.6.3. If the organization aims to identify and attribute the incident to a specific country, it further gathers and connects additional intelligence to pinpoint the entity behind the incident discussed in Section 2.6.4.

2.6.1 APT Classification

Our participants reported that the first critical step toward potential full attribution is determining whether it is, in fact, an APT (see Figure 2.2.A). To this end, participants considered several indicators of potential APT activity. We discuss each below. Note

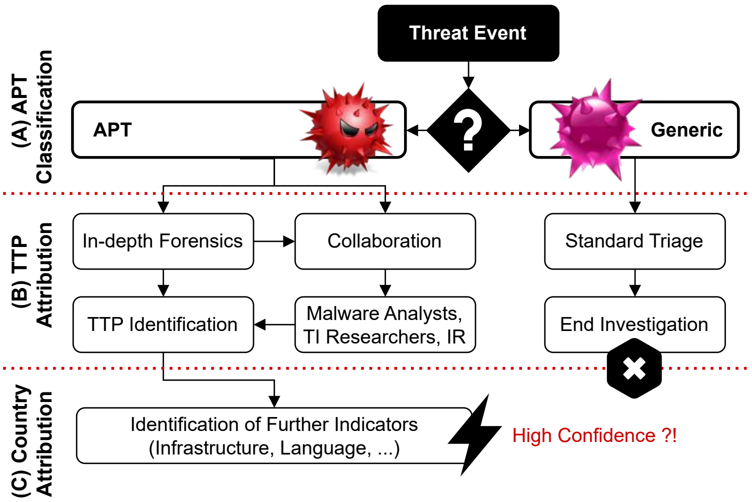

Figure 2.2: Decision tree outlining the process for attributing threats, starting from *APT Classification*, followed by in-depth forensic analysis and *TTP Attribution*. The decision process ends with a high level of confidence at *Country Attribution*.

that this layer of partial attribution acts as a classifier for initial triage, so any of these indicators can be sufficient to signal an APT.

APT attacks are characterized by low-profile tactics and targeted efforts. Participants (N=7) emphasized that APT actors rely on stealthy techniques, such as 'living-off-the-land' methods. By using existing tools and legitimate system processes, these actors blend seamlessly with regular activity, making detection more difficult. P9IR noted, "The attack is not noisy at all. It's very targeted. . . that is going to be a problem for any automation that you have, finding anomalies like that, because they are not an anomaly." P6TI suggested this targeted nature makes it necessary to compare information from other organizations to assess whether an attack was specifically targeted. They explained, "Our email gateway that we use, we can actually see all the other customers that also receive the same email as you. So that's quite a good way to know whether you're being targeted or whether it's sort of spray and pray. Because if a thousand other customers also receive the same email that you did, then it's not really going to be targeted." This method, they added, helps prioritize threats that require more in-depth threat hunting and due diligence.

APTs are characterized by their use of lateral movement. Participants (N=7) emphasized lateral movement as a key indicator for targeted attacks. APT actors are highly skilled in conducting extensive reconnaissance over several months to identify critical targets and initiate lateral movement. P9IR noted, "It's the work smarter, not harder mentality that drives the most success for APT actors." Participants observed that APT actors spend considerable time understanding and blending into systems, making them difficult to detect. As a result, lateral movement often serves as the first sign of their presence, as other actions tend to be slow and subtle. P9IR elaborated, "In many cases, when we catch them, we find that six or eight months earlier, they were already conducting reconnaissance in that space." P12IR further explained the importance of tracking APT actors' lateral movement, as it provides additional insight into their goals and the broader scope of the attack, mentioning, "This group over here is interested in point-of-sale machines. How do you know? Because they tried to attack it. They didn't get there, but they tried to move laterally to point-of-sale systems. Now we know that's part of their objective, even after containment."

APT attacks are characterized by multi-stage components. Participants (N=4) noted that APTs are often defined by their multi-stage, coordinated nature, with different components playing distinct roles in executing various parts of the attack over time. As P13IR explained, "Incidents often don't happen with just one person doing things. It's usually a continuous chain, where there might be malware on a system from a while ago that stole credentials or something like that. Then, the actors hand it over to someone else to take action on the box and do something against it." This highlights the importance of connecting seemingly disparate events and recognizing them as part of a larger, orchestrated attack rather than isolated incidents. P7IR stressed the need to thoroughly scrutinize each component in APT attacks, noting, "If you're dealing with cheap, off-the-mill malware, and you see it leaving a file, you might think it's just an error or a leftover artifact." But when dealing with APTs, they recognize different components as being part of a deliberate and coordinated effort and "look at it differently because [advanced actor] wouldn't just leave a file lying around randomly."

2.6.2 Incident Response Process for APTs

If an incident is classified as an APT-level attack, i.e., targeted, novel, and more sophisticated, participants (N=8) reported following a dedicated, more comprehensive, and collaborative protocol. This is consistent with Wong et al.'s malware analysis workflow, which suggested analysts perform more in-depth reviews when working with novel malware [273]. Below, we provide further details on the in-depth incident response (IR) process. Note that this process applies to both levels of attribution (TTP and country) as identified by the first layer in Figure 2.2.B. Additionally, we did not observe any clear differences in processes across participants from different organizations (industry, government, non-profit), apart from variations in the type of APT actors they prioritize. For example, the non-profit organization focused on analyzing and addressing threat actors targeting civil society groups. Meanwhile, the majority of our participants are from industry, and discussed prioritizing threat actors likely to impact their high-risk assets like cloud resources. Given our small sample space and lack of comprehensive data, these observations cannot be generalized to each industry, however, they may inform future research.

Analysts perform in-depth analysis for cases involving sophisticated or novel threats. Participants' response strategies shift from internal management of lower-level threats to increased investigative efforts for APTs. Participants noted that handling sophisticated threats involves a complex and layered approach (N=8). Initially, standard incident response procedures are followed, including memory dumps, disk images, log analysis, and network traffic examination. However, as P13IR noted, when an investigation reveals more serious indicators, such as attempts to manipulate code, the response escalates, and they "spin up a big bridge" and "get a bunch of other people... to take a look at [incident] and see what happened." This escalation often involves mobilizing a broader range of experts across the organization and conducting a detailed forensic investigation. For example, P11IR described doing a more extensive investigation of further lateral movement by examining "every single network artifact that has been touched," in contrast to routine malware responses that might only involve rolling credentials, disabling access, wiping, and reformatting devices.

APT incidents involve cross-team collaboration and coordination with external entities. P5MA, who is from a large organization (>5,000 employees) with a clear distinction between security teams, describes the structured, collaborative nature of an APT investigation, highlighting the division of labor between teams. In this scenario, "the malware team dissects the threat," while "threat intelligence is connecting the dots." P6TI further elaborated on the role of the threat intelligence team in guiding investigations and escalation. They explained "If we [the CTI team] do have even low confidence that an APT group is actually targeting us, we'll take it more seriously." They elaborate based on the initial bit of information they decide whether a threat is worth "becoming an investigation." Meanwhile, P7IR, who is from a government organization where privacy protocols differ significantly from those in the private sector, emphasized that "certain privacy precautions can be disabled and additional documentation is required" to ensure proper handling and coordination with external entities, such as other government agencies.

2.6.3 TTP Attribution

In this section, we discuss how, as part of the detailed investigation and collaboration, certain key activities are undertaken for TTP attribution (see Figure 2.2.B). Participants use historical knowledge to correlate incidents, conduct detailed analyses of malicious artifacts, and gather trusted intelligence. The ultimate goal of these steps is to identify the actions and objectives performed by the threat actor by closely attributing the tactics and techniques used to known patterns, thereby mitigating the incident's impact.

Participants use historical knowledge for correlating incidents. Participants (N=13) emphasized the importance of historical knowledge and threat intelligence in analyzing and correlating APT incidents. This process involves correlating and comparing the interconnected attack chain components—common in APTs (see Section 2.6.1)—to historical records, such as threat data from prior incident response engagements. As P1IR/MA noted "We leverage our research to collate and understand commonalities across engagements, building our own knowledge pool of adversaries." They further explain that they use automated methods for IoC hunting across the organization, stating

"We focus more on TTPs and IoCs, maintaining our own internal databases of confirmed threats." Participants also mentioned using manual methods for the historical lookups with P6TI using spreadsheets "with the date [threat report] was published, the source, the adversary [involved]" and P9IR documenting low-confidence indicators for future reference explaining to "note those things" to see if they have seen them before. P14IR/TI highlighted the maturity of their approach by emphasizing the focus on "TTPs rather than IoCs." Based on the concept of the Pyramid of Pain [22], they explain that "hunting based on tactics" is more robust, as relying on IoCs alone can generate an overwhelming number of "alerts."

Participants perform malware analysis using initial correlation of IoCs through threat intelligence, followed by in-depth reverse engineering. Most participants (N=11) investigate and correlate malware or file artifacts found in the attack chain using OSINT. This involves querying platforms like VirusTotal [260], Triage [208], and Joe Sandbox [103] to gather information about malware samples. Analysts consult these public forums to identify any existing intelligence that can aid in their investigation. As P11IR noted "Most of the time, we'll just do searches for the hashes to see if it's already been detected in the wild" and highlighted the use of subscription-based threat intelligence and private contracts noting "if we are confident that it is like an APT-level attack, then we would really leverage our internal, the private [threat intelligence] contracts. . . [to get] all of the details that is associated with the file, and the capabilities of the files."

Additionally, participants mention using a fuzzy hashing approach, such as ImpHash [206], peHash [266], SSDEEP [110] or internally developed techniques, to identify malware from the same family. For instance, ImpHash (Import Hash) generates a hash based on the import table of an executable, focusing on the functions or capabilities of the binary. Analysts typically obtain these hashes from malware analysis tools. These hashes are then used to search for other binaries that exhibit similar behaviors. As P11IR explained, "a lot of times you can then get a hash of those capabilities and then do searches on similar hashes to be able to identify if there are other files that match."

P7IR summarized the "two stages of attributing malware." The first stage involves correlating IoCs "by running the malware in a sandbox environment and seeing if it is trying to contact domain X, or it's creating file Y, or it has a request pattern that we've seen with malware Z." The second stage requires a deep dive into reverse engineering with tools like Binary Ninja [23] or IDA Pro [93], looking for "specific techniques, code styles, and decisions in the program logic" that can help correlate with other samples and is often done by dedicated malware analysts. However, P10TI also remarked when it comes to APT incidents while analyzing malware is important, "there's more than the malware." They emphasized the need to consider additional context, such as phishing lures and explained "Let's say the malware was propagated via phishing lures. There's the lures themselves. What do the lures say? Who could the lures be targeted at?" This broader context is important for a comprehensive analysis.

Participants use trusted relationships to share detailed information about

APTs that cannot be publicly disclosed. To exchange information about APT incidents and relevant artifacts, participants (N=7) emphasized the importance of trusted circles—a selective group of individuals who share intelligence privately among members, also mentioned in prior work by Bouwman et al. [28]. Sharing information on APT incidents is difficult due to the sensitivity and strategic nature of the data. P4IR discussed the balance between the need for information and the risk of over-disclosure, noting that trust is crucial for private exchanges: "When you're tracking a threat group and the threat group is trying to avoid being tracked, you don't want to give away too much." P6TI, from a medium-sized organization, described the daily sharing of threat data within a trusted circle, which includes discussions such as "Has anyone observed exploits targeting this CVE?" or "Has anyone seen this malware?" They described the operation of a Trust group, saying, "I also run the [anonymized] Trust group... a group of intelligence researchers that share threat data and threat information with each other on a daily basis. We have about 150 analysts." This behind-the-scenes collaboration enables the exchange of sensitive details about malware, CVEs, and IoCs facilitating attribution efforts.

2.6.4 Country Attribution

Toward the final stages of the investigation, when participants seek to precisely attribute the actor's location or origin, they use a combination of key attributes alongside identified TTPs and IoCs to perform country attribution (see Figure 2.2.C). Given the sensitivity and difficulty of this task, participants rely on confirming evidence from multiple indicators and look for clues beyond those considered in earlier levels of attribution, such as linguistic patterns or specific wording used by the threat actor.

Participants rely on IP addresses, domains, and C&C infrastructure. Most of the participants primarily rely on infrastructure elements such as IP addresses, domains, and command and control (C&C) channels for country attribution (N=12). P9IR elaborated that using IP and geolocation data can "narrow down the pool to a few possibilities. Then we would search for things like ISPs." P12IR further emphasized the importance of metadata-related infrastructure features, noting that "ASN off the IP" and "any registration data" such as the email address of the registrars, are useful for attribution. Participants also mentioned using time zone analysis to infer the threat actor's hours of operation and potentially their region (N=4). P8IR/MA explained that the North Koreans "were working six days a week from 9 to 9" which provides a useful indication for attribution based on the working hours of attacker and the timeline of attack.

Participants investigate the choice of wording. Participants (N=7) highlight the importance of analyzing the choice of wording and language within communications or malware code to infer the geographic or cultural origins of the threat actor. P1IR/MA points out that seemingly minor details, such as "the choice of the password or passphrase" can be quite revealing. Sometimes the actors "might throw in cheeky comments in their code" that can help in identifying the actor based on their vocabulary and language. P13IR adds that actors may "leave a message in the registry key" that could serve as their "call sign" significantly increasing confidence in identifying the origin of the attackers.

Participants investigate reused tools and exploits. Participants (N=7) also view reused binaries and exploits from previous incidents as valuable indicators for attribution. P1IR/MA mentions that a "[nation-state] state might have reused or slightly modified a piece of command and control" or a "backdoor Trojan" recovered during the incident allowing analysts to "infer some attribution" based on their familiarity with "something similar." P6TI further explains that if a threat actor uses a "certain piece of unique custom malware" it helps in attribution since nation-state actors who have "developed it themselves" and have not "shared it around to anyone else" leave a distinctly identifiable footprint.

Participants build threat actor profiles to inform and guide long-term remediation. One of the goals of country attribution is proactive threat actor tracking to guide organizational strategy (see Section 2.5). Towards that end, participants highlight the importance of developing comprehensive threat actor profiles for effective threat management. This process involves gathering intelligence from various sources, including OSINT, commercial threat intelligence platforms, incident response data and trusted intelligence-sharing groups. P6TI elaborated on this process within a sizeable CTI team: "We will divide our analysts up into regions like what regions they should be focusing on ... And I'll perform a quarterly report on Russian APT group." P6TI further explained how they extrapolate key information and perform mapping, stating "I'll try to extract the information from [threat] report into the diamond model." The Diamond Model [264] is a framework that examines the interactions between four key elements in a cyber attack: the adversary, their capabilities, the infrastructure used, and the victim. By mapping these relationships, the model allows analysts "to basically build up the database" for long-term planning. In practice, this process involves a deep analysis of threat intelligence sources, such as vendor reports and blogs, to associate aliases with specific threat actors. Analysts then assess how elements of the Diamond Model—such as adversaries, infrastructure, and TTPs—overlap across different reports. As P6TI described, "that's how we kind of get to Diamond Models that basically mean the same threat group." These overlaps help in building accurate threat actor profiles, enabling analysts to confirm that groups identified in various sources are indeed the same.

2.7 Result: Challenges (RQ3)

In this section, we explore the challenges participants encounter in investigating APT incidents and performing attribution. Section 2.7.1 highlights issues with relying on infrastructure features and the limitations of file analysis automation as APTs become more sophisticated. We also discuss the difficulties in data ingestion and the minimal use of machine learning, which impacts rapid correlation and attribution. Section 2.7.2 discusses challenges in current processes, such as the fragmentation of threat information across databases, inconsistencies in naming conventions, and the complexities of different threat intelligence sources, all of which affect the accuracy and integration of threat data.

2.7.1 Challenges in Tooling

Participants reported a wide range of specific challenges they faced when using tools to perform attribution related tasks. These were most often related to existing tools not supporting the data types and formats necessary for successful attribution as APTs and the TI ecosystem grow in complexity.

Lack of automation and validation in data ingestion impacts the use of historical threat data. As we discussed in Section 2.6.3, a key process for TTP attribution is querying threat data across historical records of threat intelligence and using a database of IoCs and TTPs. However, multiple participants discussed challenges in collecting and using this data (N=3). For example, P4IR mentioned a lack of tools "for crawling websites that publish reports or ingesting indicators from those reports," indicating a need for more automatic data ingestion and comparison. Participants acknowledge that for malware, there is a little more automation using machine learning; however, malware is just a small "part of a threat actor TTPs" (P10TI). To gain a sufficient understanding of threat actors, participants indicated they had to expend significant manual effort reading everything that has been written about a particular threat actor (N=9). As P13IR explained, maintaining threat actor information involves "a lot of parsing. . . It's a very manual effort."

Additionally, participants reported difficulty in not only collecting all relevant data but also validating it to avoid false positives (N=2). P13IR gave an example false positive describing the automated processing of threat intelligence reports which will indicate "the domain for this is google.com/something. Then, your intel feed down the line will be like let's look at all the domains in the reports here and then they'll say Google is bad." This is a growing concern as APTs more often use common or public infrastructure (see Section 2.6.1).

There is a lack of advanced and robust tooling to effectively analyze a variety of file formats. For malware correlation participants (N=2) face challenges when analyzing and correlating a diverse range of file types, particularly with the rise of cross-platform binaries. As an example, P10TI notes the shift from traditional languages like C and C++ to languages such as "Nim, Rust, and Golang," which allow threat actors to "target multiple platforms" simultaneously. Further, P5MA highlights the difficulties of dealing with these advancements, stating "We lack proper tooling to analyze files used in these big chains of attack." They explain "Especially for Windows and specific languages like .NET, Visual Basic, and Delphi . . . you need a lot of information, and there is a lack of tools" for effective analysis.

APTs using existing system tools rather than custom malware is making it difficult to attribute their activities. Further, to complicate correlation and attribution efforts, APT operations blend in with legitimate tools as a means of being stealthy (see Section 2.6.1). P7IR notes the increased use of built-in tools like PowerShell and threat actors pivoting toward "living off the land" (N=4). They explain: "We've observed more instances where attackers avoid using their own binaries or scripts as much as possible, relying instead on tools that are readily available on the victim's system." This poses challenges for traditional binary-based correlation methods as P11IR highlighted "There's been a really big push for some of the bigger APT groups to move away from binaries."

APTs exploit legitimate accounts and activities to evade automated detection systems. Participants highlighted key challenges in detecting and mitigating threats within complex environments (N=4). P15MA/TI mentioned that while automation streamlines a lot of their forensic analysis, such as mapping MITRE ATT&CK TTPs [175] to the specific threat actor, it often requires manual review and additional context as the automated system does not understand the "full context of the incident... if it was actually a legit user using their account or if it was the threat group using it." This challenge is exacerbated by attacks involving dormant adversaries within large networks. P6TI explained "[adversaries] can linger in your environment for years at a time. They only need to create an account and just keep it there... if you have an Active Directory with tens of thousands of users, it's really difficult to go through all of those and check."

Application of machine learning in APT correlation and attribution is limited. Despite the recognized need for robust and advanced automation for detecting and correlating APT activities, participants expressed reservations about utilizing machine learning (N=5). Key barriers to broader adoption included a lack of training resources, time constraints, insufficient datasets, the complexity of the models, and high false positive rates in alert generation. P7IR noted, "We have experimented with [machine learning]. But the results have not been, I don't want to say they have been bad. But not worth the effort." They further explained that this was not "necessarily a critique of machine learning approaches, it's more the reality of being severely resource constrained." P7IR explained this resource constraint was on the time available to tune machine learning tools to their specific environment, saying, "The only reason why we even experimented with it in the first place was because I decided to not sleep one night. And I couldn't justify, spending more of my own time on it because during regular office hours, I had other tasks to do." Beyond time constraints, we identify challenges with insufficient resources to train and fine-tune the models. P9IR added to this, emphasizing the challenges in "finding the resources to be able to train the models to do what you need," and pointed out that if not done properly can lead to high false positive rates and "ticket fatigue" in SOCs. These results are similar to Mink et al.'s [168] findings in discussions with SOC analysts regarding their use of machine learning for intrusion detection. Finally, another challenge in ML adoption lies in the lack of diverse malware datasets and the difficulty in explaining the decisions of complex models. However, the data scarcity problem for attribution is further complicated by the fact that effective machine-learning-based attribution requires many different comprehensive datasets. As P5MA explained, "To have the data set with all the functions for the different architectures for the different operating systems. . . it's really complex."

Apart from learning-based models, two participants mentioned LLMs for APT investigations. One described using internal GPT models for text summarization, noting that this usage is ad-hoc and not a company-wide system. Another participant shared their team's experience experimenting with Microsoft Copilot, highlighting its potential but also issues in dealing with meaningless and incorrect results. It is important to note that at the time of the interviews, LLMs were gaining traction, so it is possible that there have been changes in their use since then. Even so, our results offer insights into the process and can guide their effective adoption.

The reliance on IP addresses and domain names for country attribution is unreliable. As we discussed in Section 2.6.4, participants rely on infrastructure features such as IP addresses, C&C infrastructure, and domain names to identify threat group signatures. However, P11IR noted that "IP addresses and domains are not nearly as reliable of a correlation point anymore," highlighting the difficulties of APTs blending with legitimate infrastructure or using shared public infrastructure (see Section 2.6.1). P8IR/MA further points out that attribution becomes more complex in cloud-native environments, where "containers and instances are popping up and down all the time," making it challenging to track persistent infrastructure as it has become super easy to "spawn on a new machine somewhere in the world and just attack with it." Therefore, in most cases, this makes country attribution impossible in practice based on these infrastructure features.

APTs using shared infrastructure, overlapping malware, and selling attacks further complicate country attribution. Participants highlighted scenarios where attribution could be misleading or faked with P4IR stating "I do have some experiences of attribution getting very murky, like cases where, you know one threat actor might compromise and use another threat actor's infrastructure. Like we've seen some potential cases or indications where it looks like that might be what's happening" which could lead to "all kinds of misattributions." P6TI mentioned another scenario where "one company was developing all the Chinese malware that like ten different Chinese APT groups were using. So it's kind of a this is it comes back to this thing of, we may know it's Chinese APT group or China based group, but we don't know exactly which one," because they all share many capabilities these days. Finally, P9IR noted another challenge unique to country attribution is that "A lot of the APT cases had teams where we could see the A team that is doing the attack, and then the B team doing the attack. And usually, the B team is how we find them. But there's also once they're done with the attack, we know that they sell their attack on the dark web. And then criminals, just regular criminals could then use it in their attacks." These scenarios make it increasingly difficult to accurately identify the true source of APT attacks.

2.7.2 Challenges in Processes

In addition to the challenges our participants faced when using tools, they also encountered challenges in establishing an accessible and reliable threat intelligence ecosystem, as well as with effective collaboration within and among organizations.

Inconsistent data formats and naming conventions add difficulties in merging and correlating threat information from disparate sources. Participants highlighted the challenges of lack of standardization in threat information across different databases, even for government organizations (N=3). P9IR noted "CISA has a database for tracking one set of threat events, while the FDA maintains another for different events." This fragmentation complicates the process, as both organizations might be tracking the same actor without realizing it. P7IR further emphasized the challenge of sharing information as it might reveal sensitive data. They explained "No government is sharing their attacks with other governments. . . There are some standards like STIX, or using MISP, but in practice there is no secure way to do this."

Inconsistent threat naming conventions further complicate the process. Multiple participants noted that varying names for the same threat actors between reporting organizations adds to the confusion (N=13). P1IR/MA attributed this to the fact that they "haven't necessarily gone through a due diligence process to see what's out there already or made sure that this makes sense in relation to other publications that are similar. . . everyone's just busy speaking out what they think is useful. . . I think that's probably a large part resulting in some of the ambiguity and misinformation that we see."

Open-source threat intelligence has too many false positives; commercial products are too slow, so participants turn to unofficial sources. Participants in our study use both commercial threat intelligence and OSINT in their processes for TTP and country attribution (see Section 2.6). In addition to tooling performing poorly when attempting to ingest diverse TI, as described previously (see Section 2.7.1), participants identified issues in threat intelligence itself (N=7). Bouwman et al. [28] and Li et al. [126] already highlighted significant gaps in the accuracy, coverage, and timeliness of threat intelligence sources. Our results suggest a similar set of perceptions. First, our participants found OSINT unreliable due to its lack of important details and context. P2TI explained they have "tons of information which means you have more quantity and less quality. Less quality threat information means it's absolutely not attributed. . . What can I do now with the list of IP addresses?" The lack of quality information places the burden on analysts to validate the data—specifically, to understand which IP addresses are malicious, how long they should be blocked, and how to properly integrate this information into defense technologies.

Paid threat intelligence also has its challenges. When discussing paid threat intelligence reports, P14IR/TI said they were "probably our slowest means of actual detection creation. They tend to be quite granular but are less actionable." P14IR/TI elaborated that Mandiant or similar services "might identify a malicious IP address from an attack at another organization and include it in their intelligence product. By the time it reaches me as a customer, the attack has likely moved on." P9IR also echoes similar timeliness issues with the IoCs in commercial threat intelligence: "You have to be good about vetting the information" to make sure that the "IPs and domains are still valid, and the attacks are still relevant." In addition to supporting prior results, we discovered our participants go beyond official threat intelligence to unofficial sources like Twitter/X, Reddit, GitHub, and blog posts (N=3). P14IR/TI explained that they supplemented official threat intelligence through their own "live analysis of samples that are on the Internet... Twitter is a great source for this. If someone's seeing something that we think is malicious, then we can then take that in and stuff like, and use our own telemetry to work out what's going on."

Need for collaboration and open communication among teams for successful APT investigations. Several participants emphasized the role of cooperation and transparency among different entities for successful APT investigations (N=5). P9IR remarked "In all the different places I've worked, the tools have varied. For me, it's really about the people and having a collaborative team that is skilled and knowledgeable." They further noted that in siloed environments they "rarely saw progress toward identifying the source or attribution of the attacker" with investigations often hitting a wall. P9IR specifically highlighted an information sharing barrier when people "would only share details on a need-to-know basis" delaying the investigation. P3IR/TI expressed hope for a "shift in the industry" towards "more open and honest reporting on the activities of APT groups." P7IR echoed this sentiment, saying "Cooperating together is the only chance we really have at being successful."

2.8 Discussion and Conclusion

Attribution is a complex and nuanced process that balances the need for accurate identification of threat actors with the practical considerations of incident mitigation. Our findings highlight a disconnect between theories and practical realities, identifying that victim organization progresses through three distinct layers of increasing classification specificity depending on the incident and their situation. This decision process suggests that attribution should not be viewed as a single task but rather considered from one of the three layers identified, i.e., (A) APT classification, (B) TTP attribution, and (C) country attribution. Each layer presents unique goals, accuracy requirements, and challenges. Further, we found our participants often prioritized incident mitigation over identifying the perpetrator, focusing on understanding the incident's scope, assessing data compromise, and securing networks (see Section 2.5). Future research should explore whether this motivation extends beyond our sample, as our study offers valuable direction for attribution-focused research.

In addition to this decision process, we also observed several challenges for APT attribution in practice, which require further investigation. These challenges broadly are caused by existing tools being unable to support more diverse and complex data, as well as issues in collaboration, internally and externally, which is essential for the more complex levels of attribution. In this section, we provide recommendations for attribution tool development and threat intelligence sharing based on our results and discuss how existing efforts could be improved through these recommendations.

2.8.1 Recommendations for Tool Development

Identifying *what* should have priority over identifying *who*. We identified that practitioners are often less concerned with the specific entity behind an attack and more focused on determining that entity's typical TTPs and motivations. By understanding what the adversary might do, the organization can focus their incident response on systems and indicators associated with those TTPs and prioritize threats that pose the greatest risk to their organization.

Practitioners in the field express a need for more automated techniques that can accurately cluster TTPs to provide actionable insights. As P8IR/MA explained it was most important to have automation that gave them a starting point saying they wanted automation that would "give me a hint to work with about the attacker and then I can manually do some work. . . nowadays you don't have any where to start." Automating TTP-level attribution helps analysts quickly identify the relevant tactics and guide their response efforts. It serves as an initial filter, allowing analysts to narrow down the possibilities of TTPs and associated threat actors. Future research should prioritize identifying *what* threat actors are likely to do rather than focusing on *who* they are. This approach involves mapping low-level threat events to TTPs and correlating them with clusters of known TTPs using frameworks like MITRE ATT&CK. Moreover, TTP-based attribution should be interpretable, providing a high-level summary of the attack and guiding analysts in understanding the scope and magnitude of the incident. By shifting the focus towards TTP-level attribution and ensuring that these systems are computationally feasible and applicable in real-world scenarios, we can better equip practitioners to handle complex threats.

Existing attribution approaches primarily aim to identify the APT groups responsible for an incident [79, 211, 217, 263]. Focusing solely on group-based attribution can lead to missing TTPs, especially if a group modifies its tactics. Instead, by emphasizing commonalities across TTPs, even when tactics vary slightly between incidents, we can develop a more robust understanding of the specific methods used. This approach can identify nuances in how particular tactics are executed, which could otherwise be overlooked when analyzing across different adversaries.

Some research has advanced in mapping low-level events to TTPs for APT detection, with systems like HOLMES [164] and APTHunter [137] employing provenance analysis and mapping alerts to TTPs using the MITRE ATT&CK framework [175]. However, these approaches have limitations: (1) They rely on cumulative threat information from all attack stages, assuming that an APT attack completes the entire chain, and (2) they are evaluated with synthetic datasets in laboratory settings. These limitations hinder the adoption of these systems in real-world settings. Our study indicates that practitioners often lack a complete view of the attack chain initially and only uncover the full APT attack sequence through in-depth forensic analysis (see Section 2.6.2). Future research should build on existing APT attribution approaches by incorporating TTP coverage as a core metric. The effectiveness of such systems should be measured by their accuracy in identifying the correct TTPs, enabling analysts to get a comprehensive understanding of APT-level incidents.

Malware-based APT attribution demands a shift from basic binary clustering to include diverse artifacts. Attributing APTs is inherently complex, necessitating a multi-layered approach and the investigation of extensive data. As APTs increase in sophistication, reliance on infrastructure features becomes less reliable (see Section 2.6.4). The use of legitimate tools by APTs further complicates detection efforts, and the lack of tools for analyzing diverse file formats increases manual effort (see Section 2.7.1), a challenge also highlighted in our study on analyzing malicious documents [226].

Practitioners emphasize that analyzing individual artifacts in "gray areas" and understanding their connections to other components in an APT attack chain is important for assessing maliciousness. Malware clustering and classification solutions [3, 20, 83, 156, 169, 198, 215] have been researched for decades. However, current malware-based attribution practices, which focus primarily on classifying binary samples, often fail to provide comprehensive insights into APT-level activities. To enhance attribution depth, it is essential to incorporate a broader range of suspicious artifacts from the APT attack chain, such as phishing lures used to deploy malware and the exploitation of native binaries, such as PowerShell, and associated scripts. Our recent work ADAPT [225] demonstrates progress in this area by incorporating diverse file attributes for APT campaign and group attribution, highlighting the need to account for the heterogeneous artifacts in APT attack chains. Further, ADAPT employs features from secondary sources such as YARA rules and attributes from internet scanning databases such as Censys to augment malware-related features. Future research should build on these studies and develop robust automation for analyzing and extracting indicators from a diverse range of file types.

2.8.2 Recommendations for Threat Intelligence

Addressing the lack of standardization requires a combination of automated tools, manual review processes, and community collaboration. One of the major challenges in APT attribution is the inconsistent naming and labeling of threat actors and their associated TTPs [75]. As highlighted by our participants in Section 2.7.2, different organizations, research groups, and governments may assign different names or labels to the same APT group or activity based on their independent analyses, complicating the process of merging and correlating threat data, leading to delays in response efforts.

To address this challenge, future work should explore the development of a comprehensive registry that standardizes the naming conventions for APT groups and their associated TTPs. This registry could leverage existing automated relabeling approaches aimed at reclassifying and standardizing labeling for malware families [232]. To complement this automation, the registry could integrate analyst feedback to enhance potential mappings by using clustering techniques that suggest possible mappings between different naming conventions. These systems would not only propose mappings but also explain the similarities between different labels, helping analysts understand the explanation behind the proposed unification. A key feature of this registry would be its ability to facilitate manual review and validation. By allowing threat analysts to compare samples and naming conventions, a common challenge noted by our participants (see Section 2.7.2), the tool would help resolve attribution discrepancies and ensure that the standardization aligns with community consensus.

The issue of naming standardization is further complicated by the potential involvement of government agencies, which may have access to unique intelligence resources. As P4IR pointed out "Maybe these governments are doing their own attribution. Maybe they're like, oh, Mandiant, FireEye, Citizen Lab, like whatever. We're not going to even consider that. We're just going to go to our friends at the NSA who have this global view of the internet and ask them to do the attribution. Right. Like, yeah. So, I don't know, it it's always difficult to understand whether, you know, attribution is sort of being replicated behind the scenes or whether governments, for instance, are using the attribution of, of, you know, threat intelligence companies or groups." This emphasizes the need for greater transparency and collaboration between public and private sector entities in the attribution process.

Future research on threat intelligence should consider evaluating TTP coverage and accuracy, beyond traditional IoCs to address the complex nature of APT activities. Bowuman et al. [28] and Li et al. [126] explored the effectiveness of paid threat intelligence and OSINT by looking at the coverage, accuracy, and timeliness of the information presented in them. Specifically, they looked at the IoCs such as domain names, IP addresses, and file hashes. However, in our study, participants reported the unreliability of these features because of a lack of specificity and susceptibility to evasion techniques employed by sophisticated threat actors (see Section 2.7.1). The reliance on these weak IoCs does not adequately address the complexity of APT activities, which often involve sophisticated TTPs beyond simple indicators.

While some progress has been made, particularly in automating the extraction of IoCs from unstructured text [127] and TTPs from CTI reports [6] future research should build on these studies by emphasizing the evaluation of TTP coverage and accuracy associated with threat actors. This involves assessing how well current threat intelligence solutions capture and represent the complex behaviors and capabilities of APTs. Additionally, further research should focus on developing metrics to measure TTP coverage across different threat intelligence sources.

Authorship details

As a first author, I share the sole authorship of conducting the expert interviews, codebook development, and manuscript-writing. I relied on James Mattei to compute the inter-rater reliability (IRR) during the data analysis and code book finalization phase.

CHAPTER 3

ADAPT it! Automating APT Campaign and Group Attribution by Leveraging and Linking Heterogeneous Files

Recent years have witnessed a surge in the growth of Advanced Persistent Threats (APTs), with significant challenges to the security landscape, affecting industry, governance, and democracy. The ever-growing number of actors and the complexity of their campaigns have made it difficult for defenders to track and attribute these malicious activities effectively. Traditionally, researchers relied on threat intelligence to track APTs. However, this often led to fragmented information, delays in connecting campaigns with specific threat groups, and misattribution.

In response to these challenges, we introduce ADAPT, a machine learning-based approach for automatically attributing APTs at two levels: (1) the threat campaign level, to identify samples with similar objectives, and (2) the threat group level, to identify samples operated by the same entity. ADAPT supports a variety of heterogeneous file types targeting different platforms, including executables and documents, and uses linking features to find connections between them. We evaluate ADAPT on a reference dataset from MITRE as well as a comprehensive, label-standardized dataset of 6,134 APT samples belonging to 92 threat groups. Using real-world case studies, we demonstrate that ADAPT effectively identifies clusters representing threat campaigns and associates them with their respective groups.

3.1 Introduction

In contrast to conventional malware threats, an Advanced Persistent Threat (APT) represents an adversary that pursues its objectives over an extended timeframe, adapts to defenders' countermeasures, and remains committed to maintaining the necessary level of engagement to accomplish specific goals [185]. These goals often involve exfiltrating sensitive data of economic and political significance while maintaining prolonged access within the target organization. The complexity of these attacks necessitates adversaries who are well-funded professionals and, in many cases, even have state sponsorship, enabling them to operate with the support of military or state intelligence agencies [44].

For example, in January 2021, Google's Threat Analysis Group (TAG) detected an APT campaign aiming to compromise security researchers through 0-day exploits and social engineering tactics, which involved the use of fake Twitter and LinkedIn profiles [265]. This campaign entailed months-long conversations to establish trust with the targets. Strategically luring researchers, the threat actors shared a link via Twitter, blog.br0vvnn[.]io. Visiting the link installed a malicious service on the researchers' system, providing a backdoor to an actor-owned command and control server. In September 2023, TAG uncovered a new campaign, likely orchestrated by the same threat actors [123]. Both the 2021 and 2023 campaigns were coordinated operations targeting specific entities and employing similar tactics. TAG identified these threat actors as likely connected to a government-backed entity in North Korea.

As highlighted by this and similar incidents [212, 257], APT groups are well-organized entities, planning and executing multiple campaigns over time. To defend against these adversaries, the security community relies on prior knowledge of APT campaigns and groups. However, as threat campaigns and threat groups are getting more complex and sophisticated, it becomes challenging for researchers to accurately track and attribute attacks. This challenge is further exacerbated by the use of different nomenclatures and methodologies employed by security vendors to organize APT activities [163, 219]. The absence of structured, comprehensive, and easily accessible data on threat campaigns and groups hinders timely campaign identification and group attribution, thus impacting defenses [163]. For example, MITRE ATT&CK serves as a valuable resource to track threat groups and their tactics, techniques, and procedures (TTPs) [171]. However, its exclusive focus on APT groups leads to a lack of precise information on campaigns and their associated samples. Even though MITRE introduced the APT Campaign Framework in September 2022, it currently only lists 30 campaigns [172].

In response to these challenges, we introduce ADAPT ("Attributing Diverse APT Samples"), an automated machine-learning-based approach that allows for both APT *group* and *campaign* attribution. Unlike malware detection, which identifies and recognizes malicious software within a system or network, attribution seeks to determine if a malicious sample is associated with an APT campaign or group, providing insight into the adversary's tactics and identity. To the best of our knowledge, ADAPT is the first attribution approach that uniquely focuses on two crucial levels of analysis, i.e., the

APT campaign level and the APT group level. Attribution at the campaign level reveals specific campaign details, including tactics, techniques, and objectives, while attribution at the group level identifies the responsible entity behind the attack.

An automated attribution approach that promptly clusters samples to an APT campaign or group, facilitates forensic investigations. This, in turn, enables the security community to take proactive measures, such as compiling comprehensive threat reports for Open Source Cyber Threat Intelligence (OSCTI) and alerting organizations, ultimately reducing the risk of subsequent intrusions.

Previous research on malware clustering and classification has primarily focused on identifying malware families and variants without dealing with the specific domain of APT malware [3, 20, 198, 215]. Meanwhile, research on identifying APT attacks primarily focused on network-based event analysis [10, 72, 223], which can have limited visibility on APT actions. Provenance-based anomaly detection offers a complementary approach [80, 100, 125], but suffers from dependency explosion problems and challenges in recovering complex causality relationships [87]. A handful of existing research attempted to attribute malicious samples to APT groups by relying on executable features such as basic string and code features [263], function encodings from decompiled code [169], and API call sequences [79]. Nevertheless, a notable gap remains in exploring the distinctive characteristics of APTs: Threat groups use multi-stage attack campaigns frequently leveraging a variety of file types and cross-platform samples, including 0-day exploits and custom-developed malware [38, 50, 144]. To extend the state of the art, which predominantly focused on Windows-based executables, ADAPT performs feature extraction and clustering of heterogeneous file types, including executables (e.g., PE and ELF binaries) and documents (e.g., Word documents, PDFs, and RTFs). Most notably, ADAPT uses a novel set of linking features to find connections between these different file types.

In summary, we make the following contributions:

- We compile a first-of-its-kind APT dataset of 6,134 samples encompassing heterogeneous file types from the past 17 years. We manually (re)-label and identify 92 unique APT groups in the dataset. We further create a reference dataset for APT campaigns consisting of 230 samples from 22 campaigns and 17 groups.

- We develop a novel methodology and make the first attempt at APT campaign attribution for both executable and document file types, achieving a clustering precision of 93% and 95%, respectively, on the reference dataset.

- We identify and extract a set of linking features that can facilitate sample correlation, regardless of the file types, and show promising results in APT group attribution using illustrative case studies.

Artifacts: We provide our source code, features, and labeled dataset of diverse APT samples, including executables and documents, at https://github.com/SecPriv/adapt.

3.2 Background and Motivation

A *threat group*, synonymous with the terms "threat actor" or "adversary," refers to individuals and groups that pose threats but do not necessarily imply authorship [186]. Threat groups conduct *threat campaigns*, which represent specific instances of coordinated threat events associated with one or multiple threat sources, typically organized sequentially over time [187]. Each campaign usually involves a distinct set of targets, tools, and methods employed by a threat group to accomplish specific objectives.

Equipped with these definitions, we show a simplified example based on the Sidewinder APT in Figure 3.1. During the initial phase of the *threat campaign*, targeting Windows, as seen in the first highlighted block, the *threat sources* encompass a phishing email, a macro-enabled document, and a PE binary. The second highlighted block depicts a similar threat campaign conducted with altered attack vectors, now targeting Linux. In this case, the threat sources comprise a phishing email, a vulnerable RTF reader, and an ELF binary. The *threat group* is the entity that orchestrates these multiple threat campaigns targeting various organizations.

An analyst tasked with investigating the incidents similar to Figure 3.1 would begin by analyzing documents and executables separately, taking into consideration the intrinsic differences in layout, content, and the compilation process of executable threat sources (PE, ELF) and document threat sources. They then would attempt to correlate the following:

1. The macro-enabled document and the exploit-laden RTF in the document domain.
2. The PE binary dropped by the macro-enabled document, and the ELF binary dropped by the RTF in the executable domain.
3. Using the command & control (C&C) infrastructure (IP addresses, domains, and other patterns), they attempt to attribute the threat campaign X (and further campaigns Y and Z) to the threat group operating the domain $foobar.evil.com$.

In the next section, we go into more details of the Sidewinder APT as a case study to illustrate the challenges specific to correlating threat sources from multiple campaigns and associating the campaigns with their respective threat groups.

3.2.1 Motivational Case Study: Sidewinder

Sidewinder, also known as Rattlesnake and T-APT-04 [7, 173], is a threat group believed to be based out of India. In particular, we focus on how the group uses heterogeneous files and platforms, adapts their campaigns based on their goals, and the issues around shared tooling and code reuse between groups.

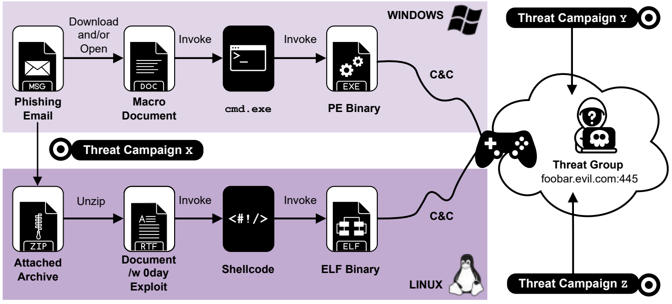

Figure 3.1: Simplified example based on the Sidewinder APT. A *threat group* conducts a number of *threat campaigns* that are targeting multiple platforms (e.g., Windows and Linux) using different file types, including macro-enabled or exploit-laden documents that drop platform-specific executables (PE and ELF).

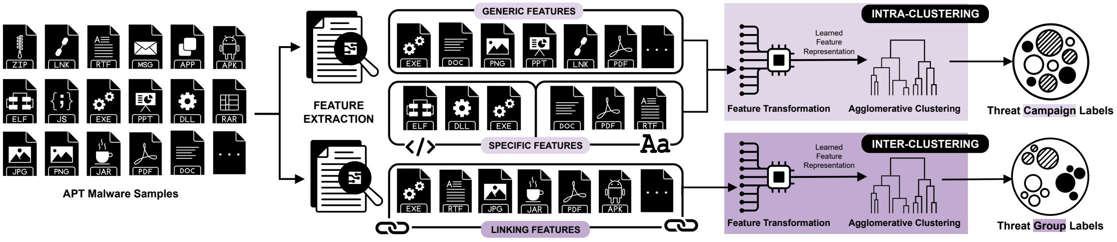

Figure 3.2: Overview of ADAPT. The first workflow (*Intra-Clustering*) groups executables and documents using specific and generic features to identify threat campaigns. The second workflow (*Inter-Clustering*) uses linking features across all file types to identify threat groups.

Heterogeneous Files and Platforms. Sidewinder uses malicious documents as one of the most common infection vectors. In addition, analysts have also observed the use of various file types in different campaigns for delivering malicious payloads [7, 50]. These file types include LNK files used to download RTF files that subsequently drop JavaScript files or ZIP files containing LNK files that download HTA files with JavaScript. Further, these files vary over the years and often evolve with each campaign in an attempt to complicate analysis and evade detection. Moreover, some of the document file types are cross-platform and embed platform-specific exploits. A noteworthy similarity lies in the group's utilization of the Microsoft Office memory corruption vulnerability CVE-2017-11882 [188] as a means to initiate the compromise on target hosts. Additionally, Sidewinder has also been spotted using the Binder exploit to attack mobile devices [256]. This demonstrates how the group strategically employs multiple file types and exploits multiple platforms, utilizing 0-day vulnerabilities, to significantly increase the likelihood of a successful attack. While the utilization of a variety of file formats, besides the widespread Windows PE binaries, is a technique employed by both APTs and conventional malware threats, as noted in previous studies [27, 226, 241], our research is specifically focused on APTs and their distinctive characteristics.

Varied and Persistent Campaigns. Sidewinder uses spear phishing to obtain credentials from targeted organizations [250] including the distribution of maliciously crafted documents containing executable payloads [50]. The themes and topics of phishing pages and malicious documents adapt to the campaign's objectives. These objectives can involve gaining sensitive information related to COVID-19 research or territorial disputes involving Nepal, Pakistan, India, and China [7]. These variations have been described as the group's effort to craft unique campaigns based on the target organization and global affairs [256]. In 2022, Sidewinder introduced a new custom tool (`SideWinder.AntiBot.Script`) to redirect victims to download the initial payload from a compromised website [84]. Sidewinder's persistence and varying tactics across multiple targets complicate attribution. Security vendors tracking the same APT group from different campaign perspectives often generate overlapping or fragmented information [213, 250], leading to an incomplete understanding.

Shared Similarities. The utilization of common components in the attack chain, code reuse, and the sharing of toolkits among different threat groups also create challenges for researchers in accurately attributing a sample to its respective threat group [19]. For instance, in at least two investigations, code sharing among APT groups with opposing targets was highlighted [39, 246]. Initially, analysts associated a set of samples with either Sidewinder or Donot groups [51]. However, on further investigation, these samples were conclusively linked to a third APT group, Transparent Tribe [174], also known as APT36. Transparent Tribe has been active in South Asia for over five years, primarily targeting the Indian government and military organizations. Despite different target regions (India for Transparent Tribe and Pakistan for Sidewinder and Donot), these groups reused portions of the Visual Basic Analysis (VBA) code. The code similarity among the groups resulted in inconsistent and erroneous attribution. To further complicate the matter, many APT groups consist of subgroups, each assigned specific tasks. The collaboration among various APTs makes it challenging to confidently track and attribute actions to a single entity [18].

3.3 Our Approach: ADAPT

Motivated by the challenges detailed above, we propose ADAPT, an attribution approach specifically designed to handle heterogeneous file types. ADAPT streamlines the attribution process by clustering samples at two levels. First, at the *campaign level*, it attributes samples based on their specific tactics and techniques. This means that regardless of the campaign's evolving nature or changing names over time, ADAPT's focus on the inherent functionalities enables us to associate specific samples with an APT campaign. Second, at the *group level*, ADAPT considers infrastructure and operational traits, providing insights into the broader context within which campaigns operate and attributes samples based on the characteristics of the APT group. Clustering malware samples is a routine task performed by malware analysts, as demonstrated by Yong et al. [273]. This highlights the importance of automating forensic investigations of APTs through campaign- and group-level clustering

Figure 3.2 illustrates our approach for analysts using ADAPT during security incident investigations. Instead of manual correlation of diverse threat sources, analysts can leverage ADAPT to *automatically* identify if recovered malicious artifacts exhibit indicators of known or ongoing APT campaigns stored in the APT dataset (see Section 3.4). ADAPT's feature extraction first identifies the file type and extracts relevant static attributes. Subsequently, for executables and documents, ADAPT's *Intra-Clustering* attributes the samples to campaigns (see Section 3.5). This categorization streamlines classification and provides insights into campaign tactics and techniques, facilitating sample analysis. ADAPT's *Inter-Clustering* enables grouping samples by the threat actor (see Section 3.6) leveraging linking attributes to facilitate correlation based on distinct characteristics and operational patterns.

Note that we maintain a dedicated clustering step for group and campaign attribution for two primary reasons. First, as highlighted in our motivating case study, threat groups execute a wide range of campaigns targeting various organizations. These campaigns, initially unattributed, become linked to an APT group following thorough analysis and information gathered from prior campaigns [145, 258]. Second, APT attacks typically involve multiple file types as shown in our motivational study. Therefore, having a dedicated group attribution process that accommodates all artifacts associated with different campaigns proves valuable for linking and associating samples. This, in turn, simplifies attributing and prosecuting recurring APT attacks, as exemplified in the FBI indictments of APT10, APT29, and APT41 [64, 65, 247].

3.4 APT Dataset

In this section, we describe our APT sample collection process, as well as the labeling process that we incorporated to ensure consistency within the dataset.

3.4.1 Group-labeled Dataset

Data Collection

To collect APT sample hashes, we use AlienVault's DirectConnect API [192], leveraging their comprehensive threat intelligence database. By specifically querying for APT-related pulses, we retrieved 5,990 unique SHA256 hashes associated with a minimum of 172 threat groups. The threat group labels are crowd-sourced from the AlienVault community and we collect them along with the APT hashes. To further expand our dataset and ensure it includes the latest hashes we extract unique SHA256 hashes from threat reports published by Unit42 [257] and Mandiant [32] between January 2022 and March 2023. We chose Unit42 and Mandiant as they are reputable sources known for their reliable and up-to-date threat reports. Through this effort, we discovered 465 hashes attributed to ten threat groups. Notably, most of these hashes were explicitly linked with the Gamaredon APT group, highlighting their prominent role in the Russia-Ukraine conflict.

Since AlienVault only provides hashes and Indicators of Compromise (IoCs), we use VirusTotal [260] to download the corresponding 6,455 samples. We also query VirusTotal to obtain analysis reports for the downloaded samples and extract metadata such as the file type, creation date, and first submission date. The most recent sample in our dataset was first submitted in March 2023, while the earliest sample submission date was in May 2006. Note that the available number of samples for APTs is significantly smaller compared to generic malware due to their targeted nature. Unlike commodity malware with its indiscriminate victim selection, APT malware focuses on high-value targets, aiming for significant impact (often espionage or sabotage) and avoids mass distribution. This limited presence makes them scarce for collection and analysis, as noted in prior studies [75, 217].

Data (Re-)Labeling

2,260 samples (35.01%) in our dataset have more than one APT name or alias, and 239 (3.7%) samples are labeled with five or more group names, with the highest number of labels being 15 for two samples. For instance, some samples are tagged as 'Bluenoroff,' 'AppleJeus,' or 'Hidden Cobra' which are all aliases originating from different campaigns of Lazarus. This highlights the challenges analysts face when correlating campaign and group data. To address the lack of standardized labels, we conduct a thorough revision and relabeling process using Malpedia [140] and MITRE [171] and establish consistent group labels. In instances when a reliable label could not be confirmed due to disagreements between Malpedia and MITRE, two researchers independently reviewed threat reports from Unit42 and Mandiant. They each assigned potential threat group labels and then discussed and resolved any mismatches or conflicts. This process included addressing the following issues:

Consistency of Existing Labels: Aliases. Numerous samples are tagged with aliases representing the same APT group, e.g., 'Refined Kitten' and 'Elfin,' which correspond to APT33. We standardize these aliases following classifications by Malpedia and MITRE.

Consistency of Existing Labels: Umbrella Names. We eliminate text-based variations and adopt a consistent naming convention of APTXX, FINXX, or TAXXX for the groups whenever feasible. For instance, we assign the name APT32 for 'OceanLotus' and 'Sea Lotus,' and APT43 for 'Kimsuky,' while APT24 encompasses 'Pitty Putter' and 'Pitty Tiger,' following Malpedia's classification.

Consistency of Existing Labels: Non-unique Names. We manually review non-unique names like 'Transparent Tribe,' 'transparenttribe,' or 'Transparent Tribe Group,' and assign the name 'TransparentTribe' as the uniform representation.

Outliers: Non-APT Samples. We came across 122 (1.89%) samples linked to the FireEye Red Team tools that were stolen during the SolarWinds attack [254]. We classify them as 'NotAPT' since they are not actual attacker tools, but instead part of the data exfiltrated by the attackers. We also came across ransomware families originally labeled as ‘Egregor’ or ‘Maze Team’ and classified them as ‘NotAPT.’ Our decision is based on the observation that these ransomware families are employed by multiple cybercrime groups for financial gain and are not typically associated with a specific APT group. We remove these samples from our dataset.

Unlabeled Threat Groups. We identified 44 (0.68%) samples for which we could not determine the APT group label with certainty due to two main factors. First, certain threat reports utilize internal operation names that are not associated with any specific groups by Malpedia or MITRE. Second, some threat reports discuss the likely origin of the attack without providing a conclusive group name. We retain these samples without labels in our dataset.

Dataset Characteristics

After following the approach outlined above and excluding 321 (4.97%) ‘NotAPT’ samples, we are left with a dataset of 6,134 samples. The (re-)labeling effort identified 92 distinct APT groups, resulting in a decrease of 80 tags compared to the initial 172 group tags extracted from AlienVault. Table 3.1 shows the sample count for the top 15 APT groups in our dataset, along with the number of aliases provided by Malpedia (lower bound). The median sample count for APT groups in our dataset is 24.

Diversity of File Types. To extract file type information, we rely on the VirusTotal file analysis reports associated with the samples. Through experimentation, we observed that VirusTotal consistently provided the most accurate and standardized file type information compared to other methods, such as libmagic [91]. Table 3.2 provides an overview of the number of distinct file types in our dataset, as well as whether ADAPT treats them as executables or documents (see Section 3.5). Our dataset contains a total of 3,603 (58.73%) executable binaries, including samples targeting Windows, Linux, and macOS, the latter of which have received limited attention from the research community so far [46, 132]. The document file class is another understudied domain [226] and consists of 1,611 (26.26%) files, includes formats such as Microsoft Word, Excel, PowerPoint, RTF, PDF, and ZIP, Note that we consider ZIP files as documents because our feature extraction is capable of extracting attributes from ZIP files that contain valid document formats. Finally, our APT dataset also includes Android apps (APK), Windows shortcuts (LNK), and script files, as well as 152 unknown file formats.

3.4.2 Campaign-labeled Dataset

Compared to datasets labeled by threat groups, the scarcity of publicly available, structured, and comprehensive information for threat campaigns presents an even bigger challenge. To address this gap and build a reference dataset for threat campaigns, we use MITRE’s APT Campaign Framework [172]. This campaign tracking data lists campaigns with standardized identifiers (CXXXX format). For example, the C00024 campaign is associated with the SolarWinds compromise by the threat group APT29.

Table 3.1: Top 15 threat groups in our APT dataset.

Threat Group Label	**Number of Aliases**	**Sample Count**
Lazarus	29	527 (8.59%)
Gamaredon	11	446 (7.27%)
TransparentTribe	9	403 (6.56%)
WizardSpider	3	401 (6.53%)
TA505	9	307 (5.00%)
FIN7	8	293 (4.77%)
APT41	16	278 (4.53%)
APT1	11	251 (4.09%)
APT29	15	224 (3.65%)
Turla	21	203 (3.30%)
Kimsuky	5	173 (2.82%)
APT28	23	169 (2.75%)
APT32	15	147 (2.39%)
Sidewinder	4	133 (2.16%)
APT34	10	126 (2.05%)
Others	-	2,054 (33.48%)
Total	-	6,134

We extracted MD5, SHA1, and SHA256 file hashes from MITRE reports and downloaded the corresponding samples using VirusTotal. However, since not all samples were available on VirusTotal, we expanded the dataset with samples mentioned in threat reports from Mandiant [146], which offered in-depth coverage of APT campaigns as recent as March 2024. This combined effort resulted in the successful download of 231 samples. To ensure accuracy, we thoroughly examine the corresponding threat reports for each sample, identifying and removing any false positives and assigning the appropriate campaign label and group label. Our final dataset consists of 230 samples representing malware from 22 campaigns attributed to 17 distinct groups. These samples encompass various file types, including 142 executable binaries (139 PE, 2 MachO, 1 ELF) and 62 documents (34 DOCX, 9 ZIP, 8 DOC, 5 PDF, 4 HWP, 1 XLSX). We make both this *campaign-labeled* as well as the above *group-labeled* dataset publicly available.

3.5 APT Campaign Attribution

Unlike detection tasks that solely focus on identifying maliciousness, our research automates the analyst's workflow for malware campaign classification. This process involves grouping attacks based on shared characteristics, necessitating information beyond simple "benign" vs. "malicious" attributes. Establishing connections with known threat campaigns (related executables and documents) serves as a crucial starting point for prioritizing investigation in identifying attacker objectives and potential consequences [152, 273].

As shown in Table 3.2, executables and documents constitute the majority of our dataset. We extract features specifically tailored for these prevalent file types. Beyond this *file-*

Table 3.2: Different file types in our curated APT dataset. The file class indicates whether ADAPT specifically handles files as executables (</>), documents (📄), or only extracts generic features.

File Type	File Class	Platform	Total #
Windows Portable Executable (PE; EXE)	</>	Windows	2,516 (41.01%)
Windows Portable Executable (PE; DLL)	</>	Windows	1,019 (16.61%)
OOXML Document	📄	Cross-Platform	286 (4.66%)
Microsoft Word Document (DOC[X])	📄	Cross-Platform	262 (4.27%)
Rich Text Format (RTF)	📄	Cross-Platform	245 (3.99%)
Microsoft Excel Spreadsheet (XLS[X])	📄	Cross-Platform	239 (3.89%)
Android Application Package (APK)		Android	227 (3.70%)
Windows shortcut (LNK)		Windows	223 (3.63%)
ZIP Archive	📄	Cross-Platform	129 (2.10%)
OOXML Spreadsheet	📄	Cross-Platform	104 (1.69%)
Text		Cross-Platform	91 (1.48%)
JavaScript (JS)		Cross-Platform	66 (1.07%)
Hypertext Markup Language (HTML)		Cross-Platform	65 (1.05%)
Portable Document Format (PDF)	📄	Cross-Platform	60 (0.97%)
Visual Basic for Applications (VBA)		Cross-Platform	56 (0.91%)
Powershell		Windows	44 (0.71%)
Executable and Linkable Format (ELF)	</>	Linux	39 (0.63%)
Adobe Flash		Cross-Platform	38 (0.61%)
RAR Archive		Cross-Platform	38 0.61%)
Mach Object (MachO)	</>	macOS	29 (0.47%)
Hangul Word Processor (HWP)	📄	Cross-Platform	27 (0.44%)
Microsoft PowerPoint (PPT[X])	📄	Cross-Platform	15 (0.24%)
Others		-	355 (5.78%)
22+ File Classes		4+ Platforms	6,134

specific features, we enrich both executable and document representations with *generic features* derived from their string content. These generic features, extractable without specialized parsing, target the detection of malicious techniques and patterns and apply to various file types. While we do not perform specific feature extraction for APK and LNK files, as well as scripts, in this work, the incorporated generic features are applicable to these file types as well. We acknowledge the potential for future exploration in this area. Our approach prioritizes identifying the most critical features that characterize campaign-level similarities, rather than creating an exhaustive list of indicators. Focusing on these key features enables interpretable and automated campaign attribution across diverse malicious file types, as demonstrated by our case studies (see Section 3.9).

Executable Specific Features (EXF). Leveraging prior research in malware detection [9], [46], [241], and classification [3], [156] we select features that provide generalizability across the executable file types while enabling fast clustering. Our decision not to use type- and platform-specific features collected through more heavy-weight static and dynamic analysis (e.g., disassembled executables, call graphs, and system-level execution traces) allows us to compile a simple yet representative feature set. This reduces limitations associated with missed run-time behaviors due to the unavailability of attacker infrastructure or dependence on specific host artifacts.

We extract a comprehensive set of features from executables, such as section names, libraries, and imported/exported functions. We use LIEF [248] to parse and extract these features from the 3,603 (3,535 PE files, 39 ELF files, and 29 MachO) executable binaries. Because imported functions produce a large number of values with little discriminative power, and libraries, and section names are common to many different types of malware, we narrow down the feature set to focus on *exported functions* and the *configuration version* for PE executables. These features demonstrated discriminative potential in our initial experiments on a subset of malicious samples from distinct campaigns operated by the same threat actor.

For PE, ELF, and MachO executables, we extract 15,047 unique exported functions from 3,603 binaries, such as `DllRegisterServer`, `DllUnregisterServer`, `runtime.gosched_m`, `FileRipper`, `net. dnsDefaultSearch`, and `runtime.prefetcht0`. Additionally, we identify 11 unique configuration values for the PE executables, such as `WIN_VERSION.SEH`, `WIN_VERSION.WIN_8_1`, and `WIN_VERSION. WIN10_0_15002`, which can provide indicators about the attacker's build system used in a specific campaign.

Document Specific Features (DCF). From the document file types, we extract features including macros, obfuscated strings, document author (if available), application language, and suspicious keywords. Similar to prior research on document analysis [108, 167], we use the open-source Python package oletools [117] to parse and extract malicious content from document file formats, including ZIP. For ZIP archives, we iteratively parse and extract attributes from individual files. Consequently, we obtain features from 1,552 files (96.33%) out of a total of 1,611 document samples. Among those, we identify two informative and distinctive features for our clustering algorithms: the *Application Language Code* and the list of *Suspicious Keywords* with a total of 2,578 unique values. Suspicious keywords include malicious patterns such as auto-executable macros, VBA keywords used by malware, anti-sandboxing, and anti-virtualization techniques identified through pattern matching on macro scripts embedded in document formats (e.g., Word and Excel documents). Examples of suspicious keywords include `keywords-AMANICRYPTED.exe`, `keywords-PrivateFunctionF()`, and `keywords-AutoExecute`. Finally, examples of language codes include `1251: ANSI Cyrillic (Windows)`, `ANSI/OEM Korean (Unified Hangul Code)`, which are standardized codes used to identify specific languages within the application.

Generic Features: Capabilities (CAP). We use features to detect and extract distinct characteristics and capabilities in malicious files. To do this, we leverage the targeted and effective set of YARA rules developed by the Malcat Community [138]. These rules are designed to detect modular capabilities within the program, such as code injection, remote thread routines, privilege escalation for lateral movement, persistence using scheduled tasks, as well as packers. With a comprehensive set of 108 rules, we successfully identified at least one rule for 4,026 samples (65.63%). We also considered using the Yara-Rules [272] and Elastic's security detection rules [59] but found that they often exhibit noise or insufficient coverage.

Following the extraction, we use one-hot encoding to transform the categorical features from both executables and documents. These features collectively form the set S of all categorical features, $S = S_1, S_2, S_3, ..., S_i$. For instance, consider the feature "Capabilities", (S_1), with values `MSVC_2017_linker`, `MSVC_2017_rich`, and `DownloadUsingWinHttp`, where $i = 1$ and $k = 3$ unique values. To represent each value, we employ a binary vector S_{ik} where, $S_{ik} \in \{0, 1\}$. We consider the entire set of unique values, assigning a value of 1 if a particular element is present in the sample.

Generic Features: Strings (STR). Extracting strings from malware samples can give insights into a threat group's preferred syntactic construction and vocabulary, which can be used to identify the associated campaigns. We use FLOSS [147] to extract all possible ASCII and UTF texts from 6,114 samples (99.67%). In addition to extracting embedded ASCII and UTF strings, FLOSS is also capable of identifying stack strings (strings constructed on the stack at run time) and decoded strings (strings decoded in a function) from PE files, which can enhance the basic static analysis of malicious files.

We preprocess and filter the extracted strings to remove numeric characters, special characters, non-printable characters, spaces, and stop words. We also perform Unicode normalization to decompose Unicode characters into their individual components, ensuring consistent representation (e.g., 随 → U+968F). After preprocessing, we employ the CountVectorizer technique with n-grams. This method calculates the frequency of string tokens in the document. The effectiveness of this approach has been demonstrated in prior research on malicious application attribution and malware source code identification [106, 216]. In our implementation, we set the parameters of the vectorizer to use n-grams of size 1 to 3, with a maximum of 10,000 features. After obtaining the vectorized representation of the string tokens, we normalize the values of each token. This step plays a crucial role in integrating the string features with other numerical and categorical features, ensuring that all features are scaled within the range of 0 to 1 for the clustering task.

3.6 APT Group Attribution

Achieving both campaign and group attribution necessitates addressing the inherent heterogeneity of artifacts employed across multiple APT campaigns by a specific group (see Section 3.2). Thus, our feature selection process leverages domain expertise to identify attributes that effectively link these diverse threat sources with group-level signatures. For instance, in the supply chain attack uncovered by Mandiant [148], the analysis of 11 distinct campaigns targeting diverse sectors revealed the use of shared resources (digital certificates, infrastructure, development tools) across all campaigns, indicating a single threat group. Inspired by real-world investigations [148, 149, 257], we focus on extracting linkable features commonly used by analysts to connect malicious artifacts across campaigns to threat actors. ADAPT automates the process of extracting *linking features* beyond basic file analysis to facilitate group attribution in APT incidents.

Pattern-based (PAT). To identify the linking characteristics, we first extract specific patterns from the string features, part of our generic features (see Section 3.5). These patterns include IP addresses, URLs, authentication keys, API keys (e.g., Slack, Gmail, and AWS), embedded MD5, SHA1, SHA256 hashes, Bitcoin addresses, email addresses, and Unix and Windows file paths. We choose these features as they can reveal important traits about the operating threat group. For example, the presence of a specific file path like `C:\Windows\System32\drivers\<susp_driver_name>.sys` within an APT sample could indicate a threat group's preference for installing malicious drivers in a specific location. We leverage a set of 24 regular expressions to automate pattern matching across our samples. Out of 6,134 samples in our dataset, we found one or more patterns in 4,506 samples (73.45%).

Infrastructure (INF). After extracting URLs and IP addresses statically from the sample's string content, we then use them to collect more detailed infrastructure information. This includes the BGP prefix, autonomous system number (ASN), country code, certificate fingerprint, and issuer organization. Leveraging the Censys search engine API [57], we query these identified IP addresses and URLs for granular host and domain details. Note that we use the 'datetime' query to narrow our results on those matching the first submission date found in the VirusTotal file report. This approach serves two benefits. First, it helps us focus on a limited set of results, as some domains can generate a large number of certificate and host results. Second, by targeting the period when the sample was most likely to be active, we ensure that the results are more relevant to the threat group's activity. Moreover, we exclude the top 500 domains (e.g., wordpress.com, europa.eu, drive.google.com) and a list of reserved IP blocks (e.g., 0.0.0.0/8, 127.0.0.0/8) from our search to ensure that we focus solely on identifying domains and IPs specific to threat groups. Following this process, we were able to extract infrastructure features for 2,345 samples (38.22%) in our dataset. Although these features were not available for all samples, we included them in our clustering process as they complement the pattern-based features. A complete list of the pattern-based and infrastructure features is available as part of our artifact.

To transform the raw *linking* features, we use a sentence transformer to generate text embeddings for clustering. Specifically, we use the pre-trained sentence transformer model, trained on a large dataset of 215 million (question, answer) pairs from various sources [210]. This semantic search model encodes textual features into a compact vector space, allowing it to capture subtle variations in complex patterns. These generated embeddings are then compared, for instance, using cosine similarity, to identify sentences with similar meanings. To illustrate this process, consider four samples, each containing the URLs: `http://a0711854.xsph.ru`, `http://a0713099.xsph.ru`, `http://a0714424.xsph.ru`, and `http://ca.mtin.es`, respectively. Employing the sentence transformer model, we generate encodings for these URLs and set a similarity threshold of 0.8. As a result, we identify the first three URLs as highly similar, leading us to assign these three samples to the same bucket, indicating their similarity based on the closely related URLs. The last sample remains in a separate bucket. We tried different pre-trained models [89, 90] and thresholds, and found that the best combination was the "multi-qa-MiniLM-L6-cos-v1" model with a 0.8 similarity score. We extend this approach to certificate issuer organizations (e.g., "Verisign," "DigiCert Inc."), email addresses, and Windows and Linux file paths.

Further, to eliminate overly common features, we analyzed their frequency of occurrence in the dataset. We perform feature selection on a subset of samples and determine thresholds by examining the distribution of feature frequencies in the validation set. Specifically, we identify a threshold of 0.75. To retain the most meaningful features, we calculate the percentage of samples in which each feature appears and remove those present in more than 75% of the samples. Common examples of such elements include generic cloud URLs, IP addresses, country codes, and common certificate names. This approach refines the dataset by highlighting rare and potentially more meaningful relationships between the samples.

3.7 Clustering Implementation

Due to the general lack of reliable ground truth labels for APT malware, accurately classifying samples is challenging, as demonstrated by our relabeling efforts (see Section 3.4). Therefore, unsupervised clustering becomes a feasible solution. We use agglomerative hierarchical clustering [229], a method frequently employed in malware clustering [20, 197, 198]. Hierarchical clustering's strength lies in its ability to identify clusters of arbitrary shapes, making it well-suited for capturing complex relationships within the APT domain. The agglomerative clustering recursively merges similar clusters, resulting in a dendrogram-like structure. Initially, each data point is treated as a separate cluster (referred to as a leaf), and the algorithm computes the distance between these individual clusters.

Our experiments showed that the agglomerative approach can be computationally expensive for large feature spaces due to its recursive nature. To address this, we incorporate autoencoders to learn a latent representation of our features. Autoencoders offer the ability to learn compact representations of input data by capturing the underlying structure and removing noise or irrelevant information [237]. This compressed representation allows the clustering algorithm to converge faster. The autoencoder architecture employed in our approach consists of four layers: an input layer, two hidden layers, and an output layer. The input layer has the same number of units as the transformed feature set, while the hidden layers comprise 32 and 16 units, respectively. To activate the hidden layers, we employ the rectified linear unit (ReLU) activation function, due to its computational efficiency [88]. The output layer mirrors the input layer in terms of the number of units and employs the sigmoid activation function.

For optimal clustering results, we perform cluster validity analysis, using the metric Sum of Squared Errors (SSE) [196]. We iteratively merge clusters by selecting the below configuration that results in the lowest change in SSE:

- *Distance Metrics*: We consider the Euclidean and Manhattan distances to measure the dissimilarity between clusters.
- *Compute Full Tree*: We set the parameter 'compute_full_tree' to True in order to compute the complete hierarchical tree without early pruning.
- *Linkages*: We iterate over different linkage algorithms, including ward, complete, and average, to determine the optimal method for merging clusters.
- *Number of Clusters*: We explore various numbers of clusters to find the most suitable configuration.

Further, to identify the optimal number of clusters, we use the elbow method. This method involves running the clustering algorithm for different values of clusters (k) and plotting the SSE for each run. The elbow point, where the change in the SSE starts to diminish significantly, indicates the optimal k for our clustering tasks. We also experimented with other clustering algorithms, such as HDBSCAN and K-means, achieving comparable results.

We employ the *ADAPT Intra-Clustering* for APT campaign attribution (see Section 3.5 and top of Figure 3.2) for our executable and document samples. This approach groups samples within dominant file domains based on their shared similarities. Subsequently, the *ADAPT Inter-Clustering* (see Section 3.6 and bottom of Figure 3.2) for APT group attribution groups samples across all files incorporating the transformed linking features (PAT & INF) extracted from string content. It is worth noting, that our approach incorporates all file types from the dataset (see Table 3.2). Although some files may not yield readily identifiable patterns through static features, their inclusion allows for a more holistic analysis, and the potential discovery of subtle connections. However, this inclusivity can lead to singleton clusters or misclustering. To mitigate this, we employ an analyst-defined threshold for excluding samples with insufficient features, ensuring focus on the most informative samples for accurate threat group identification.

3.8 Evaluation and Results

In this section, we evaluate ADAPT's performance and effectiveness using the campaign-labeled dataset curated from MITRE information (discussed in Section 3.4.2). We report quantitative performance metrics for the attribution tasks in Section 3.8.1. We further explore the relative importance of different features for executables and documents in attributing threat campaigns in Section 3.8.2.

Experimental Setup. We implemented ADAPT using the Python programming language. We performed all experiments on a Windows 11 Pro machine with the following specifications: 12th Gen Intel(R) Core(TM) i9-12900KS, 3400 Mhz, 16 Core(s), 24 Logical Processor(s), and 32 GB RAM. To maintain a controlled research environment, the machine is connected to a dedicated router with no other devices connected to the network. Furthermore, to prevent interference from Windows Defender Antivirus, we place the dataset directory in an exclusion folder with real-time monitoring disabled [158]. The dataset comprises 6,134 samples with raw extracted features stored in JSON files, requiring 15.6 GB of storage.

3.8.1 Cluster Validity Analysis

Assessing the results of unsupervised clustering algorithms is challenging due to the lack of ground truth labels. There is no standard method for validating the output of clustering results [97]. Clustering algorithms aim to group similar objects together based on metrics such as distance between and within clusters. However, a key challenge that arises is determining the optimal number of clusters. As mentioned in Section 3.7, we determine the optimal number of clusters by minimizing the Sum of Squared Errors (SSE). SSE is calculated by summing the squared differences between the data points in a cluster and the cluster's centroid. A lower SSE indicates a more compact and dense cluster, implying that the objects within the cluster are closely related [244]. SSE measures the error or deviation within clusters based on the internal structure of the data, without using external ground truth labels. However, it is also helpful to analyze the clustering results by comparing them to existing ground truth labels [20].

Precision and Recall. We use the campaign-labeled dataset derived from MITRE information (discussed in Section 3.4.2) to validate our clustering results. This dataset, allows us to measure the level of agreement between the clusters we obtain and the information associated with the clustered samples. Bayer et al. [20] proposed to use precision and recall by establishing a mapping between the outcomes of their system-level behavioral clustering system and the *reference* clustering. In a similar vein, Perdisci et al. [197] presented an alternative method to evaluate the credibility of clustering results, concentrating on assessing the cohesion within individual clusters and the separation between different clusters. Based on these previous studies, we propose a method to normalize labels between clustering results and the reference cluster.

Given a dataset of samples $S = \{S_1, S_2, S_3, ..., S_n\}$, comprising n samples identified by their SHA256 hash, and a set of t distinct ground truth/reference clusters $T = \{T_1, T_2, T_3, ..., T_t\}$, we apply a clustering algorithm to the dataset S resulting in a set of predicted clusters $C = \{C_1, C_2, C_3, ..., C_c\}$. Here, c represents the total number of distinct predicted clusters, and each predicted cluster C_i contains an arbitrary number of samples and is assigned a cluster label i.

To normalize cluster labels, we take the dataset S, the predicted clusters C, and the ground truth clusters T as inputs. For each predicted cluster C_i within C, we retrieve the subset of samples S_i from the dataset S that belong to that specific predicted cluster C_i. Subsequently, we count the occurrences of each ground truth cluster label t within S_i and identify the label with the highest count, denoted as t_{majority}. We assign t_{majority} as the normalized cluster label N_i for the predicted cluster C_i. We repeat the normalization process for all predicted clusters in C, generating the collection of normalized cluster labels $N = N_1, N_2, N_3, ..., N_c$. Using normalized cluster labels N and the ground truth labels T, we define:

True Positives (TP): The number of samples that are correctly assigned to the same cluster both in the normalized cluster labels N and the ground truth labels T.

$$TP = \sum_{i=1}^{c} \sum_{j=1}^{t} I(N_i = T_j) \cdot I(C_i \cap T_j \neq \emptyset)$$

False Negatives (FN): The number of samples that are incorrectly assigned to a different cluster in the normalized cluster labels N but belong to the same cluster in the ground truth labels T.

$$FN = \sum_{i=1}^{c} \sum_{j=1}^{t} I(N_i \neq T_j) \cdot I(C_i \cap T_j \neq \emptyset)$$

False Positives (FP): The number of samples that are incorrectly assigned to the same cluster in the normalized cluster labels N but belong to a different cluster in the ground truth labels T.

$$FP = \sum_{i=1}^{c} \sum_{j=1}^{t} I(N_i = T_j) \cdot I(C_i \cap T_j = \emptyset)$$

Here $I(x)$ denotes an indicator function that takes an argument x and returns 1 if x is true, and 0 otherwise.

Clustering Results. Using the above-defined metrics, we calculate precision, recall, and F1-score to evaluate the performance of our campaign and group attribution tasks (see Section 3.5 and 3.6) on the reference dataset. Further, we report the clustering metrics, Silhouette coefficient (SC) [230], and the SSE. SC, a common metric for unsupervised tasks, ranges from -1 to 1. It measures the average distance between a sample and its assigned cluster compared to the distance to the next nearest cluster. Values near 0 indicate clusters with some overlap, while negative values suggest samples being placed in the wrong cluster. Our goal is to identify the number of clusters that minimize SSE and have a positive SC.

Among the executable samples, we achieve a precision of 0.93, a recall of 0.92, and a combined F1-score of 0.91. The SSE is 1.45 and the SC is 0.50. Document samples demonstrate higher precision (0.95) and recall (0.94), resulting in an F1-score of 0.92. The SSE for documents is 0.72 with an SC of 0.36. We hypothesize that the clustering performance of documents is better than that of executables because they tend to

Table 3.3: List of feature categories for *executables* and their performance on the campaign-labeled reference dataset. We evaluate combinations of *Executable Specific Features* (EXF) and *Generic Features*, including Capabilities (CAP) and Strings (STR).

Feature Category	**# Features**	**Precision**	**Recall**	**F1-score**
EXF	22,042	0.85	0.72	0.70
EXF + CAP	22,099	0.88	0.87	0.85
EXF + STR	77,697	0.91	0.90	0.89
EXF + CAP + STR	77,759	0.93	0.92	0.91

Table 3.4: List of feature categories for *documents* and their performance on the campaign-labeled reference dataset. We evaluate combinations of *Document Specific Features* (DCF) and *Generic Features*, including Capabilities (CAP) and Strings (STR).

Feature Category	**# Features**	**Precision**	**Recall**	**F1-score**
DCF	85	0.88	0.84	0.79
DCF + CAP	142	0.93	0.93	0.92
DCF + STR	44,035	0.92	0.91	0.91
DCF + CAP + STR	44,097	0.95	0.94	0.92

exhibit consistent, unique patterns across campaigns. In contrast, executables often use obfuscation techniques and display polymorphic behavior, making it difficult to identify distinguishing features for clustering. However, the 93% precision for executables demonstrates that ADAPT was able to effectively distinguish between samples from different campaigns. For the threat group attribution task, we achieve a precision of 0.92, a recall of 0.89, and an F1-score of 0.89. While this F1-score reflects relatively good performance, the SC of 0.41 obtained with the lowest SSE of 2.70 highlights the challenges of accurately correlating diverse samples across executable and document domains. Further, we report the number of clusters identified in our reference clustering consisting of 230 samples with 22 campaigns and 17 threat groups. For the threat campaign attribution task involving executables, we identified 18 clusters with a precision of 93% and a recall of 92%, while for documents, we identified 9 clusters with a precision of 95% and a recall of 94%. For the group attribution task, we identified 15 clusters with a precision of 92% and a recall of 89%.

3.8.2 Feature Importance

In the context of unsupervised clustering algorithms, we assess the suitability of different features by analyzing their impact on precision and recall in our reference dataset. Table 3.3 shows the importance of different feature categories for clustering executable samples. The importance of a specific feature is determined by the degree to which it increases the overall F1-score. From the table, we can see that by solely incorporating the executable-related features (EXF) like exported functions, we observe a relatively low F1-score of 0.70. However, combining these with the detected capabilities (EXF +

CAP) and string features (EXF + STR) significantly improves the F1-score to 0.85 and 0.89 respectively.

Similarly, Table 3.4 shows the importance of different features for accurate clustering of document samples. Notably, relying solely on document-related features (DCF), like suspicious keywords, yields a relatively low F1-score of 0.79. However, incrementally incorporating modular capabilities (CAP) and strings (STR) into the feature set results in an F1-score improvement, closely approaching the highest F1-score of 0.92 achieved when combining all available features. Our findings suggest that for executables, features such as capabilities, and string artifacts can serve as strong indicators of similarity or dissimilarity between samples. Notably, incorporating string features significantly boosted the F1-score for executable clustering (from 70% to 89%), whereas the impact on document clustering was less pronounced (from 79% to 91%). This is likely because printable strings in executables are more informative than hierarchically structured formats of documents and PDFs. Šrndic et al. [241] also highlight this observation in their work on PDF and Adobe Flash files.

3.9 Qualitative Case Studies

In this section, we evaluate ADAPT's clustering using the group-labeled dataset (discussed in Section 3.4.1). We focus on two practical use cases for analysts: (1) We discuss the results of ADAPT's campaign and group attribution for randomly selected clusters from Gamaredon, APT29, and Lazarus, and investigate reasons for misclustering (see Section 3.9.1). (2) We discuss ADAPT's attribution of unlabeled samples to known threat groups (see Section 3.9.2).

3.9.1 Attributing Labeled Samples

Insights on Threat Campaigns: Gamaredon's 2017 & 2022 Campaigns. Table 3.5 (in the Appendix) shows distinct threat campaigns perpetrated by Gamaredon, a suspected Russian cyber espionage threat group [171]. The campaigns occurred in 2017 and 2022, and using ADAPT Intra-Clustering, we successfully grouped the samples belonging to these campaigns into the Clusters #C1 and #C2, respectively.

Unit42 reported the Cluster #C1 samples as a part of the 2017 campaign, which primarily targeted individuals involved in the Ukrainian military and national security establishment [107]. The threat actors distributed custom-developed Windows malware that could download additional payloads, which were distributed as password-protected self-extracting Zip-archive (.SFX) files. These SFX files wrote a batch script to disk and installed a remote access trojan. ADAPT clustered the samples together based on the following shared similarities: the privilege escalation capability via `AdjustTokenPrivileges`, the unique linker `MSVC_2008_linker`, a distinct string formatting pattern used in naming temporary files (%s.%d.tmp), and the use of RIPEMD160 for file encryption.

Unit42 reported the Cluster #C2 samples in January 2022 after observing Gamaredon's attempt to compromise a Western government entity in Ukraine following the Russia-Ukraine conflict [257]. This campaign uses a custom Windows Remote Access Trojan (RAT) with anti-detection features to evade antiviruses and sandbox environments. Further, it is capable of downloading and executing files, capturing screenshots, and running arbitrary commands on compromised systems. Although the samples in this campaign evolved over time, ADAPT successfully found unique elements that persisted through changes, such as the use of a shared linker and compiler (`MSVC_2005_linker` and `msvc_uv_55`) and the use of `GetKeyState` for keylogging. Additionally, the unique batch script pattern `7ZSfx%03x.cmd` and the embedded icons with Russian-language naming convention (`ru-ru`) helped ADAPT to cluster the samples.

Takeaway 1: ADAPT's feature extraction improves basic static analysis for threat analysts, giving quick insights into related samples. Additionally, ADAPT's Intra-Clustering model can identify samples from different campaigns, helping analysts build and track adversary profiles.

Insights on Threat Groups: APT29 & Lazarus. Table 3.6 (in the Appendix) shows a subset of sample hashes associated with APT29. These hashes were exposed in a July 2020 advisory report, revealing the WellMess and WellMail custom malware [182]. This malware targeted COVID-19 vaccine developers in Canada, the United States, and the United Kingdom to steal vaccine-related data. Interestingly, Japan's Computer Emergency Response Team (CERT) observed the deployment of WellMess within a Japanese organization as early as 2018 [251], indicating the malware's presence in various campaigns targeting different organizations. Initially, unattributed, further investigations by the NSA and NSCS linked WellMess and WellMail to APT29, Russia's Foreign Intelligence Service, which is also suspected of orchestrating the SolarWinds campaign [212].

Using group-based linking characteristics to attribute samples across multiple file types, ADAPT successfully clustered these hashes belonging to the same threat group in Cluster #G1, encompassing WellMess and WellMail campaigns dating back to 2017. The samples include both ELF and PE executables, with some of these binaries programmed in Go for cross-compatibility. Inspecting the shared features that caused ADAPT to group the samples together revealed that all the samples used similar Golang module file paths, such as `/home/ubuntu/GoProject/src/bot/botlib.Work`, `/usr/local/go/src/net/fd_mutex.go`, and `/golang_org/x/crypto/curve25519.freeze`. This suggests the use of mutexes and the curve25519 crypto library. Additionally, we identified similar MD5 hashes and matching regex patterns embedded in the string content of the samples, including `1DecemberDuployanDurationEthiopic` and `15625AdjustTokenPrivilegesAlaskan`.

In another example, Table 3.7 (in the Appendix) shows two MachO samples associated with the North Korean threat group, Lazarus, namely JMTTrade and CelasTradePro. These malware samples, disguised as legitimate cryptocurrency trading applications, have been used in separate campaigns by Lazarus to target both Windows and Mac systems since at least 2018 [42]. ADAPT effectively grouped these samples together in Cluster #G2 due to a recurring Bitcoin pattern embedded within their string content. Notably, the associated Bitcoin wallet addresses remain active and have received a total of $179,442 in funds [24]. Furthermore, inspecting the common linking feature set we identified similar URLs and shared email addresses, such as `knzg75@jmttrading.org` and `altancan73@jmttrading.org`, that facilitated the clustering of these samples. Additionally, the samples share a distinct certificate issuer organization, such as `WoTrus CA Limited`, as observed from the infrastructure feature set. In February 2021, the United States government charged three individuals in connection with this attack, which resulted in a $1.3 billion theft. These charges were based on infrastructure similarities and online accounts observed in prior campaigns [56].

These examples demonstrate the effectiveness of using pattern-based features, such as system file paths, unique file names, email addresses, Bitcoin identifiers, certificate authorities, and domain names or URLs, to identify threat actor indicators across campaigns. While these distinct signatures have traditionally been used by threat analysts in manual analysis, ADAPT streamlines and automates the process of extracting and clustering these key patterns.

Takeaway 2: ADAPT's Inter-Clustering model uses distinct patterns and infrastructural details to identify threat actor signatures. This capability expedites the process of connecting attacks carried out by the same threat actor.

Clustering Issues. Although effective in most cases, ADAPT's clustering might not be ideal in specific scenarios. For instance, while analyzing document clusters we identified an outlier: A document, linked to 'Operation Dream Job' (espionage using fake defense jobs [172]), was not grouped within that category. While this sample exhibited some differences compared to others in the campaign, a unique characteristic was the use of similar themes within the document content (e.g., job descriptions, industry jargon) and a Boeing image on the first page. This demonstrates that including image extraction in the document processing pipeline can improve the ability of ADAPT to cluster similar documents based on visual features.

For executables, obfuscation can hinder clustering. For example, the 'Andromeda' campaign [172] used known packers to obfuscate malware. Older samples (first observed in 2013) employed NSIS-based packers, while newer versions (observed in 2022) were unpacked. Our current clustering process grouped the packed samples together, but the unpacked version ended up in a separate cluster, highlighting the need for extracting obfuscated content for efficient clustering. We discuss the issue of packing and obfuscation, including a cursory study on how widespread these techniques are across our dataset, in Section 3.10.

In some cases, the performance of ADAPT's clustering for group attribution was limited by a lack of distinctive threat group characteristics. For instance, some samples from Gamaredon, WizardSpider, and APT29 were grouped together due to the absence of infrastructure-based indicators and they accessed common file paths (e.g., `C:\Windows\System32\cmd.exe` or `C:\Windows\System32\WindowsPowerShell\v1.0\powershell.exe`). As discussed in Section 3.6, we successfully extracted Infrastructure (INF) features for 38.22% of the samples. Consequently, the clustering relied on Pattern-based features (PAT), which may not be sufficient for accurate group attribution. However, as detailed in Section 3.7, these cases can be identified through feature analysis and can be avoided using threshold-based clustering techniques.

Finally, we also recognize limitations in ADAPT's clustering, particularly when APT groups use publicly available offensive security tools, such as Cobalt Strike, and employ shared infrastructure. This can result in campaigns from unrelated groups being clustered together. Possible remediations include disassembled code analysis to identify code-level differences, and examining tool-related characteristics such as Cobalt Strike license numbers. Another avenue for improvement is incorporating features from dynamic and behavioral analysis, yet with the caveat of dealing with environment-sensitive samples [131].

3.9.2 Attributing Unlabeled Samples

ADAPT's group attribution identified nine out of the 44 unlabeled samples in the group-labeled dataset (see Section 3.4.1) as belonging to specific APT groups. To verify these attributions, we compared the samples to all publicly available information, including community comments on VirusTotal. Four of the samples had comments linking them to the APT group that ADAPT clustered them in, even though the samples were not initially labeled with those groups because there were no public reports of attribution. These samples were clustered based on shared URLs, IPs, similar country codes, and ASNs. Additionally, one of the samples was clustered with three other samples from APT40 (Chinese state-sponsored cyber espionage group) based on similar BGP prefixes and ASNs. The only public information about the sample's attribution was that it was linked to a Chinese state actor. Another sample was clustered with four other samples from APT10, but the community information linked it to APT3, which are both Chinese threat actors. Finally, three samples were clustered with known APT groups that had never been publicly attributed to any threat actor. The first sample was grouped with another APT3 sample due to a Bitcoin address pattern. The second sample was linked with six APT28 samples based on string patterns. The third sample was linked to two Kimsuky document samples because of distinct file paths and shared Korean domains in the URLs.

We acknowledge that other organizations might have information about these samples and can validate our attributions. To help the community further analyze and verify our findings, we list ADAPT's clustering results and the samples' hashes in Table 3.8 (in the Appendix), as well as release their features and relevant information as part of our artifact.

Table 3.5: Gamaredon threat campaign analysis. (see case study in Section 3.9.1)

Campaign	APT Hashes	Label
Gamaredon 2017	2c5d55619d2f56dc5824a4845334e7804d6d306daac1c23bec6f078f30f1c825	Cluster #C1
	3ef3a06605b462ea31b821eb76b1ea0fdf664e17d010c1d5e57284632f339d4b	Cluster #C1
	4d1a6fe0df9b00f34e3461cb0119224b242c0257b991e8c44a51f0e3304771ea	Cluster #C1
	63fcfab8e9b97d9aec3d6f243003ea3e2bf955523f08e6f1c0d1e28c839ee3d5	Cluster #C1
Gamaredon 2022	61e67302a85ff98eabc589572dbf3bf6e1012207d399b9f2b6b38527833e9198	Cluster #C2
	b9dd1e5ec018090b404dd7550d4423ff38ee1f016a5ab214f128544f5b399759	Cluster #C2
	cbe1dbd167bccbf61ee8608092a767ce3fbfb5fe5f6e959848d9a8d9091402fb	Cluster #C2
	3dca96ef38d4b8d1dbb4afed43a22ace93cc3a0a105120d4cf637e6dafe129e9	Cluster #C2

Table 3.6: APT29 threat group analysis. (see case study in Section 3.9.1)

Normalized Group Label	APT Hashes	Group Cluster Label
APT29	84b846a42d94431520d3d2d14262f3d3a5d96762e56b0ae471b853d1603ca403	Cluster #G1
	00654dd07721e7551641f90cba832e98c0acb030e2848e5efc0e1752c067ec07	Cluster #G1
	0322c4c2d511f73ab55bf3f43b1b0f152188d7146cc67ff497ad275d9dd1c20f	Cluster #G1
	5ca4a9f6553fea64ad2c724bf71d0fac2b372f9e7ce2200814c98aac647172fb	Cluster #G1
	bec1981e422c1e01c14511d384a33c9bcc66456c1274bbbac073da825a3f537d	Cluster #G1

Table 3.7: Lazarus threat group analysis. (see case study in Section 3.9.1)

Normalized Group Label	APT Hashes	Group Cluster Label
Lazarus	7ea6391c11077a0f2633104193ec08617eb6321a32ac30c641f1650c35eed0ea	Cluster #G2
	c0c2239138b9bc659b5bddd8f49fa3f3074b65df8f3a2f639f7c632d2306af70	Cluster #G2

Table 3.8: Attribution of unlabeled samples. (see discussion in Section 3.9.2)

Potential Group Label	APT Hashes	Group Cluster Label
APT3	71b201a5a7dfdbe91c0a7783f845b71d066c62014b944f488de5aec6272f907c	Cluster #G3
Transparent Tribe	bff6270b7c6240c394515dc2505bb9f55d7b9df700be1777a8469143f78d0eb6	Cluster #G4
APT40	f659b269fbe4128588f7a2fa4d6022cc74e508d28eee05c5aff26cc23b7bd1a5	Cluster #G5
APT28	4a9efdfa479c8092fefee182eb7d285de23340e29e6966f1a7302a76503799a2	Cluster #G6
	eae62bb4110bcd00e9d1bcaba9000defcda3d1ab832fa2634d928559d066cb15	Cluster #G7
	b3cee881b2f9d115c98d431b70a75709aade2317a82a0792c15dce2ffa892679	Cluster #G7
APT15	12e1b00af73101cb297387b6ee5035c4cae04211d995ddd233fb375deb492b0a	Cluster #G8
Kimsuky	fa71eee906a7849ba3f4bab74edb577bd1f1f8397ca428591b4a9872ce1f1e9b	Cluster #G9
APT10	df5f1b802d553cddd3b99d1901a87d0d1f42431b366cfb0ed25f465285e38d27	Cluster #GA

3.10 Discussion and Future Work

Long-lived APT campaigns pose a significant threat due to their stealthy nature. Traditional methods struggle, requiring the tracking of numerous, chronologically linked events over extended periods [4, 30]. Our approach, ADAPT, bypasses this limitation by analyzing inherent features within suspicious or malicious files, independent of their execution sequence. As detailed in our background and motivation (see Section 3.2), APT groups employ variations in attack vectors and sample modifications throughout a campaign. By identifying shared techniques and capabilities from these heterogeneous artifacts, ADAPT helps analysts rapidly attribute the attack, prioritize analysis, and streamline investigations – even when the attack unfolds asynchronously. Still, we acknowledge open challenges and avenues for future work:

Packing and Obfuscation. Following the discussion in Section 3.9 we explored the application of common obfuscation techniques on our group-labeled dataset comprising 6,134 samples. Based on previous experiments and recognizing the limitations of existing packer detection tools [2, 151], we used a combination of Manalyzer, Detect it Easy, and Yara rules. These tools collectively flagged 222 samples (3.61%) from the entire dataset, indicating the presence of common obfuscation techniques in those samples.

Content Extraction from Heterogeneous Files. The goal of ADAPT is to use static features for efficient clustering of diverse files. To improve the attribution results, we aim to develop techniques for identifying and extracting obfuscated and embedded content (e.g., PDFs) from executables. Additionally, future efforts will focus on incorporating robust malicious content extraction from heterogeneous file formats. Our orthogonal study on malicious documents [226] highlighted the complexity of non-binary files, particularly Microsoft Office documents (Word, Excel, PowerPoint) and RTFs, and their widespread use among sophisticated attackers. Our study revealed limitations in current document analysis approaches, such as the inability to correctly identify file formats and manage emerging file types like OneNote.

Adversarial Manipulation and Evasion. We acknowledge that certain features within ADAPT may be susceptible to manipulation by adversaries, potentially enabling them to evade accurate attribution. However, it is important to note that, to the best of our knowledge, ADAPT is the first system that explores the feasibility of performing both campaign and group attribution using lightweight features from diverse threat sources. Our approach leverages domain expertise to identify features useful for attribution. Following this initial selection, as a part of future work, we aim to rigorously evaluate the robustness of these features against adversarial manipulation, particularly the use of "false flags" [19]. False flags differ from traditional evasion techniques that aim to bypass detection (e.g., benign-appearing actions) or hinder analysis (e.g., anti-analysis tricks). Instead, false flags allow attackers to mask their true identity and deflect attribution towards another nation state. This is the closest adversarial manipulation technique seen in the wild among APTs. To this extent, we identified samples known to exhibit code reuse across multiple threat actors. In particular, we analyzed a set of malware

Table 3.9: Overview of Prior Studies on APT Detection and Attribution. Our approach is the only one that handles executables (</>) and documents (📄) and performs both threat group and campaign attribution.

Approach	**Dataset**	**Samples**	**Groups**	**Campaigns**	**Artifacts**
APT detection using SIEM/EDR alerts	Synthetic data [72]	–		●	
	Third-party Enterprise Dataset [223]	–	●	●	
APT detection through data provenance	DARPA TC and enterprise logs [80, 100, 125, 137]	–		●	●
APT attribution with knowledge graphs	1,041 OSCTI reports [211]	–	●		
APT attribution based on malware samples	1,569 samples, 16 APT groups [263]	</>	●		
	864 samples, 5 APT groups [79]	</>	●		
	3,200 samples, 2 APT groups [217]	</>	●		
	287 samples, 7 APT campaigns [169]	</>		●	◐
Our Approach: ADAPT	**6,134 samples, 92 APT groups** **230 samples, 17 APT groups & 22 APT campaigns**	</>, 📄	●	●	●

samples exhibiting code reuse across disparate threat actors [246]. These samples were initially misattributed by analysts to a single threat actor due to shared VBA macro code (embedded code in documents). Using ADAPT, we were able to correctly cluster the samples into distinct groups. This differentiation was achieved by analyzing unique identifiers within the samples, such as file paths for destination folders and scheduled task creation methods present in a specific subset of samples.

Concept Drift. Concept drift refers to the phenomenon where the underlying distribution of data changes over time. In our context, this could manifest as new APT campaigns are emerging, potentially leading to changes in how samples are grouped during clustering. Future work will focus on investigating concept drift's impact on ADAPT's *unsupervised* setting using rigorous experiments. This would necessitate using a timestamped dataset to track how the underlying data distribution evolves. Specifically, we are interested in observing if new binaries consistently fall within existing clusters or form entirely new clusters as the attack landscape changes.

Representative Dataset. The absence of ground truth datasets that encompass multiple threat groups and include heterogeneous file types remains a significant open research challenge. Furthermore, we did not encounter any dataset with threat campaign labels, prompting us to curate our own dataset (see Section 3.4), which we provide as part of our artifact. Still, the scale of our dataset is limited, with, on the one hand, APT samples inherently being less widespread than generic malware and the effort involved in manually (re-)labeling the samples.

Furthermore, Arp et al. [15] showed that data sampling biases and data snooping can invalidate the results of machine learning models in security applications. We mitigate these risks by using real-world malicious files from trusted sources instead of synthetic datasets. However, sampling bias is still possible due to the over-representation of certain groups and file types. Additionally, data snooping is less straightforward in unsupervised clustering than in supervised learning, so we exercise caution in data handling. We use the reference dataset only to evaluate clustering results based on the chosen algorithm and generalized parameters developed during the clustering phase.

3.11 Related Work

Table 3.9 shows a summary of the most closely related prior work on APTs. In the following, we discuss related APT datasets, and prior approaches on APT detection and attribution.

APT Datasets. Gray et al. [75] developed a promising APT dataset comprising 17,513 APT samples belonging to 275 APT groups by mining threat reports. However, their dataset only included executable samples (PE, and ELF files) and lacked other file types, particularly documents. Similarly, Laurenza et al. [120] curated a dataset comprising exclusively binary samples, whereas the dataset from cyber-research [49] contains APT samples belonging to only 12 APT groups. Our dataset includes a wide range of file types from 92 APT groups, overcoming the limitations of previous datasets.

APT Detection. Existing research in APT detection leverages alert correlation to identify anomalous behaviors or APT footprints. Ghafir et al. [72] proposed MLAPT, a machine learning system for APT detection using network traffic data, while Sachinananda et al. [223] focus on correlating security alerts from various sources, such as Intrusion Detection and Prevention Systems (IDS/IPS), Endpoint Detection and Response (EDR), and Security Information and Event Management (SIEM) to cluster alerts associated with the same APT attack scenario. Provenance graphs have also emerged as a state-of-the-art approach in APT detection. ANUBIS [10] leverages provenance graphs to capture causality and detect APTs. Similarly, APTHunter [137], Unicorn [80], NODLINK [125], and MAGIC [100] focus on provenance-based anomaly detection using audit logs. These approaches, however, require raw log access, suffer from dependency explosion issues, and present challenges in reconstructing complex APT attack causality [87]. Existing research focuses on APT detection and hunting, leaving a gap for a comprehensive framework in APT sample correlation and attribution. We address this distinct yet complementary aspect. Our work leverages malicious artifacts to facilitate correlation, aiding investigations even without a complete understanding of the attack causality chain.

APT Attribution. Marquis-Boire et al. [152] manually extracted static features specific to APT malware, such as C&C infrastructure, string constants, and data exfiltration methods, to link executables from the same authors. Rosenberg et al. [217] propose deep learning for APT group attribution using sandbox analysis reports of PE binaries. While they perform classification between Chinese and Russian APT groups, specific groups remain undisclosed. Wang et al. [263] explore string and code features with random forest and DNN classifiers for APT malware attribution across 16 APT groups (1,569 samples). Han et al. [79] use dynamic API sequences for APT malware detection and group identification using 864 APT samples. However, limited public availability of the system and dataset, along with potential ground truth issues, hinder the broader evaluation of these approaches. Additionally, Mirzaei et al. [169] propose Scrutinizer, a system for detecting code reuse in PE malware binaries via function-level decompiled code similarity analysis. Through manual verification of samples, they identified 12 previously unknown APT-linked samples. However, Scrutinizer's reliance on a custom sandbox environment for intermediate results limits reproducibility and an unlabeled dataset of hashes hinders rigorous benchmarking. Finally, Ren et al. [211] propose a cybersecurity knowledge graph model for APT group attribution leveraging OSCTI information.

While the problem of attribution for non-APT malware has been studied extensively [8, 34, 75, 121, 198, 218], these related approaches focus on identifying the author of a binary file and extracting their stylistic features. Another line of related work on commodity malware focuses on so-called lineage [81, 99, 130], i.e., identifying the evolution of and relationships between malware families and variants. In contrast, our research looks at attribution more broadly, and extracts features based on the tactics and techniques of the APT group responsible for the attack and the campaign they are executing. ADAPT advances the state-of-the-art in APT malware attribution by addressing its unique challenges. First, we establish a comprehensive understanding of the APT landscape and its complexities through practical case studies. This includes recognizing the distinct characteristics of APTs, such as low-and-slow tactics and multi-stage attacks involving heterogeneous artifacts. Existing studies primarily focus on PE binaries, while ADAPT performs attribution for executables and the most common initial attack vectors [241], including document file formats. Finally, existing solutions either perform campaign-level or group-level attribution. ADAPT automates both, allowing for a more systematic attribution process. Furthermore, to encourage future research and facilitate reproducibility, we open-source both our dataset, and source code. We aim for transparency and allow researchers to benchmark their approaches against our results.

3.12 Conclusion

Unlike conventional malware threats, APTs are technically sophisticated adversaries conducting well-organized, stealthy, and repeated campaigns targeting a wide range of organizations. In this thesis, we introduce ADAPT, an automated attribution approach that provides insight into the adversary's tactics and identity by performing APT campaign and group clustering. ADAPT's two-tiered approach offers a solution to the challenges of human attribution in the face of evolving and strategic threat campaigns, the use of different file types in attacks, and the collaboration among threat groups that complicates the attribution process. ADAPT clusters executable and document samples by analyzing malicious characteristics. It also uses linking features to identify group traits and signatures, helping connect samples from different campaigns to the same threat group. Practical case studies on real-world APTs and the association of unattributed samples to known APT groups demonstrate how ADAPT simplifies the attribution process, empowering security practitioners with automated tools to assess and compare attribution claims. We envision our work involving the collection of campaign- and group-labeled APT datasets, automated feature extraction for diverse file types, and the use of clustering techniques for attribution analysis as a source of inspiration for future research.

CHAPTER 4

Exploring the Malicious Document Threat Landscape: Towards a Systematic Approach to Detection and Analysis

Despite being the most common initial attack vector, document-based malware delivery remains understudied compared to research on malicious executables. This limits our understanding of how attackers leverage document file formats and exploit their functionalities for malicious purposes. In this thesis, we perform a measurement study that leverages existing tools and techniques to detect, extract, and analyze malicious Office documents. We collect a substantial dataset of 9,086 malicious samples and reveal a critical gap in the understanding of how attackers utilize these documents. Our in-depth analysis highlights emerging tactics used in both targeted and large-scale cyberattacks while identifying weaknesses in common document analysis methods. Through a combination of analysis techniques, we gain crucial insights valuable for forensic analysts to assess suspicious files, pinpoint infection origins, and ultimately contribute to the development of more robust detection models. We make our dataset and source code available to the academic community to foster further research in this area.

4.1 Introduction

Documents are a widely used method to deliver malicious payloads during a cyberattack: In 2016, the Microsoft Defender Security Research Team reported that 98% of Office-targeted attacks utilized malicious macros [159]. This dominance of macro-based threats was further corroborated by a recent ReasonLabs cybersecurity report, which identified them among the top 10 threats detected in 2022 [207]. Moreover, Microsoft's disclosure of 59 vulnerabilities, including zero-day exploits, in Word documents during 2023 highlights the criticality of analyzing these formats [76]. Despite the significant threat posed by malicious documents, our understanding of these threats remains limited [122, 167, 222].

Document macros are embedded sequences of commands within a document, similar to the instruction sets found in executable programs [160]. These macros can be weaponized to download malware, exfiltrate sensitive data, or exploit vulnerabilities in the processing software to achieve unauthorized system access. Furthermore, attackers often leverage social engineering tactics to manipulate users into interacting with malicious documents containing clickable links or attachments, or deceptive prompts encouraging users to enable macros [255].

Prior research has explored different approaches for detecting malicious documents. Yan et al. proposed DitDetector, which leverages bimodal machine learning models to combine visual and textual information for macro malware detection [270]. Cohen et al. presented a Structural Feature Extraction Methodology (SFEM) specifically targeted towards Office Open XML (OOXML) document formats, employing machine learning for malicious document identification [43]. A significant portion of document analysis research focuses on extracting and analyzing macro code. Extraction is typically achieved using tools like oletools [117], followed by training detection models. These are based on techniques like Latent Semantic Indexing (LSI) [167], Natural Language Processing (NLP) using Bag-of-Words and Term Frequency-Inverse Document Frequency (TF-IDF) [166], or identification of specific macro code keywords (e.g., AutoOpen and Shell) [111]. Beyond code analysis, recent work by Casino et al. explores the potential of detecting deceptive information within documents by constructing lightweight signatures from file components (e.g., "enable editing" and "enable content") for malware detection [37]. Ruaro et al. took a more targeted approach, focusing on symbolic execution for automated deobfuscation and analysis of Excel 4.0 macros (XL4) prevalent in Microsoft Excel files [222].

While existing research primarily focused on the binary classification of documents as either "malicious" or "benign," we argue that a comprehensive understanding of the evolving landscape of malicious documents is required for effective defense strategies. This is mainly because of two key factors: (1) *The diverse nature of file formats* (e.g., OLE and OOXML) *and macro types* (e.g., Visual Basic for Applications (VBA) macros [160] and Excel 4.0 macros [193]) presents challenges for extracting file metadata and macro code. This variety allows attackers to develop new variants utilizing different formats, effectively evading signature-based detection [271]. (2) *Attackers actively employ obfuscation techniques to evade analysis* and hinder the effectiveness of existing tools in accurately analyzing these samples [183, 184]. As a result, malware analysts are often limited by the capabilities of available tools. When these tools encounter incompatible file formats or obfuscated code, analysts resort to time-consuming and resource-intensive manual analysis.

The aforementioned challenges highlight the need for a deeper understanding of the malicious document ecosystem. Building upon prior efforts in malicious content detection, we conduct a measurement study on a large dataset of recent malware samples. This study focuses on prevalent threats embedded within Microsoft Office documents (Excel and Word) and Rich Text Format (RTF) files to inform future research efforts in this domain. We leverage state-of-the-art automated feature extraction techniques specifically designed for document-based malware analysis. Through a comprehensive evaluation, we assess our capability to parse complex document file formats, detect malicious indicators, and ultimately, identify gaps in current automated document analysis approaches.

In summary, our main contributions are as follows:

- We collect a dataset of malicious documents covering the period from January 2020 to January 2024 and encompassing 9,086 samples across various file formats (.docx, .xlsx, .doc, .xls, and .rtf).
- We present a methodology for robust file type identification, particularly crucial when dealing with a diverse set of 10+ file formats. This methodology enhances the accuracy of malicious indicator detection by ensuring proper file parsing.
- We identify the most prevalent document-based threats and pinpoint limitations in current analysis methods. We also provide valuable insights to inform the future development of robust analysis tools and defense mechanisms.

Artifacts. We believe that our findings will contribute to the advancement of knowledge in the malicious documents ecosystem by informing the development of more robust analysis mechanisms. To foster further research in this domain, we make our dataset and source code available at https://github.com/SecPriv/malwaredocumentanalysis.

4.2 Document File Types

Office File Format. Microsoft Office utilizes a variety of file formats for documents, each with its own capabilities. This variety can be seen within Excel file formats alone, ranging from the older binary formats (.xls, .xlsb) to the XML-based formats (.xlsx, .xlsm). Despite these differences, all Excel files share a core structure: a workbook containing one or more spreadsheets.

Microsoft introduced the Office Open XML (OOXML) format in 2006. This XML-based format, standardized as ECMA-376 [58] and later adopted as ISO/IEC 29500 [96] in 2016, has become the de facto standard for representing documents, spreadsheets, and presentations. OOXML leverages zip-compressed archives to store data. These archives contain multiple files and directories, with the Extensible Markup Language (XML) used to describe the actual document content and associated elements like images or stylesheets. OOXML supports a wide range of features, from spreadsheet formulas and form fields to integration with other XML formats like SVG or MathML. Additionally, features like digital signatures, document encryption, and macro support (in various languages like Basic, JavaScript, and Python) are possible within OOXML documents [180].

Prior to OOXML, Microsoft Office used legacy binary file formats, primarily the Compound File Binary (CFB) format, also known as OLE (Object Linking and Embedding), and the Compound Document File (CDF) format. Introduced with Office 97, this format uses file extensions like .doc (Word), .xls (Excel), and .ppt (PowerPoint) [128].

Analogous to a traditional file system, OLE files are composed of storage objects and streams. Storage objects act as containers, potentially holding additional storage objects or streams. Streams, on the other hand, represent the actual data content, such as text, images, or embedded objects within the file. Although legacy, they remain available as an alternative document-saving option for compatibility purposes.

Rich Text Format (RTF). Introduced by Microsoft in the 1980s, RTF (.rtf) provides a method for encoding formatted text and graphics for use across different applications. This format facilitates interoperability, enabling document exchange between Microsoft products and various word-processing software. Consequently, RTF files can be transferred between operating systems without compromising document formatting [66]. The widespread support of RTF by most word processors, text editors, and document viewers allows for easy sharing and distribution. Unlike the previously discussed formats, RTF files rely on unformatted text, control words, groups, backslashes, and delimiters for formatting. The control words, according to RTF specification, begin with a backslash (e.g., `\fonttbl`), with parameters enclosed in curly braces ({...}). The braces can contain multiple control words (forming a group), plain text, or even nested braces for more complex formatting.

4.2.1 Document-based Threats

Attackers embed macro malware in documents, such as Word, Excel and PowerPoint files, which can download additional payloads from remote servers, extract malicious code directly from the document, or steal data from the victim's machine. A targeted phishing campaign in September 2023 exemplifies this threat [63]: APT34, a suspected Iranian cyberespionage group, used a seemingly legitimate document titled "MyCv.doc" to deploy the Menorah.exe malware. This document contained hidden macros that downloaded and dropped a .NET malware executable. Phishing emails are a common method for attackers to deliver these malicious documents [146]. However, with increased security awareness and Microsoft's default macro-disabling policies, attackers now resort to deceptive social engineering techniques [255]. This involves using misleading images or text within the document to trick users into enabling editing or running macros. The prevalence of document-based threats is further substantiated by Botacin et al. [27], who observed a significant increase in CDFs used in regionalized and targeted malware attacks. Below, we discuss the specifics of macros and their utility in different file formats.

Macros. Macros, essentially embedded code within office documents, can create custom functions that automate specific actions when the document is opened. Visual Basic

Listing 4.1: Example of a malicious VBA macro payload: The macro runs automatically when the document is opened, extracts and decodes the "Comments" property, writes the content to a temporary file and executes it via the shell in a platform-specific subroutine for Windows or macOS.

```
Attribute VB_Name = "ThisDocument"
Attribute VB_Base = "1Normal.ThisDocument"
Attribute VB_GlobalNameSpace = False
Attribute VB_Creatable = False
Attribute VB_PredeclaredId = True
Attribute VB_Exposed = True
Attribute VB_TemplateDerived = True
Attribute VB_Customizable = True

Sub AutoOpen()
    On Error Resume Next
    Dim found_value As String

    For Each prop In ActiveDocument.BuiltInDocumentProperties
        If prop.Name = "Comments" Then
            found_value = Mid(prop.Value, 56)
            orig_val = Base64Decode(found_value)
            #If Mac Then
                ExecuteForOSX(orig_val)
            #Else
                ExecuteForWindows(orig_val)
            #End If
            Exit For
        End If
    Next
End Sub


Sub ExecuteForWindows(decodedString As String)
    Dim tempFilePath As String, fileNum As Integer
    tempFilePath = Environ("TEMP") & "\tempfile.bat"
    fileNum = FreeFile
    Open tempFilePath For Output As #fileNum
    Print #fileNum, decodedString
    Close #fileNum
    Shell "cmd.exe /c " & tempFilePath, vbHide
End Sub


Sub ExecuteForOSX(decodedString As String)
    ...
End Sub
```

for Applications (VBA) macros are code snippets written in a variant of Visual Basic, specifically designed for scripting within Microsoft Office applications like Word, Excel, and PowerPoint [160]. Listing 4.1 shows an example of a VBA macro that extracts malicious code from the document's properties, decodes it, and executes it based on the victim's operating system.

VBA macros reside within a VBA project structure, but their location depends on the document type and file format. In Legacy Binary Formats (1997–2003) VBA projects reside within an OLE storage named "Macros" at the root of the OLE file for Word documents, while Excel stores them in storage called "_VBA_PROJECT_CUR." PowerPoint integrates macros directly into the binary presentation structure, and not in a dedicated storage [115].

The introduction of OOXML formats (2007+), such as .docx, .xlsx, and .pptx, changed how VBA macros are handled. These formats cannot store macros directly by default due to security considerations. Only specific files with enabled macros, denoted by the 'm' at the extension end (e.g., .dotm, .docm, .xlsm, .pptm), can contain them.

When enabled, macros are stored in a separate binary OLE file named “vbaProject.bin” within the zip archive structure that forms the core of OOXML formats. This file maintains the same VBA project structure as legacy formats for consistency. However, the location of “vbaProject.bin” within the zip archive varies depending on the document type. For example, in Word documents, it is located at “word/vbaProject.bin”, while in Excel 2007 and later it is found at “xl/vbaProject.bin”. Similarly, PowerPoint stores it at “ppt/vbaProject.bin”. While “vbaProject.bin” is the standard name for the VBA macro storage file in OOXML formats, the standard itself allows for some flexibility. Developers can use custom file names for the macro storage, as long as the relationships are correctly defined within the XML files.

Similar to OOXML files, RTF specifications do not allow embedding VBA macros directly within an RTF file [124], however, there is an exception. RTFs can embed OLE objects with potentially malicious content. Specifically, OLE “Package" objects can store any file type, including executables and scripts. If a user double-clicks such an embedded object, the system will launch the file [114]. For instance, security researchers have documented cases where malicious actors included macro-enabled Excel sheets within RTF documents and tricked users into executing payloads [14].

The extraction and analysis of malicious content depend on understanding the file formats and identifying the embedded macro location. Olevba [115] can parse Office file formats and supports both OLE and OpenXML, detecting VBA macros and extracting their source code. Similarly, extracting embedded objects from RTF files requires parsing their nested control words. Tools like rtfobj from oletool [117] and RTFScan from OfficeMalScanner [25], along with antivirus engines, rely on this parsing capability to identify potential threats within RTF files.

Advanced Techniques. VBA macros have been a popular choice for malicious actors [160], but attackers are constantly exploring advanced techniques like XL4 macros [193] and remote template injection vulnerabilities [202, 209]. Microsoft Excel 4.0 introduced XL4 macros in 1992, a tool for automation through dedicated macro worksheets. This concept differed significantly from VBA macros, introduced one year later in Excel 5.0. Unlike VBA macros, XL4 macros cannot reside within regular worksheets. Instead, they are confined to specific Excel 4.0 macro sheets, where each cell holds a single formula defining the macro’s action. In newer file formats, XL4 macros are stored in separate XML files [193]. These macros are often hidden within apparently legitimate documents and under several layers of obfuscation, making them difficult to analyze [222].

Beyond hiding and obfuscating macros in documents, attackers leverage other file formats to deliver malware. RTF’s object-linking capabilities make it an attractive target for attackers who exploit the format’s control words for malicious purposes. By manipulating the `\template` control word, attackers can reference a URL on a malicious server instead of a legitimate template file. This bypasses the intended functionality and directly retrieves the malicious payload upon opening the file [202, 252]. Furthermore, RTF’s `\object` control words, with associated parameters and data, allow attackers to embed executable files like PE, VBS, and JS directly within the document [234, 271]. These embedded objects can then download and execute code, effectively transforming the RTF file into a downloader and launcher for malicious content. Identifying such linked objects requires examining the RTF document's specific control words, particularly `\template` and `\object`.

4.3 Automating Malicious Document Analysis

In this work, we analyze malicious documents used in a wide range of attacks, from common threats (e.g., Qakbot and Emotet [41]) to targeted campaigns by nation-state actors (e.g., Gamaredon and Lazarus [146]). We employ a comprehensive approach that involves building an extensive dataset and meticulously evaluating the effectiveness of state-of-the-art methods in extracting malicious content while identifying their limitations.

We automate the analysis of malicious documents by using a combination of features extracted through static analysis. This includes identifying macro functionalities, extracting relevant textual information, and detecting malicious techniques and code patterns using Yara signatures. We focus on widely used binary file formats (.doc, .xls, .ppt), OOXML formats (.docx, .xlsx, .pptx), and RTF (.rtf) documents based on their ubiquity, the prevalence of recent vulnerabilities discovered in them [48], and their frequent utilization in targeted attacks [235].

4.3.1 Dataset

Understanding and mitigating the threats posed by malicious documents requires analyzing a dataset of diverse and representative samples. We achieve this by leveraging VirusTotal academic API [260], allowing us to collect a large dataset of malicious files focusing on specific file types. Using VirusTotal has become a standard way and is part of the best practices for doing malware research [40]. We start by identifying malicious documents submitted to VirusTotal between January 2020 and January 2024 using the 'tag: documents' filter and focusing on files classified as malicious by at least 20 antivirus engines. This initial search yielded a vast dataset of 97,288 unique files (identified by SHA256 hashes).

To create a more representative sample size, we implemented a two-step process. We used Microsoft Defender's Threat Labelling Engine to analyze file metadata, where Microsoft Defender consistently provided the most reliable labels, as confirmed by a previous study [222]. This helped us identify prominent malicious tags, threat categories, and malware families. This analysis revealed a significant presence of specific malware families, including Emotet (49,119 files) and Laroux (8,002 files). To avoid the overrepresentation of dominant malware families like Emotet, we employed a random sampling approach. We selected samples from the top 15 identified families, ensuring a diverse representation within the final dataset. Table 4.1 details the distribution of samples across these families. By using random sampling, we narrowed down the initial dataset to 9,086 files on which

Table 4.1: Malicious documents across the top 15 families.

Malware Family	# of Samples	Percentage
Emotet	1,335	14.69%
Thus.G	831	9.14%
Laroux	654	7.20%
Obfuse	431	4.74%
EncDoc	423	4.66%
Mailcab	284	3.12%
Donoff	240	2.64%
Sadoca	236	2.60%
Marker	199	2.19%
LionWolf	163	1.79%
Woreflint	120	1.32%
Madeba	104	1.14%
Hancitor	93	1.02%
Xaler	91	1.00%
Leonem	74	0.81%
Total Dataset	9,086	100%

we perform our experiments. However, to facilitate further research and benefit the community, we are releasing the complete dataset consisting of 97,288 SHA256 hashes.

4.3.2 Malicious File Analysis

The exploration of the malicious document ecosystem remains largely understudied. This is partly due to the variety and complexity of Microsoft's document formats, characterized by intricate specifications [128]. Malware authors leverage these complexities, crafting documents that exploit vulnerabilities in static parsers while maintaining compatibility with parsers within Office applications [234].

To understand the prevalence of different document formats within our dataset, we employed a systematic categorization based on each file's Multipurpose Internet Mail Extensions (MIME) type. A standardized identifier, the MIME type specifies a file's format and key characteristics. Essentially, it acts as a digital fingerprint attached to a file, conveying crucial information to systems about its content. As discussed in Section 4.2, this information is useful for the proper handling, interpretation, and processing of a file. Table 4.2 details the over 20 MIME types and file formats identified in our dataset. A significant portion of the files (about 87%) fall into Word or Excel formats, including both OOXML and pre-OOXML versions.

Filetype Identification. Our analysis revealed limitations in current approaches to identifying the correct file MIME type. To investigate the prevalence of file misclassification

Table 4.2: List of MIME types identified within our dataset along with the number of samples detected for each mime type.

MIME Type	File Type	Magika	Libmagic	ExifTool
application/vnd.ms-excel	Excel Spreadsheet	3,100	2,812	2,602
application/msword	Word Document	2,884	3,148	2,937
application/vnd.openxml-formats-officedocument.wordprocessingml.document	Word OOXML	1,156	1,076	1,630
application/vnd.openxml-formats-officedocument.spreadsheetml.sheet	Excel OOXML	841	779	137
application/vnd.ms-outlook	Outlook Message	521	-	-
text/rtf	Rich Text Format	251	214	213
application/vnd.ms-powerpoint	PowerPoint Presentation	188	9	6
application/x-msi	Microsoft Installer	106	1	-
application/x-iso9660-image	ISO 9660 CD-ROM	9	-	-
text/plain	Plain Text Document	6	12	13
application/vnd.openxmlformats-officedocument.presentationml.presentation	PowerPoint OOXML	6	4	2
application/chm	Windows HtmlHelp Data	4	-	-
text/vbscript	Visual Basic Source	3	-	-
inode/x-empty	Empty File	3	3	-
audio/mpeg	mp3 Media File	3	-	-
application/x-java-applet	Java Archive Data (JAR)	3	7	-
application/zip	ZIP Archive	1	52	8
application/x-bytecode.python	Python Compiled Bytecode	1	-	-
application/CDFv2	Compound Document Format Version 2	-	663	-
application/octet-stream	Binary Data	-	147	-
application/encrypted	Encrypted Data	-	100	-
application/x-hwp	Hangul Word Processor Document	-	66	-
image/vnd.fpx	Media Type Flashpix	-	-	1,240
application/vnd.ms-word.template.macroEnabledTemplate	Microsoft Word Macro-enabled Template	-	-	226
unknown	N/A	-	-	71

Table 4.3: Pairwise agreement in identifying MIME types.

	Magika	**Libmagic**	**ExifTool**
Magika	1.000000	0.825336	0.698657
Libmagic	0.825336	1.000000	0.742681
ExifTool	0.698657	0.742681	1.000000

within our dataset, we employed three distinct tools: Python-magic [91], ExifTool [82], and Magika [69]. Python-magic is a widely used library that leverages libmagic, the ubiquitous library behind the file command, to identify file types using a predefined database of magic numbers and header patterns. ExifTool, though primarily focused on metadata extraction, can also identify file types based on header information in various formats. Finally, Magika, an open-source library developed by Google, utilizes a deep learning model to perform file type identification, including complex or less common formats.

Table 4.2 shows significant discrepancies among the three MIME identification tools used in our analysis. Notably, Magika classified 521 files (5.73%) as “application/ms.outlook” (indicating Microsoft Outlook messages), while libmagic assigned the generic category “application/CDFv2” to all of them. “CDFv2" typically references Compound Document Format Version 2, however, it serves as a container format and lacks the granularity to differentiate between specific document types like Word, Excel, or PowerPoint. Furthermore, libmagic assigned the “application/msword” label to 264 files (8.38%), while Magika categorized them as “application/vnd.ms-powerpoint”, “application/vnd.ms-excel", or even “application/x-msi” (Microsoft Installer) for two files. Interestingly, Magika identified a considerably higher number of MSI files (106, or 1.16%) compared to ExifTool and libmagic, which only identified one. The presence of these MSI files within the dataset warrants further investigation as part of future work. Libmagic also exhibited limitations in identifying RTF files, misclassifying 37 (14.74%) as “text/plain” or “text/octet-stream” (generic binary data). ExifTool presented further inconsistencies. It failed to identify the MIME type for 71 (0.78%) files and assigned two unexpected types: “application/vnd.fpx” (flashpix image) for 1,240 (13.67%) files and “Microsoft word macro-enabled template” for 226 (2.48%) files. This analysis highlights the limitations of current approaches in handling the diverse document formats.

To further assess the degree of inconsistency between different approaches, we calculated the pairwise agreement rates in identifying MIME types for the 9,086 samples within our dataset. Table 4.3 presents the computed agreement rates between libmagic, ExifTool, and Magika. As our results show, the agreement rates vary across tool pairs. Magika and libmagic exhibit the highest agreement of 82.53%. Conversely, the agreement between Magika and ExifTool is the lowest at 69.86%. These discrepancies suggest potential for malicious actors to disguise file formats to evade detection and show the limitations of current approaches in handling less common file formats.

Our Solution: Majority Voting. Effective extraction of malicious content relies on accurate file type identification. As established in Section 4.2, distinct file formats possess unique structures and characteristics, and misidentifying the file type can lead to errors and missed malicious indicators. To overcome this challenge, we employ a multi-step voting mechanism to ensure the most accurate file type identification and increase the likelihood of extracting malicious content. Initially, we strive for a unanimous decision among libmagic, ExifTool, and Magika. If all three agree, we adopt their consensus. In the case of disagreement, a majority vote (two out of three tools) determines the file type. However, when the tools disagree or identify generic formats (e.g., CDFv2), we use Magika’s “best guess” indicator. This functionality utilizes the deep learning model to provide its most likely type based on previous encounters and confidence levels. To demonstrate the efficiency of our approach, we ran our malicious content extractor (described below) without the file type detection. Our approach processed and extracted indicators from 13.34% more files (99.80% vs. 86.46%). This highlights the importance of using a multi-faceted approach for analyzing malicious documents, as dependence on a single approach can introduce limitations.

4.3.3 Malicious Content Extraction

Prior research has explored various aspects of document-based malware, including analyzing malicious VBA macros [166, 167], or analyzing embedded images and textual information within documents [270]. Additionally, research efforts have addressed the challenge of complex macro code by developing techniques for automated extraction of Indicators of Compromise (IoCs) through deobfuscation [108, 222]. Building upon prior research, our content analysis pipeline utilizes oletools [117], a well-established suite within the malware forensics community for document parsing and macro extraction.

To understand the social engineering tactics embedded within documents, we leverage the Python library textract [139]. This library extracts text content from a wide range of document formats. However, we encounter extraction errors with certain file types, particularly newer OOXML documents containing a large number of pages and complex layouts. For these file types, we employ a dedicated extraction pipeline using the python-docx module [36], ensuring comprehensive text extraction.

Finally, we employ Yara, a rule-based tool that matches text phrases, code snippets, and unique patterns within a file to identify potential threats. We leverage a curated set of Yara rules from reputable sources maintained by active developers, including Inquest Labs [94], Florian Roth [220], and Ditekshen [54].

For streamlined analysis, we perform asynchronous scanning of malicious documents. We use 7zip [1], a widely used archive utility, to extract all storage and streams from the OLE compound files and OOXML zip containers. This process creates a folder and an individual file for each successfully extracted storage and stream. Subsequently, we apply the Yara rules to both the original and extracted content. Our Yara analysis pipeline outputs detailed information for each processed file, including the number of extracted files, file matches, matched strings or byte patterns, and the corresponding Yara rule.

Table 4.4 details the success rates for extracting various features from our dataset of 9,086 files. We were able to extract macro content from 6,944 samples (76.42%) and textual information from 3,414 samples (37.57%). Finally, 8,934 files (98.44%) triggered at least one Yara detection rule. Moreover, Table 4.5 details the frequency of the different Yara rules identified in our dataset. "Office Documents with VBA Projects" are the most frequently triggered rule (87.98%), followed by "Microsoft Excel Hidden Macrosheets" detected in 36.20% of samples.

It is worth noting that our processing pipeline encountered errors during the analysis of 18 files (0.2%), resulting in a processed file count of 9,068 files as shown in Table 4.4. These errors stemmed from two main file types. First, thirteen files were identified as Microsoft Word documents with obfuscated tags in VirusTotal. This obfuscation might render them unprocessable for further analysis. The five other files were RTFs, which we identified to be associated with the Royal Road Hacking Tool used by the Chinese APT group APT29 (also known as Royal Road) [55]. This sophisticated malware exploits previously unknown vulnerabilities to create corrupted RTF documents that trick victims into executing malicious code. Notably, this malware was used in a high-profile attack

Table 4.4: Frequency of extracted features.

Stage of Feature Extraction Pipeline	# of Samples
Processed Files	9,068 (99.80%)
Extracted Macros	6,944 (76.42%)
Extracted Text Content	3,414 (37.57%)
Yara Rules Triggered	8,934 (98.32%)

Table 4.5: Frequency of triggered Yara rules.

Triggered Yara Rule [54, 94, 220]	# of Samples
Office_Document_with_VBA_Project	7,860 (87.98%)
Microsoft_Excel_Hidden_Macrosheet	3,234 (36.20%)
Windows_API_Function	2,932 (32.82%)
Microsoft_Excel_with_Macrosheet	1,838 (20.57%)
Suspicious_PowerShell_WebDownload	679 (7.60%)
Powershell_Command_Fileless_Malware	654 (7.32%)
SUSP_Excel4Macro_AutoOpen	490 (5.48%)
Base64_Encoded_URL	368 (4.12%)
Office_AutoOpen_Macro	365 (4.09%)
PowerShell_in_Word_Doc	343 (3.84%)

against the Mongolian Ministry of Foreign Affairs, highlighting its potential for significant disruption [55].

4.4 Malicious Document Characteristics

Macros. Our macro extraction pipeline was able to extract macros from 6,944 samples (76.42%), meaning 2,124 files had no detectable macros. However, further analysis revealed 708 files (33.33%), where our pipeline identified the presence of macros but failed to extract the clear-text macro code. Supporting this, Table 4.5 shows the Yara rule for VBA projects in Office documents (macro-enabled documents) triggered in 87.98% of the files analyzed. We hypothesize that these unextracted macro codes could be corrupted or heavily obfuscated, requiring manual analysis or advanced techniques for extraction. Another possibility is that they might be embedded unconventionally and located within non-standard parts of the document.

To further analyze the threats within macro-enabled documents, we investigate the combination of triggered Yara rules. The most prevalent and insightful combination is documents flagged as "Office Document with VBA Project" appearing alongside "Microsoft Excel Hidden Macrosheet" (374 occurrences). This suggests the utilization of *hidden* macrosheets within Excel documents (described in Section 4.4.1), likely to bypass detection. Another common combination occurring in 344 samples is "Microsoft

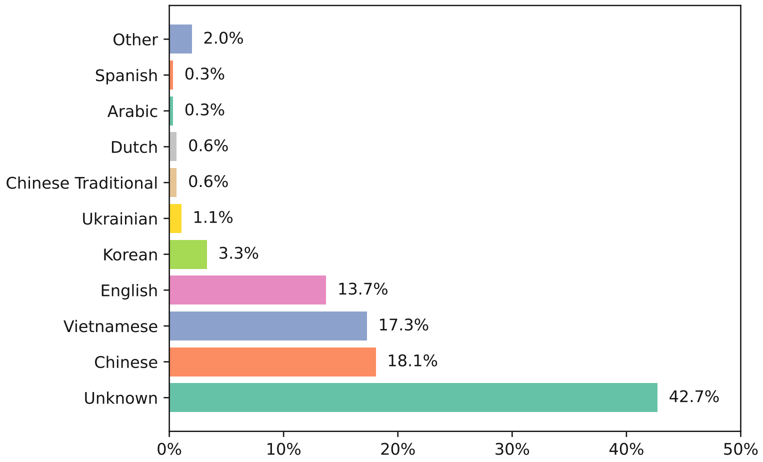

Figure 4.1: Distribution of languages within the malicious documents.

Excel with Macrosheet" with "Office Document with VBA Project," highlighting the general risk associated with Excel macrosheets. Finally, we observed another common Yara combination of "Office Document with VBA Project" with PowerShell commands and fileless malware (e.g., "Powershell Command Fileless Malware"). This suggests the sophisticated use of macro-enabled documents as a delivery mechanism for fileless malware attacks. We further investigated the signature associated with this technique and identified the use of several embedded Windows API calls. PowerShell, a scripting language, can utilize Windows API functionalities to perform a variety of malicious actions, such as downloading additional malware, manipulating files, or executing commands.

Deceptive Text. We further investigate the textual content of socially engineered documents within our dataset by analyzing the language distribution. Figure 4.1 shows the frequency of the most common languages, such as Chinese, Vietnamese, and English, likely reflecting international campaigns. Interestingly, a significant portion of documents fell into the 'unknown' category for two key reasons. First, some documents triggered low-confidence language accuracy, requiring further analysis. Second, a manual examination of a sample set revealed documents containing scripts or macro commands embedded within the textual content. Additionally, examining the relationship between text and macros, we found that 1,468 (42.97%) of the 3,414 files containing text lacked any evidence of macro code, suggesting not all malicious documents with textual content rely on macros for malicious behavior. However, the majority (70.98%) of these macro-less files were Excel documents, where further analysis revealed the presence of XL4 macros embedded within the cell content of the spreadsheets (discussed in Section 4.4.1).

No Indicators. We identified 46 DOCX, 30 XLSX, and 12 OneNote documents with no immediate malicious indicators such as macros or text content. OneNote files (.one) present a unique challenge for analysis. While they represent the native format for Microsoft OneNote, a digital note-taking application, current parsing tools struggle to handle them effectively and Magika misclassified nine OneNote files as ISOs and three as MP3s. This is particularly concerning as recent research by Proofpoint indicates the TA577 cybercrime group is increasingly using OneNote documents for malware delivery, potentially paving the way for similar tactics by others [136]. This group has also been linked with high confidence to a March 2021 Sodinokibi ransomware attack that initially compromised victims via malicious Microsoft Office attachments containing macros that downloaded and executed IcedID malware [118]. While parsing OneNote documents presented challenges, we further identified 28 out of 30 XLSX files that lacked any malicious indicators to be linked with the Bluenoroff cryptocurrency campaign associated with North Korean threat actors [195]. By comparing the file hashes to those identified in public reports, we discovered a repetitive exploitation pattern in these XLSX files. All of them relied on the well-known CVE-2017-0199. The exploit involves fetching malicious content from a URL embedded in the document's metadata and then downloading a remote macro-enabled template for further execution. We discuss this technique in detail in Section 4.4.2. These findings suggest both the prevalence of Excel files as a delivery method in cyberattacks and the use of unconventional exploitation techniques involving Excel and OneNote files by more advanced attackers.

4.4.1 Evasion Techniques

When delving deeper into the challenges associated with extracting and analyzing malicious content we focus on Excel files: Microsoft Excel allows hiding sheets within a workbook, although one sheet must remain visible [161]. While hidden sheets can be unhidden through the interface or file manipulation, another state marked as "very hidden" requires hexadecimal editing [161]. Malware authors exploit this feature to embed malicious macros in hidden sheets, making them harder to detect (e.g., XL4 macros) [183, 184]. While both VBA and XL4 macros (discussed in Section 4.2) can be exploited maliciously, XL4 macros introduce unique complexities for analysis due to their intricate structure and dispersed execution logic [98].

In our dataset, we identify 300 occurrences of "Microsoft Excel Hidden Macrosheet" and "Microsoft Excel with Macrosheet," 270 occurrences of "Microsoft Excel Hidden Macrosheet," "Microsoft Excel with Macrosheet" and, "SUSP Excel4Macro AutoOpen" Yara rule combinations. Both entries highlight the prevalence of malicious macros hidden within Excel documents. The presence of the "SUSP Excel4Macro AutoOpen" rule in the second entry suggests an additional layer of concern. These documents not only contain hidden macros but also have macros specifically designed to run automatically upon opening the file. This increases the risk of immediate execution of malicious code if the user opens the document.

We further identified 708 files where our extraction pipeline detected the presence of macros but failed to extract the clear-text macro code. Interestingly, of the 708 unextracted macros, 625 (88.27%) originated from Excel files containing XL4 macros. This highlights the versatility of XL4 macros and the challenges in extracting the macro code for malicious content analysis. In response to these challenges, Ruaro et al. [222] designed Symbexcel, a symbolic execution engine specifically for analyzing and deobfuscating XL4 macros in

Excel 4.0. This technique aims to infer the "correct" values of any environment variables, leading to the deobfuscation of the malicious payload hidden within the macro.

We used Symbexcel to analyze 625 Excel files identified as containing XL4 macros. Symbexcel successfully extracted IoCs from the macro logic within 458 files (73.28%). However, it encountered limitations in processing of 154 files. One of the primary limitations is the lack of complete grammar that more accurately describes the Excel 4.0 Grammar. In several parsing scenarios, the grammar parser failed to evaluate formulas, such as `ShrFmla`, due to them deviating from their expected behavior. Additionally, certain non-standard function calls like `_xlfn.CONCAT` instead of `CONCAT()` also contributed to parsing errors. Parsing XL4 formulas correctly is crucial for analyzing these malicious documents. While it might initially appear straightforward, the syntactical features of Excel formulas are quite complex. These findings suggest the need for a more precise formula grammar, ideally one that completely matches the one implemented in Excel, to effectively handle complex Excel 4.0 malware.

4.4.2 Other Advanced Techniques

RTF exploits. Rich Text Format (RTF) files offer great versatility in document formatting but present significant challenges for robust security analysis. RTF heavily relies on control words to define document presentation (see Section 4.2). These control words, along with their associated parameters and data, can introduce vulnerabilities when parsing errors occur [234]. Attackers exploit these errors to embed malicious resources within the RTF file. Historical vulnerabilities such as CVE-2010-3333 and CVE-2014-1761 demonstrate the potential dangers of flawed RTF parsing implementations [271].

Our analysis confirmed the complexities involved in detecting threats within RTF documents. Apart from the ability of five malicious RTF files to evade parsing altogether (see Section 4.3.3), we were unable to extract malicious indicators within a substantial portion (60.55%) of the RTF files. For these files, we encountered "malformed OLE object" errors during analysis. These errors prevented the extraction of OLE objects, which could potentially contain hidden malicious content.

Our analysis also identified a subset of RTF files (4.71%) that caused errors due to malformed headers. According to Microsoft's specifications, a valid RTF document should begin with `{rtf1}` (RTF version 1.x). However, manual inspection revealed deviations in these files, where the header started with `{rt0}` or simply `{rt}`. While Microsoft Word can handle these variations, many analysis tools misinterpret them as plain text and fail to process them as RTF documents. Interestingly, all the files with malformed headers were linked to a Remcos RAT (Remote Access Trojan) malware campaign. Remcos, also known as Remote Control and Surveillance Software, is a sophisticated tool that grants attackers full control over compromised machines. In this campaign, the attackers used an RTF document with a malformed header and an Equation object to spread a variant of Remcos [274].

Despite these limitations, our Yara rule-based detection provided valuable insights. We identified 24 unique rules triggered across 256 RTF files. Table 4.6 details the broad classification of these rules and the corresponding number of detections for each category. The most prevalent threat among the RTF files is exploitation through specific CVEs, particularly CVE-2017-11882, CVE-2017-8759, and CVE-2018-0802 [48]. CVE-2017-11882 exploits a buffer overflow vulnerability within the Microsoft Equation Editor, allowing attackers to execute malicious code on opening a specially crafted RTF document. CVE-2017-8759 targets a SOAP WSDL parser code injection vulnerability within Microsoft Office RTF document,s allowing attackers to inject arbitrary code during parsing. Finally, CVE-2018-0802 represents a more general memory corruption vulnerability within Microsoft Office applications.

Beyond the exploitation of known vulnerabilities, our analysis revealed two additional frequently occurring indicators. The first involved a combination of embedded malicious objects within the RTF files and deviations from the standard RTF specifications. The second indicator focused on anti-analysis techniques through RTF header manipulation, header obfuscation, and embedding obfuscated OLE object headers.

External Relations and Template Injection. Standard detection tools often focus on malicious code embedded directly within documents. However, Microsoft Office allows documents to reference various resources, including external or remote templates. Attackers exploit this functionality by modifying the document's properties and utilizing *external relations*. These modifications point the document to a malicious template hosted on a remote server, such as a URL or a GitHub repository. Once the compromised document is opened, the malicious code from the external template is loaded and executed. Recent threat reports detail the external relations exploit where attackers crafted DOCX files to trigger macro execution from a remote .dotm template [67, 209]. To assess the prevalence of this technique within our dataset, we examined document relationships within the .rels folder, particularly the `settings.xml.rels` file. We identified 224 (2.47%) samples exhibiting signs of utilizing external relations to potentially download remote malicious content. Furthermore, Yara rules successfully flagged a majority of these samples, triggering indicators like "OLE RemoteTemplate" or "XML WebRelFrame RemoteTemplate." However, three files (less than 1.5%) bypassed detection. These outliers included two XLSX files using hyperlinks to malicious URLs and one PPTX file containing a link to a malicious OLE object. Notably, all three files had text content, suggesting a potential blend of social engineering and malicious links. Interestingly, these three samples were linked to targeted APT attacks, with one associated with a Chinese group known for targeting the Tibetan community [201].

While remote template injection poses a significant threat within OOXML formats, attackers have also exploited similar vulnerabilities in RTF documents. A recent report by Proofpoint details a technique known as RTF template injection [202]. RTF files store formatting instructions as plain text, and attackers manipulate the *template property* to redirect the file to execute a malicious script. Proofpoint observed a rise in this technique, primarily employed by APT groups linked to India and China. The simplicity

Table 4.6: Classification of Threats among RTF documents.

Threat Category	**Count**
Exploitation of CVEs (2017-1182, 2017-8759, 2018-0802)	264
Malicious Embedded Objects	253
RTF Header Manipulation	166
RoyalRoad Exploits	25

and effectiveness of RTF template injection suggest its potential for wider adoption by various cyber criminals. The trickle-down effect in cyberattacks further amplifies this concern [231]. As seen with the Royal Road tool, initially used by APT groups, the techniques become more widespread over time [55].

Dynamic Data Exchange (DDE). DDE is a protocol originally designed for data sharing between Microsoft Office applications. DDE was partially superseded by Object Linking and Embedding (OLE) and is currently maintained in Windows systems for backward compatibility [242]. However, attackers exploit DDE to execute malicious commands, including downloading additional payloads. This technique works in both OLE and OOXML file formats. While newer Office versions alert users about DDE commands within documents, attackers have adapted their phishing tactics to bypass these warnings. This method is commonly used by threat actors like APT28 and FIN7 [194]. One method of bypassing warnings involves manipulating the DDE syntax to craft obfuscated prompts. Attackers can achieve this by directly invoking PowerShell through modified parameters, resulting in prompts that appear less suspicious to users. This technique increases the likelihood of user interaction and subsequent compromise [52]. Basic DDE detection can be achieved through string scanning, which involves manually searching the text content of a file for keywords like `DDEAUTO` or `DDE`, although it can be time-consuming and can miss obfuscated instances. Another approach is using Yara rules to identify keywords and specific features associated with DDE exploitation. Our malicious content extraction pipeline, combining Yara and msodde [116] from oletools, detected 240 (2.64%) files using DDE techniques.

4.5 Threats to Validity

Our study offers valuable insights into the use of malicious documents, however, we acknowledge limitations in the dataset that might affect the generalizability of our findings. Our focus on malware delivered solely via malicious Office documents detected by VirusTotal introduces a potential bias. Limiting the scope to a specific attack vector excludes other methods employed by attackers, such as drive-by downloads, malicious PDFs, and weaponized email messages. This limitation restricts our understanding of the complete threat landscape. Moreover, VirusTotal relies on various antivirus engines and threat intelligence feeds to identify malware. While a valuable source, some malicious samples might evade detection by some engines, leading to the underrepresentation of certain types of malware in the dataset. This can skew the results towards malware types that are more easily identified by VirusTotal.

Despite these limitations, our findings can still be considered a valuable lower bound for the overall threat landscape. The high prevalence of malicious documents within our dataset suggests its significance as an attack vector. Furthermore, our insights can inform the development of mitigation strategies specifically for document-based threats.

4.6 Recommendations and Future Directions

Automating malicious document analysis remains critical for both post-mortem incident response and proactive threat prevention. In post-mortem analysis, understanding malicious actions performed is essential, while identifying IoCs helps track future attacks. However, current solutions face several limitations:

File Identification. Analyzing malicious Microsoft Office files is challenging due to the complexity and variety of file formats, each with its own potential for exploitation. Unfortunately, current methods often misidentify files, especially those that are incompatible or deliberately disguised. This leads to missed threat indicators and necessitates time-consuming manual analysis. However, our findings in Section 4.3.2 demonstrate that using a combination of multiple tools can substantially improve the identification process and reduce processing errors.

Macros and Obfuscation. Macro malware in documents continues to remain a prevalent threat, as detailed in Section 4.4, where 76.42% files in our dataset contained macros and the majority of them are Excel and Word documents. However, Excel files pose the most significant challenge. Their layered evasion and obfuscation techniques make content analysis difficult. Existing methodologies often struggle with heavily obfuscated macro code and incomplete grammar, as discussed in Section 4.4.1. To effectively combat complex Excel 4.0 malware, developing a comprehensive Excel 4.0 formula grammar and parser is essential. Here, Large Language Models (LLMs) offer a promising solution. LLMs trained on large datasets of real-world Excel formulas, encompassing both standard and non-standard functions, could potentially learn to recognize and interpret functions beyond those explicitly defined in the current grammar. Furthermore, as Excel evolves with new features and functions, the LLM's ability to learn continuously from new data would enable it to adapt and handle these changes automatically, reducing the need for constant grammar updates.

Unconventional Attack Vectors. RTF files have become a prominent attack vector. These techniques include embedding malicious resources directly within the document, exploiting remote template injection vulnerabilities, or leveraging parsing errors. As shown in Section 4.4.2, current static analysis approaches fail to extract indicators from a substantial portion of 60.55% files. Notably, in our analysis, we identified instances of sophisticated APT actors successfully using RTFs as attack vectors. Furthermore, OneNote file formats present additional parsing challenges. While Yara rules, with their predefined patterns and known threat signatures, offer some defense against known threats, they struggle with entirely novel or obfuscated malware. This creates a reactive approach, hindering our ability to proactively understand the evolving threat landscape and adapt to new file types and attack methods. The development of advanced static analysis tools that can effectively detect malicious document structures, reliably handle different file formats (RTFs, OneNote), and extract embedded malicious content would be a step towards proactively aiding in threat detection and analysis.

Future Directions. As an immediate improvement to the current static analysis methods we aim to enhance RTF file analysis techniques by focusing on two key areas: (1) identification and extraction of control words commonly used for embedding or obfuscating objects (e.g., `\objdata`, `\objemb`, `\template`), and (2) anomaly detection through analysis of control word order, combinations, and nesting patterns within the RTF structure to pinpoint unusual behavior. Additionally, we will investigate embedded OLE objects within identified control word groups, searching for non-standard characters and byte sequences that might indicate the presence of potentially malicious code or obfuscation. We further plan to investigate the social engineering tactics employed within the documents. Here, our initial focus is on incorporating an image extraction component to capture deceptive images. We will leverage OCR tools capable of processing images across various document formats to effectively analyze the content of embedded images. Furthermore, we plan to explore the translation of non-English content to English for a more comprehensive analysis of the types and variations of prompts used to lure users into opening or clicking on malicious documents. Finally, we will explore methods for identifying, parsing, and extracting critical file metadata and potentially malicious content from Microsoft OneNote documents.

4.7 Related Work

Malware analysis and automated malware detection techniques have long been a well-established field. While significant research has focused on executable files, malicious document files remain a relatively understudied area. Within the document space, extensive research exists on malicious PDF detection [119, 133, 239, 240, 275]. However, these approaches often rely on format-specific features and do not generalize to other file formats like Microsoft Office documents due to structural differences.

Several studies have explored document analysis through file structure examination. Otsubo et al. [191] investigated deviations from standard file formats in documents containing executables. Cohen et al. [43] extended this concept to XML-based Office documents, employing machine learning on extracted structural features for malicious document detection.

Other studies have explored VBA macro analysis for malicious document detection. Bearden et al. [21] classified macros using K-Nearest Neighbors on features extracted from p-code opcodes (translated VBA code) with TF-IDF weighting. Mimura et al. [165, 167] investigated raw VBA code with Doc2vec models and LSI for malicious macro detection. Kim et al. [108] proposed a machine learning approach for detecting obfuscated VBA macros, categorizing obfuscation techniques based on prior research. More recently, Casino et al. [37] proposed perceptual hashing of document images by extracting key document components for lightweight signature creation. Yan et al. proposed DitDetector [270], a bimodal learning approach using visual and textual information from document previews for macro malware detection.

In a more targeted line of research, Ruaro et al. [222] proposed Symbexcel, a symbolic execution approach for automated deobfuscation and analysis of Excel 4.0 macros (XL4), a growing attack vector within documents. Complementing this line of research, Blonde et al. [122] performed a measurement study on targeted attack documents of 3,815 samples identifies attacker focus. The authors identified several attacks targeting specific regions and ethnicities, highlighting the trend of socially engineered malware. They also found a focus on exploiting known vulnerabilities, indicating attackers prioritize readily available attack vectors.

Extending on prior research, we conducted a measurement study of malicious document samples from the past four years (2020-2023). We focused on Microsoft document formats to identify both the common techniques employed by attackers and the limitations of existing approaches in detecting these threats.

4.8 Conclusion

Non-binary files, particularly Microsoft Office documents, are a prevalent initial infection vector. The widespread use of Office suites, coupled with their inherent complexity and persistent vulnerabilities, creates a prime target for attackers. Thus, understanding these malicious vectors is as crucial as analyzing executables. Our analysis revealed significant limitations in current document analysis approaches. In 17.47% of the analyzed files, the malicious file type could not be definitively identified, leading to a 13.34% error rate in processing. We attribute this primarily to the obfuscated file formats and the inability of toolchains to handle emerging file types. Furthermore, current approaches struggle to extract malicious content from obfuscated Excel macros (24.64%) and reliably parse and extract malicious indicators from RTF file formats (60.55%). These issues suggest that malicious documents remain a prominent attack vector. Achieving a level of maturity in document analysis comparable to executable file analysis necessitates robust frameworks for the automated processing of diverse file formats. Parsing tools capable of handling a wider range of formats and reliably extracting malicious components for forensic analysis are essential. With this study, we aim to inform and guide further research efforts in the malicious document ecosystem.

CHAPTER 5

From IOCs to Group Profiles: On the Specificity of Threat Group Behaviors in CTI Knowledge Bases

Indicators of Compromise (IOCs) such as IP addresses, file hashes, and domain names are commonly used for threat detection and attribution. However, IOCs tend to be short-lived as they are easy to change. As a result, the cybersecurity community is shifting focus towards more persistent behavioral profiles such as the Tactics, Techniques, and Procedures (TTPs) and the software used by a threat group.

However, the distinctiveness and completeness of such behavioral profiles remain largely unexplored. In this work, we systematically analyze threat group profiles built from two open cyber threat intelligence (CTI) knowledge bases: MITRE ATT&CK and Malpedia. We first investigate what fraction of threat groups have group-specific behaviors, i.e., behaviors used exclusively by a single group. We find that only 34% of threat groups in ATT&CK have group-specific techniques. The software used by a threat group proves to be more distinctive, with 73% of ATT&CK groups using group-specific software. However, this percentage drops to 24% in the broader Malpedia dataset. Next, we evaluate how group profiles improve when data from both sources are combined. While coverage improves modestly, the proportion of groups with group-specific behaviors remains under 30%. We then enhance profiles by adding exploited vulnerabilities and additional techniques extracted from more threat reports. Despite the additional information, 64% of groups still lack any group-specific behavior. Our findings raise concerns on the belief that behavioral profiles can replace IOCs in threat group attribution.

5.1 Introduction

Indicators of Compromise (IOCs) – such as malicious IP addresses, file hashes, domain names, emails, and cryptocurrency addresses – are widely used for detecting and attributing threats but offer only a snapshot of an adversary's activity. IOCs are often ephemeral and easily changed by threat actors, limiting their long-term effectiveness for threat detection and attribution.

To address these shortcomings, prior work argues to instead focus on behavioral characteristics, such as the Tactics, Techniques, and Procedures (TTPs) used to gain access, move laterally, maintain persistence, and exfiltrate data [16, 28, 102, 214, 228, 238, 268].

Behavioral characteristics are thought to be more robust, remain more stable over time, have a larger cost for adversaries to change them, and be able to link seemingly unrelated attacks from the same threat group. Models like the Pyramid of Pain [22] place TTPs at the top of the pyramid in terms of cost for an adversary to change them. Apart from TTPs, other behavioral characteristics also exist, for example, the software tools used by a threat group may be distinctive, particularly if the malware is developed in-house. Similarly, exploited vulnerabilities can be characteristic, especially when a group targets uncommon software or uses custom-developed exploits.

Even the textual content used in campaigns can be characteristic of a threat group with recent works leveraging phishing SMS contents [181], ransomware notes [129], and cross-file-type features [225] to identify attacks from the same campaign and threat group.

We refer to a threat group's observed behaviors as its *group profile*. Accurate threat group profiles are fundamental for incident correlation, attribution, building behavioral detection rules, and proactive threat hunting. Group profiles can be built from the contents of the threat reports published by security vendors and analysts, which typically describe, in natural language, the analysis of specific attacks and their attribution to specific threat groups. Threat reports may come from a single source (e.g., a specific cybersecurity vendor) or be aggregated by cyber threat intelligence (CTI) knowledge bases [199, 243] and sharing platforms [29, 102].

While the potential of such behavioral profiles has been widely acknowledged, few studies have examined how distinctive the behavioral profiles of threat groups truly are, and how complete our understanding of those behaviors is, especially given the varying quality and scope of the data sources used to build these profiles. One concern is that many behaviors in these profiles can be *generic*, i.e., used by many threat groups, thus providing little information on the groups using them. These include common techniques (e.g., spearphishing, malware auto-start through registry keys), widely available software (e.g., abused penetration testing tools, open-source projects, malware kits sold in underground forums), and prevalent vulnerabilities (e.g., those affecting popular software with public exploits). Multiple threat groups often acquire such tools and exploits for reasons of convenience, reduced operational cost, or to obscure attribution.

This raises the question of which behaviors are truly *group-specific*, i.e., used by only a single threat group. Group-specific behaviors, when observed on a protected system, can serve as behavioral signatures that uniquely identify the responsible threat group. However, what fraction of threat groups exhibit group-specific behaviors remains an open question. Determining whether a behavior is truly group-specific requires not only analyzing the group that exhibits it, but also having comprehensive coverage of behaviors across other threat groups. Without this broader context, a behavior might appear unique when it is not. For example, as more threat reports become available, a behavior initially believed to be exclusive to group A may also be observed in group B, indicating it is not unique.

In this work, we perform a systematic analysis of group profiles in CTI knowledge bases, their complementary value, and how they can be extended. We focus on knowledge bases that are open (i.e., non-commercial), index many threat reports, are periodically updated, provide a taxonomy of threat groups, and organize information into group profiles. These profiles include basic group metadata (e.g., name, aliases, country), references to related threat reports, and descriptions of group behaviors mentioned in those threat reports. We find two knowledge bases satisfying those properties: MITRE's Adversarial Tactics, Techniques, and Common Knowledge (ATT&CK) [175] and Malpedia [70]. We discard other open projects like the MISP threat actor galaxy [170], ThreatMiner [249], and APTnotes [13], as they collect threat reports and associate them with threat groups but do not extract or include behavioral information from those reports into their group profiles. Moreover, Malpedia incorporates data from the MISP threat actor galaxy, allowing us to cover that project indirectly. We also exclude commercial services that provide specialized threat reports to paying customers [28], as their knowledge bases are proprietary. Both ATT&CK and Malpedia provide their own taxonomies of threat groups and associated software tools. A key distinction is that ATT&CK also includes a taxonomy of TTPs, integrating attack techniques directly into the group profiles. Using the data in these two knowledge bases, we analyze the following three research questions:

RQ1: What fraction of the threat groups in ATT&CK and Malpedia have group-specific behaviors? We separately analyze the group profiles created using only the information available in ATT&CK and Malpedia. Identifying groups based on TTPs is challenging as only 52 (34.2%) groups in ATT&CK have group-specific techniques. It is somewhat easier to identify groups through the software they use with 111 (73.0%) groups in ATT&CK having group-specific software. However, this percentage is significantly lower in Malpedia, where only 192 (24.0%) have group-specific software. This discrepancy stems from threat reports disproportionately focusing on a subset of high-notoriety groups, leaving less-known groups with sparse reports to build their profiles. Combining techniques and software into joint profiles increases the groups with group-specific behaviors from 111 (73.0%) to 124 (81.6%).

RQ2: How complementary is the information in both datasets? How much do group profiles improve when combining data from both sources? Both datasets differ substantially in volume. Malpedia provides a broader coverage of the threat landscape than ATT&CK, comprising 5.2 times more threat groups (800 vs. 152) and 4.2

times more software (3,367 vs. 794). This expanded scope stems from Malpedia indexing 16.9 times more threat reports (15,699 report URLs vs. 930). To assess their overlap, we normalize group and software names across the datasets. Both datasets have little overlap with only 145 groups and 498 software entries in common. The corresponding Jaccard Index values are 17.7% for groups, 13.5% for software, and just 3.2% for report URLs. The low intersection indicates that each dataset captures a different view of the threat group landscape highlighting their complementary nature.

We create joint group profiles using the data from both datasets, identifying 236 (29.2%) groups with group-specific behaviors, compared to 124 groups using only ATT&CK and 192 using only Malpedia. Despite combining both datasets, over 70% of groups have no group-specific behavior.

RQ3: What additional information currently not in ATT&CK and Malpedia could make threat group profiles more complete? We examine how group profiles can be improved with additional information extracted from threat reports. First, we extract CVE identifiers to build vulnerability profiles for each threat group. The number of groups with at least one group-specific vulnerability is 48 (31.6%) in ATT&CK, 112 (14.0%) in Malpedia, and 119 (14.7%) when combining both datasets. Thus, exploited vulnerabilities tend to be less distinctive than the software used, but more than the techniques used. Next, we extend the group profiles with additional technique identifiers extracted from the threat reports. Since Malpedia does not provide TTPs, this step allows us to extend its group profiles with techniques. Incorporating these extracted techniques increases the number of groups with group-specific behaviors from 52 (6.4%) to 68 (8.4%). Finally, we combine all available behavioral indicators, including techniques, extracted techniques from reports, software, and vulnerabilities, into unified group profiles, identifying 291 (36.0%) groups with at least one group-specific behavior. Despite leveraging all available information, a majority of groups (64%) have no group-specific behaviors.

To better understand the limitations of current group profiles, we also discuss the impact of under-reporting, i.e., incomplete coverage of threat group behaviors. We observe that the number of technique identifiers extracted from the ATT&CK threat reports is larger than the number of techniques officially cataloged in ATT&CK from those same reports. This discrepancy likely arises from the manual nature of the report analysis process by ATT&CK contributors, emphasizing the need for automated approaches to extract TTPs from threat reports [6, 92, 205]. We also observe that only 46.3% of techniques and 64.1% of software entries in ATT&CK, and just 28.6% of software in Malpedia, are currently associated with at least one threat group. The remaining entries were likely added to the taxonomies because they were observed being used by adversaries in the wild. However, their lack of association with specific threat groups highlights the incomplete coverage of group profiles.

Artifacts. Our open-source code and data is at: github.com/SecPriv/ThreatGroupCTI.

5.2 Dataset Comparison

This section first details the information in ATT&CK [175] and Malpedia [70] and then sets the base for answering RQ2 by analyzing the extent of data overlap between the two datasets and assessing how their contents complement each other.

5.2.1 Datasets

ATT&CK. ATT&CK provides taxonomies of offensive and defensive techniques, software tools used by adversaries, and threat groups. The techniques taxonomy comprises three *domains*: Enterprise, Mobile, and ICS. Each domain defines a set of *tactics* that correspond to different steps in the kill chain, such as Reconnaissance (TA0043), Persistence (TA0003), and Lateral Movement (TA0008). Each tactic includes a set of *techniques*. For example, Active Scanning (T1595) and Phishing for Information (T1598) are techniques under the Reconnaissance tactic. Techniques can also contain *sub-techniques*. For example, T1204.002 corresponds to the Malicious File sub-technique under the User Execution (T1204) technique.

The threat group taxonomy covers nation-state actors, advanced persistent threats (APTs), and some large for-profit actors such as ransomware groups. The profile of a threat group contains a unique identifier, a name, a list of aliases (called *Associated Groups*), and the techniques and software used by the threat group.

Entries in the software taxonomy are categorized into Tools and Malware. Tools include commercial software (e.g., Cobalt Strike), open-source frameworks (e.g., Metasploit, Mimikatz), and built-in OS tools (e.g., PsExec, ipconfig). Malware includes families specific to a single threat group (e.g., Carbanak) as well as malware kits available in underground markets and used by multiple threat groups (e.g., PoisonIvy RAT). The focus is on software used by APTs listed in the group taxonomy; however, ATT&CK also catalogs non-APT malware such as the Conficker worm [177] and the SimBad Android malware [178]. Each taxonomy entry contains URLs to threat reports related to the entry, such as reports describing the techniques and software used by a threat group.

Since its public release in 2015, ATT&CK publishes a new version approximately every six months, with the latest version at the beginning of this work being 15.1, released in April 2024. Each version may add new taxonomy entries (e.g., groups, techniques, software), remove *revoked* entries, or mark entries as *deprecated* (i.e., to be revoked soon).

Malpedia. Malpedia provides taxonomies of threat groups and software. It does not provide a taxonomy of techniques nor reference the techniques in ATT&CK. Similar to ATT&CK, the threat group taxonomy focuses on APTs and nation-state actors, whereas the software taxonomy aims to cover any malware family, regardless of whether it is used by APTs or other types of attackers (e.g., for-profit actors). The software taxonomy also includes a few security tools (e.g., Cobalt Strike) but does not differentiate between malware and tools. Each taxonomy entry for a threat group or software includes URLs of threat reports related to the entry. We collect information about groups and software

Table 5.1: Dataset summary. Malpedia does not have a techniques taxonomy. The low intersection and Jaccard Index show that both datasets have little overlap. We use the union of both datasets to build group profiles.

Data	**ATT&CK**	**Malpedia**	∩	∪	**Jaccard**
Groups	152	800	145	807	17.7%
Techniques	839	-	-	839	-
Software	794	3,367	498	3,663	13.5%
Report URLs	930	15,699	522	16,107	3.2%
Report FQDNs	218	2,002	194	2,026	9.6%
Reports	920	14,983	80	15,816	0.5%

through the Malpedia API and metadata about threat reports (e.g., URL, title, author, publication date) from the provided BibTex file. Malpedia is updated daily by adding new bibliographic references labeled with the associated threat groups and software. We obtained Malpedia data on February 18, 2025.

Dataset comparison. Table 5.1 summarizes the contents of both datasets. ATT&CK v15.1 contains 152 groups, 358 techniques, 481 sub-techniques, 794 software entries, and 930 URLs of threat reports from which those associations are extracted. Of the 358 techniques, 121 (33.8%) have at least one sub-technique, while 237 (66.2%) do not have sub-techniques. Among the total 839 techniques and sub-techniques, 637 (75.9%) belong to the Enterprise domain (202 techniques and 435 sub-techniques), 119 (14.2%) to Mobile (73 techniques and 46 sub-techniques), and 83 (9.9%) to ICS (83 techniques and no sub-techniques). For simplicity, in the remainder of this manuscript, we use the term *techniques* to refer to the combined set of 839 techniques and sub-techniques

In contrast, Malpedia does not include techniques; however, it is much larger, containing 800 groups (5.2 times more), 3,367 software (4.2x), and 15,699 (16.9x) report URLs. The report URLs in ATT&CK come from 218 domains, compared to 2,002 (9.2x) domains in the Malpedia URLs, showing that Malpedia draws from a significantly more diverse set of sources (e.g., cybersecurity vendors and analyst blogs). We download the content of each URL, filter errors, and identify reports by the SHA256 of the downloaded content (most often an HTML page or a PDF document). In total, we downloaded 920 unique reports from the 930 ATT&CK URLs and 14,983 unique reports from 15,699 Malpedia URLs. We use these downloaded reports to extract additional information for extending the group profiles discussed in Section 5.4.

5.2.2 Dataset Intersection and Union

This section examines the overlap between the two datasets and evaluates the benefits of combining their data. A key challenge in this comparison is that the names of groups and software differ across the datasets. To address this, we first created a mapping to align them.

Each knowledge base provides a name and a list of aliases for each threat group. We first normalized all names and aliases by converting them to lowercase, removing common suffixes such as “group” or “framework,” replacing terms like “team” with a space, converting terms like “threat group” to “TG,” and removing prefixes such as “TEMP”. Then, we compute the intersection between the set of names and aliases for each group in each taxonomy. If a group in ATT&CK shares a name or alias with a group in Malpedia, we merge them by performing the union of their sets. After the merging, we select a unique name for each normalized group. The selected name is the one used in ATT&CK by default and the one used in Malpedia if the group is not in ATT&CK. The normalization process identified 145 groups common to both datasets, seven groups unique to ATT&CK, 655 groups found exclusively in Malpedia, and 807 groups in the union of both datasets. We perform a similar normalization for software. We first normalize all software names and aliases by removing common prefixes (e.g., trojan, win, apk, elf) and replacing special characters (e.g., _rat to rat). Then we merge software entries that share at least one normalized name. The normalization identifies 498 software that appear in both datasets, 2,869 only present in Malpedia, 296 only present in ATT&CK, and 3,663 in the union of both datasets. We will publicly release our mappings of group and software names.

Table 5.1 also presents the overlap and union of report URLs, their fully-qualified domain names (FQDNs), and the SHA-256 hashes of the downloaded reports. We find only 522 report URLs shared between ATT&CK and Malpedia, resulting in a low Jaccard Index (JI) of 3.2%, indicating minimal overlap in referenced sources. The overlap based on actual report content is even smaller, only 80 downloaded reports have identical SHA-256 hashes, yielding a JI of just 0.5%. This discrepancy arises because downloading the same URL multiple times, especially in the case of HTML pages, can produce different files due to non-deterministic content such as dynamic metadata or embedded advertisements. Overall, the overlap between the datasets is quite low, with Malpedia providing a much broader view of the threat landscape. This disparity may be partly due to ATT&CK accepting contributions only from selected entities, which restricts the number of threat reports included in its analysis. However, this selective approach contributes to under-reporting. To address this limitation, we build group profiles by combining group and software information from the union of both datasets. Note that technique-level information is only available from ATT&CK and thus cannot be supplemented from Malpedia.

Takeaway: Malpedia provides a larger coverage of the threat landscape, including 5.2 times more groups and 4.2 times more software than ATT&CK. While not a strict superset of ATT&CK, Malpedia covers 95.4% of ATT&CK’s groups and 62.7% of its software. Combining both datasets increases the overall coverage of the threat landscape.

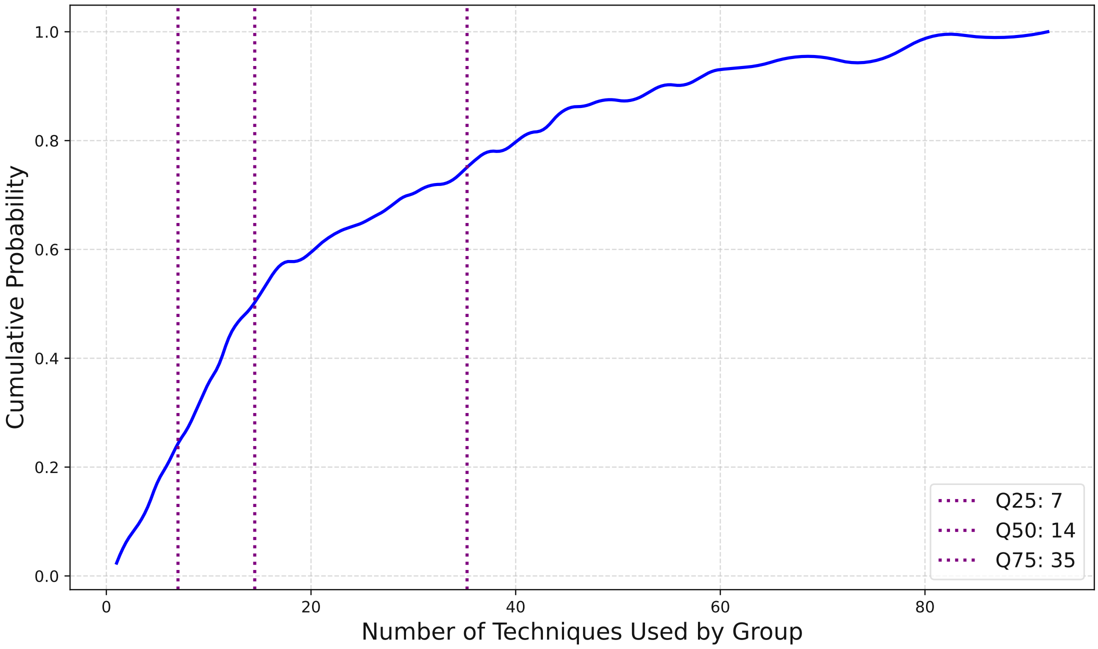

Figure 5.1: **CDF of the number of techniques per group:** 25% of groups use 7 or fewer techniques, 50% use 14 or fewer, and 75% use 35 or fewer.

5.3 Group Profiles in the Datasets

This section first addresses RQ1 by quantifying the proportion of threat groups in ATT&CK and Malpedia that have group-specific behaviors. It then addresses RQ2 by evaluating whether combining the datasets improves the group profiles.

5.3.1 Technique Profiles

In ATT&CK, the association of techniques to groups is provided as three separate group spreadsheets, one per domain. We combine the three group spreadsheets to obtain the set of techniques associated to each group, which we term the group's *technique profile*.

We first measure the size of the technique profiles. Figure 5.1 shows the cumulative distribution function (CDF) of techniques per group. On average, each group uses 23.2 techniques. 38 (25%) have at most 7 techniques, 76 (50%) have between 7 and 36, and 38 (25%) have more than 35 techniques. The Lazarus Group (G0032) has the highest number of techniques, with 92 techniques. Four groups have no associated techniques and therefore cannot be identified through their TTPs. We next examine whether the remaining 148 groups contain group-specific techniques.

We build a mapping from each technique and sub-technique to the threat groups that use them. Among the 839 techniques cataloged in the ATT&CK framework, 388 (46.3%) have not been associated with any group, 147 (17.5%) are linked to a single group, 287

Table 5.2: Top generic techniques, i.e., used by the largest number of groups.

ID	Technique Name	Groups
T1204.002	User Execution: Malicious File	79 (9.8%)
T1105	Ingress Tool Transfer	76 (9.4%)
T1566.001	Phishing: Spearphishing Attachment	72 (8.9%)
T1059.001	Command & Scripting: PowerShell	69 (8.5%)
T1588.002	Obtain Capabilities: Tool	66 (8.2%)
T1059.003	Command & Scripting: Win Command Shell	60 (7.4%)
T1036.005	Masquerading: Match Legitimate Name or Location	50 (6.2%)
T1547.001	Boot or Logon Autostart Execution: Registry Run Keys / Startup Folder	50 (6.2%)
T1071.001	Application Layer Protocol: Web Protocols	47 (5.8%)
T1082	System Information Discovery	46 (5.7%)

(34.2%) are associated to 2–37 groups, and 17 (2.0%) are used by at least one quarter (38) of all groups. The fact that 388 (46.3%) of all techniques in ATT&CK are not associated with any group raises concerns about coverage, as these techniques were presumably added to the taxonomy based on observed adversary behavior, yet remain unlinked to any known group.

Generic techniques. We call techniques used by many groups *generic*, as their presence in a protected environment offers limited value in distinguishing specific adversaries. Table 5.2 lists the top 10 techniques by number of groups. The most common technique is Malicious File (T1204.002) used by 79 groups where adversaries rely on users opening a malicious file, followed by Ingress Tool Transfer (T1105, 76 groups) where adversaries transfer tools or files from an external system into a compromised environment, and Spearphishing Attachment (T1566.001, 72 groups) where emails with a malicious attachment are used as a vector of initial compromise.

Additionally, we analyze which pairs of techniques tend to occur together. For this we compute the co-occurrence rate $\frac{|A \cap B|}{max(|A|,|B|)}$ where A and B are the sets of groups using technique A and B, respectively. We find five pairs with a co-occurrence rate of at least 0.75. The highest rates are 0.951 between Malicious Link (T1204.001) and Spearphishing Link (T1566.002), followed by 0.886 for Malicious File (T1204.002) and Spearphishing Attachment (T1566.001). The four techniques in these two pairs are generic, each used by at least one quarter of the groups, and are also semantically related, where a spearphishing link is a type of malicious link, and the attachment in a spearphishing email is a malicious file that the user is encouraged to open.

Technique profile similarity. We compute the similarity of the technique profiles of each pair of groups using the Jaccard Index. The mean Jaccard Index is 0.07, the median is 0.06, and the maximum Jaccard Index is 0.55. Since most pairs have low similarity scores, it suggests that each group's technique profile is quite unique. However, this may

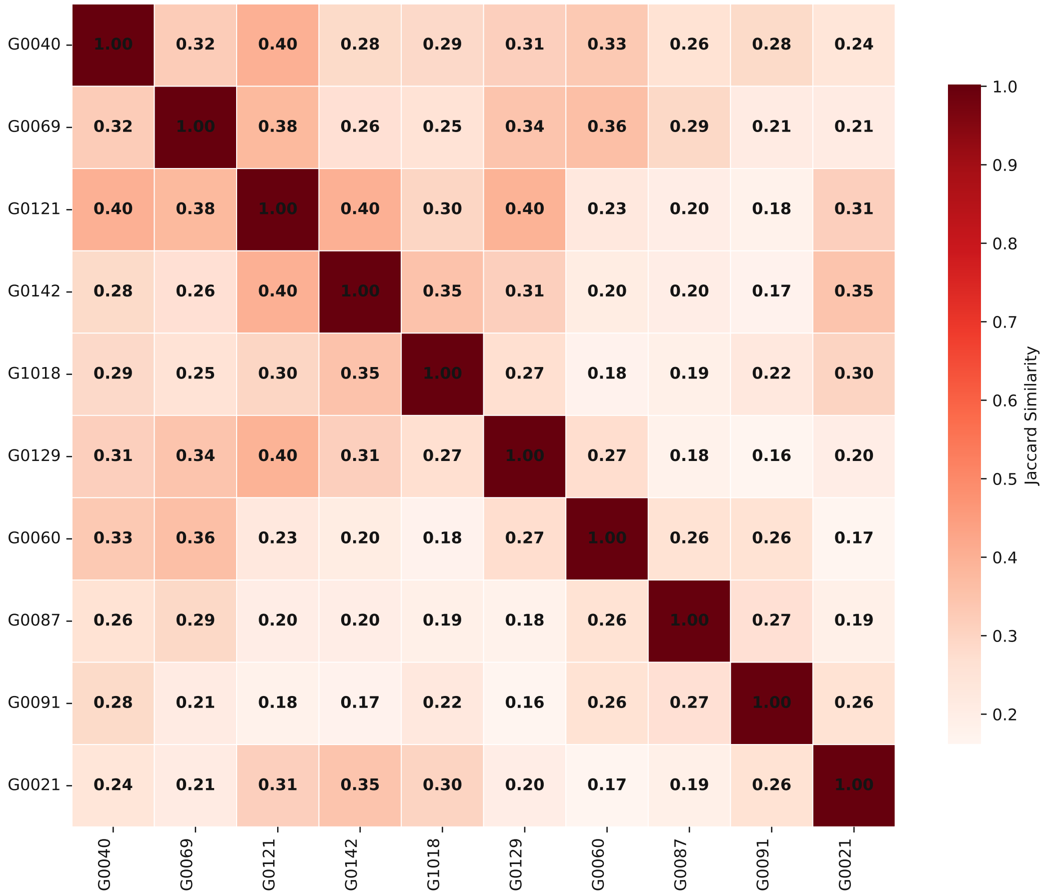

Figure 5.2: Jaccard Index between the 12 most similar groups (i.e., the ones with Jaccard Index ≥ 0.4). The mean Jaccard Index across all groups is 0.07, the median is 0.06, and the maximum Jaccard Index is 0.55.

be due to a large number of possible techniques and limited visibility or incomplete data available in ATT&CK. We identified only 12 pairs of groups with a Jaccard Index larger than or equal to 0.4.

Figure 5.2 shows a heatmap of these 12 group pairs. We observe that the moderate similarity between these pairs is often driven by generic techniques. For example, groups G0062 (TA459) and G0005 (APT12) each have five techniques in their profiles, sharing three, resulting in a Jaccard Index of 0.43. However, these three shared techniques are all generic techniques in Table 5.2: Exploitation for Client Execution (T1203), Spearphishing Attachment (T1566.001), and User Execution: Malicious File (T1204.002). This illustrates how generic techniques artificially increase group similarity.

Group-specific techniques. We call *group-specific* to the 147 (17.5%) techniques associated with a single group. Only 52 (34.2%) groups have group-specific techniques.

Table 5.3: Examples of group-specific techniques, some groups have multiple group-specific techniques.

Group Name	Technique ID	Technique Name
APT12	T1568.003	DNS Calculation
APT28	T1550.001	Application Access Token
	T1546.015	Component Object Model Hijacking
	T1001.001	Junk Data
	T1137.002	Office Test
	T1211	Exploitation for Defense Evasion
	T1498	Network Denial of Service
APT32	T1552.002	Credentials in Registry
	T1564.004	NTFS File Attributes
APT37	T1123	Audio Capture
APT38	T1562.003	Impair Command History Logging
	T1565.003	Runtime Data Manipulation
	T1565.001	Stored Data Manipulation
	T1565.002	Transmitted Data Manipulation
APT39	T1546.010	AppInit DLLs
	T1059.010	AutoHotKey & AutoIT
	T1056	Input Capture
APT41	T1596.005	Scan Databases
APT5	T1554	Compromise Host Software Binary
Axiom	T1563.002	RDP Hijacking
	T1001.002	Steganography
	T1553	Subvert Trust Controls
Chimera	T1110.004	Credential Stuffing
	T1556.001	Domain Controller Authentication
Cobalt Group	T1218.008	Odbcconf
DarkVishnya	T1200	Hardware Additions
Darkhotel	T1497	Virtualization/Sandbox Evasion

The mean number of group-specific techniques is 0.99. While most of the techniques are commonly shared among groups, a few stand out by using distinct techniques, with the maximum number of group-specific techniques used by any group being 16 for Windshift (G0112).

A key question is whether these group-specific techniques appear unique because of limited coverage in ATT&CK, or if they truly represent capabilities developed or exclusively adopted by a single group. Table 5.3 provides examples of group-specific techniques. Some of these group-specific techniques appear indeed quite unique to their respective groups. For example, APT12 is the only group using DNS Calculation (T1568.003), where adversaries perform calculations on addresses returned in DNS results to determine which port and IP address to use for command and control. Conversely, some group-specific techniques may not be truly unique to their groups. For example, APT28 is the only group associated with Network Denial of Service (T1498), a fairly common attack technique likely to be used by other groups at some point, suggesting this uniqueness may reflect limited coverage rather than actual exclusivity.

Table 5.4: Top generic software by number of groups using them, their type in ATT&CK, whether they are in Malpedia, and the number and percentage of groups using them.

ID	Name	ATT&CK Type	Malpedia	Groups
S0002	Mimikatz	Tool	✓	46 (5.7%)
S0029	PsExec	Tool	✗	31 (3.8%)
S0039	Net	Tool	✗	30 (3.7%)
S0154	Cobalt Strike	Malware	✓	26 (3.2%)
S0013	PlugX	Malware	✓	25 (3.1%)
S0363	Empire	Tool	✗	15 (1.8%)
S0012	PoisonIvy	Malware	✓	14 (1.7%)
S0100	ipconfig	Tool	✗	13 (1.6%)
S0097	Ping	Tool	✗	13 (1.6%)
S0349	LaZagne	Tool	✓	12 (1.5%)

Takeaway: Only 52 groups (34.2%) have group-specific techniques. However, other groups may still be distinguishable by unique technique combinations, as seen by the low mean Jaccard Index of 0.06. Under-reporting remains a concern, as only 53.7% of ATT&CK techniques are observed in group profiles, and some seemingly group-specific techniques may not be truly unique.

5.3.2 Software Profiles

In this section, we explore how uniquely the software used by each group identifies it (see the classification of software in 5.2.1). For each group, we build three *software profiles* using the sets of normalized software names associated to the group in each dataset, and their union.

We first examine each dataset separately. Of the 794 software in ATT&CK, 509 (64.1%) are associated to at least one group. For Malpedia, the fraction is significantly smaller, where out of 3,367 software only, 963 (28.6%) are associated to at least one group. Software not associated to groups typically corresponds to non-APT malware. For example, the Conficker worm [177] and the Babuk ransomware [176] each appear in both ATT&CK and Malpedia and are not associated to groups in either dataset. The lower ratio of software associated to groups in Malpedia is likely due to Malpedia's larger coverage of non-APT malware.

The fraction of groups with a non-empty software profile is also larger in ATT&CK where 138 out of 152 (90.8%) groups have associated software compared to 220 (27.5%) out of 800 groups in Malpedia (see Table 5.1 for the number of groups in each dataset). However, Malpedia contains 16.9 times more threat reports than ATT&CK, offering significantly more data for building software profiles. This difference arises because many smaller or lesser-known groups have few reports to support comprehensive profiling.

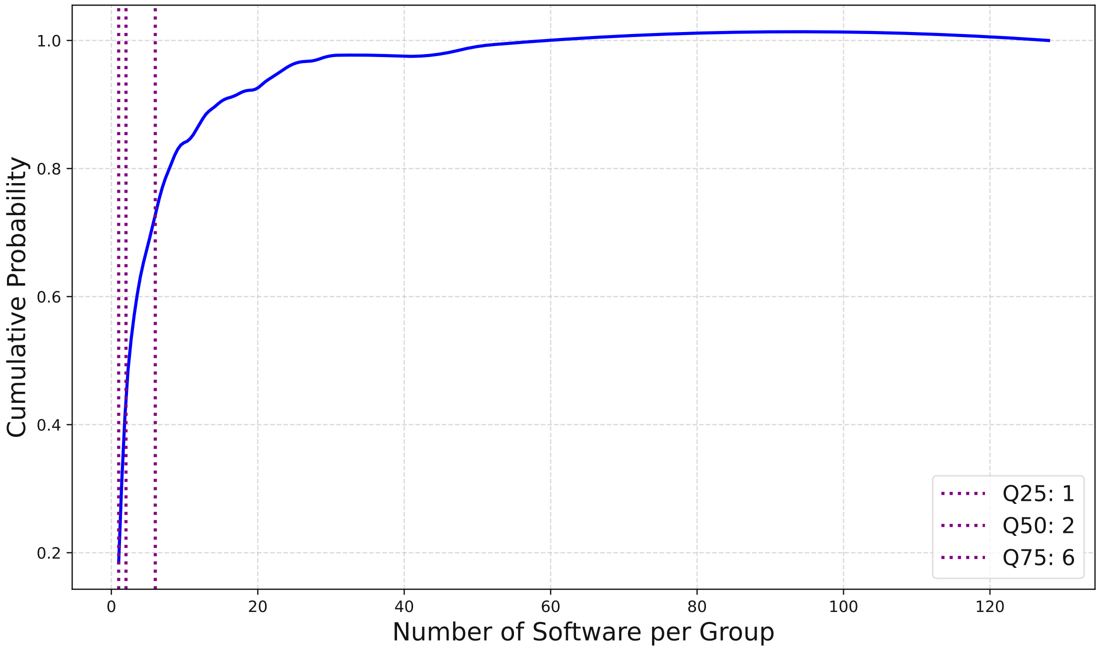

Figure 5.3: **CDF of the number of software per group.** 25% of groups used 1 or fewer software. 50% of groups used 2 or fewer software. Most groups used a relatively small number of software, with 75% using 6 or fewer.

Next, we examine the unified software profiles. Out of the total 807 groups in both datasets, 264 (32.7%) groups have a non-empty software profile. Thus, over two-thirds of the groups cannot be identified by their associated software. Figure 5.3 shows the CDF of software per group for groups with at least one software. On average, each group uses 6.2 software but most groups used a relatively small number of software with 25% of groups have only one associated software and 75% using 6 or fewer. The APT38 group (G0082) has the highest count with 120 software.

Generic software. We call the software used by many groups *generic*, as their detection offers limited value in distinguishing specific adversaries. Table 5.4 lists the top 10 software by number of groups where the software appears in the group's profile. All ten software appear in ATT&CK, while only five are present in Malpedia. The ones missing from Malpedia include four operating system tools (PsExec, Net, ipconfig, Ping) and the open-source Empire remote administration and post-exploitation framework [60]. Of these ten software entries, seven are classified as tools in ATT&CK and three as malware. Malpedia does not provide such classification. Among the three labeled as malware in ATT&CK, Cobalt Strike arguably should be categorized as a tool, as it is a commercial penetration testing package [68], while PlugX and PoisonIvy are remote administration tools (RATs) commonly available in underground markets. In summary, generic software typically refers to tools and malware kits that are either commercially sold or widely

Table 5.5: Summary of group profiles across different data sources and combinations. The table reports the number of groups with non-empty profiles and the subset with at least one group-specific behavior. The top section presents results for profiles built using only techniques, only software, and a combination of both, as analyzed in Section 3. The middle section shows results from profiles enriched with extracted CVE identifiers and additional techniques from downloaded threat reports, discussed in Section 4. The bottom row shows the most comprehensive profiles, combining all available behavioral indicators.

Profile	**ATT&CK**		**Malpedia**		**ATT&CK ∪ Malpedia**	
	Non-Empty	**w/ Group-Specific**	**Non-Empty**	**w/ Group-Specific**	**Non-Empty**	**w/ Group-Specific**
Techniques	148 (97.4%)	52 (34.2%)	–	–	148 (18.3%)	52 (6.4%)
Software	138 (90.8%)	111 (73.0%)	220 (27.5%)	192 (24.0%)	264 (32.7%)	213 (26.3%)
Techniques ∪ Software	151 (99.3%)	124 (81.6%)	220 (27.5%)	192 (24.0%)	331 (41.0%)	236 (29.2%)
Vulnerabilities	86 (56.6%)	48 (31.6%)	261 (32.6%)	112 (14.0%)	277 (34.3%)	119 (14.7%)
Techniques* (extracted)	149 (98.0%)	60 (39.5%)	204 (25.5%)	69 (8.6%)	242 (30.0%)	68 (8.4%)
Tech* ∪ Soft. ∪ Vuln.	152 (100%)	128 (84.2%)	391 (48.9%)	265 (33.1%)	418 (51.7%)	291 (36.0%)

accessible.

We identify 88 software tools that are not marked as “tools” in ATT&CK but are used by multiple groups. These likely correspond to publicly available malware kits that are sold or shared in underground forums. In addition to Cobalt Strike, PlugX, and PoisonIvy (Table 5.4), other commonly reused malware include the gh0st RAT (used by 10 groups), China Chopper web shell (8), 8.t dropper (8), njRAT (7), and ShadowPad (7). Of these, gh0st is open-source [71], njRAT’s source code was leaked [62], China Chopper is publicly available [35], and both 8.t and ShadowPad have been reported to be privately shared among Chinese threat groups [142, 233]. In some cases, the groups using the same software may be related. For instance, Bistromath is reported by Malpedia to be used by both Lazarus Group and Silent Chollima, the latter being the subsidiary of Lazarus [141].

Group-specific software. We term software as group-specific if it has only been associated to one group and is not classified as a tool in ATT&CK. We exclude tools even if associated with a single group, because they can be easily adopted by others in the future, thus providing weak attribution. Of the 3,663 software across both datasets, 952 (26.0%) are associated with a single group, making them *group-specific* software. The detection of group-specific software may allow attributing the group behind an attack. Of the 807 total groups, 213 (26.3%) have at least one group-specific software associated with them. Among these 213 groups, the mean and median number of group-specific software are 4.5 and 1.0, respectively, and 130 groups (16.1%) have only group-specific software. Some threat groups develop many custom tools for their attacks, for example, APT38 (G0082) has the highest number, with 99 group-specific software.

Takeaway: In ATT&CK, 111 groups (73.0%) have group-specific software, whereas in Malpedia, only 192 groups (24.0%) have that. Combining both datasets increases this number to 213 groups (26.3%), which still remains relatively low. Among the 264 groups with non-empty software profiles, the median number of software is 2, indicating that most threat groups operate with limited toolsets. However, some groups maintain extensive custom toolsets; for example, APT38 has 99 group-specific software.

5.3.3 Combining Group Profiles

Table 5.5 summarizes the number of groups with non-empty profiles and those with at least one group-specific entry across all the different profile-building methods examined in this work. The first three rows correspond to the profiles discussed in this section, based on techniques only, software only, and the combination of both.

To answer RQ1, the results show that identifying groups using techniques is challenging, as only 52 (34.2%) groups in ATT&CK have group-specific techniques. Identification is easier using software, with 111 (73.0%) of ATT&CK groups having group-specific software. However, this percentage is significantly lower in Malpedia, where only 192 (24.0%) of groups have group-specific software. The lower ratio in Malpedia likely reflects its focus on a subset of high-profile groups, while many smaller groups have too few reports to build robust profiles. Joint profiles combining both techniques and software improve identification in ATT&CK by 12.6 percentage points, increasing from 111 (73.0%) to 124 (81.6%) groups with group-specific behaviors.

To answer RQ2, the joint profiles built combining data from both datasets identify 236 (29.2%) groups with group-specific behaviors as compared to 124 groups using only ATT&CK and 192 using only Malpedia. Nonetheless, even when combining techniques and software, over 70% of groups do not have any group-specific behavior.

5.4 Extending the Group Profiles

So far, the group profiles have included techniques and software from both ATT&CK and Malpedia. In this section, we address RQ3, i.e., whether we can extend the group profiles with additional data extracted from the downloaded threat reports. In Section 5.4.1 we build a *vulnerability profile* for each group with the vulnerabilities that the threat reports refer to as being used by the group in its attacks. Then, in Section 5.4.2, we discuss how we extend the technique profiles with additional technique identifiers extracted from the threat reports. This allows incorporating techniques mentioned from the threat reports in Malpedia and to examine how complete the technique extraction was in ATT&CK.

Table 5.6: Summary of CVE and technique identifiers extracted from the downloaded threat reports. For each dataset, it presents: (i) the total number of reports analyzed, (ii) the number of reports containing at least one CVE identifier, (iii) the number of unique CVEs extracted, and (iv) the number of threat groups associated with those CVEs. It then provides similar statistics for the extraction of technique identifiers.

Dataset	Reports	Reports w/CVEs	CVEs	Groups	Reports w/Tech	Tech.	Groups
ATT&CK	920	266 (29.0%)	325	86	122 (13.2%)	470	63
Malpedia	4,414	943 (21.3%)	853	261	541 (12.2%)	626	204
All	5,827	1,186 (20.3%)	906	277	650 (11.1%)	658	211

5.4.1 Vulnerability Profiles

The selection of which vulnerabilities to exploit is largely group-specific since it depends on the expected software used by the targets and the exploits the group has access to. The observation of specific vulnerabilities being exploited in a monitored system could potentially be used to attribute the group behind an attack. To build the vulnerability profiles, we first extract CVE vulnerability identifiers from the downloaded threat reports using a regular expression provided by the iocsearcher open-source tool [33], which can extract IOCs from text, HTML, PDF, and Word files. Then, we assign the CVEs to groups. ATT&CK reports are associated with a single threat group, whereas Malpedia reports may reference multiple groups. When a report mentions multiple groups, it is unclear which group the CVEs within the report should be assigned to. Therefore, we extract CVEs only from the 4,414 (29.4%) Malpedia reports referencing a single group, along with all 920 ATT&CK reports.

The left part of Table 5.6 summarizes the extraction of CVE identifiers from the downloaded threat reports. Among the 5,827 reports analyzed, 1,186 (20.3%) contain at least one CVE identifier for a total of 906 unique CVEs associated to 277 groups. Of the 807 groups, 277 (34.3%) groups have a non-empty vulnerability profile. The other 530 (65.7%) groups have no vulnerabilities that can be used to identify them. The mean CVEs per group is 8.9, and the maximum is 176 CVEs reported for APT28 (G0007).

Generic vulnerabilities. Overall, there are 368 vulnerabilities used by at least two groups, 114 used by more than five groups, and 28 used by more than 10 groups. Table 5.7 lists the top 10 CVEs by the number of groups using them. These generic vulnerabilities target popular software, with Microsoft Office being the most targeted with four vulnerabilities. Nine of the 10 vulnerabilities have publicly available proof of concept (PoC) exploits, either in the Exploit Database [61] or on GitHub. We did not find any PoC for CVE-2022-38028, which was a zero-day on the Windows Print Spooler used by Russian threat groups [74].

Group-specific vulnerabilities. Of the 906 CVEs identified in the reports, 538 (59.4%) are associated to a single group. We call these *group-specific vulnerabilities*. The *Vulnerability* row in Table 5.5 summarizes the generated profiles. Of the 807 groups, 119 (14.7%) have at least one group-specific CVE. Across these 119 groups, the mean

Table 5.7: Top generic CVEs, i.e., used by most groups, and whether a PoC exploit is publicly available.

Vulnerability	Affected Software	PoC	Groups
CVE-2017-11882	Microsoft Office	✓	46 (5.7%)
CVE-2012-0158	Microsoft Office	✓	41 (5.1%)
CVE-2017-0199	Microsoft Office	✓	34 (4.2%)
CVE-2021-44228	Apache Log4j	✓	28 (3.5%)
CVE-2022-30190	Microsoft Windows (MSDT)	✓	25 (3.1%)
CVE-2022-26134	Atlassian Confluence	✓	21 (2.6%)
CVE-2018-0802	Microsoft Office	✓	20 (2.5%)
CVE-2022-38028	Windows Print Spooler	✗	17 (2.1%)
CVE-2023-38831	RARLAB WinRAR	✓	17 (2.1%)
CVE-2024-37085	VMware ESXi	✓	16 (2.0%)

Table 5.8: Examples of group-specific vulnerabilities, some groups have multiple group-specific vulnerabilities.

Group Name	Vulnerability	Affected Software
APT37	CVE-2015-3636	Linux Kernel
	CVE-2016-0147	MSXML
Akira	CVE-2019-6693	Fortinet FortiOS
	CVE-2023-29336	Microsoft Windows
	CVE-2023-35078	Ivanti Endpoint Manager
Kimsuky	CVE-2012-4873	GNU Board
	CVE-2018-14745	Samsung Galaxy
	CVE-2018-2628	Oracle WebLogic Server
Gorgon Group	CVE-2015-7036	Apple iOS
	CVE-2019-8457	SQLite3
	CVE-2019-8598	iOS, macOS
Sidewinder	CVE-2018-4876	Adobe Experience Manager
	CVE-2018-7445	MicroTik RouterOS
	CVE-2019-2215	Google Android
Scattered Spider	CVE-2015-2291	Ethernet driver on Windows
	CVE-2021-35464	ForgeRock Acess Management
	CVE-2022-0001	Intel Processors
Carbanak	CVE-2013-2463	Oracle JRE
	CVE-2015-2426	Microsoft Windows
	CVE-2016-1010	Adobe Flash Player

and median vulnerabilities per group are 4.5 and 2.0, respectively. The maximum is for the Gorgon group (G0078), which uses 78 CVEs. Table 5.8 shows some example group-specific CVEs.

Takeaway: Only 119 (14.7%) groups have a group-specific vulnerability. The set of vulnerabilities exploited by a group is less unique than the set of software used (27.8% groups have group-specific software) likely because many groups focus on the same generic vulnerabilities affecting popular software, often with publicly available exploits. However, vulnerabilities tend to be more unique than techniques, as only 6.4% groups have group-specific techniques.

5.4.2 Technique Identifiers in Reports

In this section, we extend the technique profiles with explicit mentions of technique identifiers in the downloaded reports. It is important to note that threat reports may also include implicit references to techniques, such as stating that a rootkit was used without explicitly providing a technique identifier. We discuss the extraction of implicit references later in this section. To identify technique identifiers, we use a regular expression provided by iocsearcher. We observe that the technique identifiers, if present, are typically provided in a table at the end of the threat report, although they may also appear throughout the text.

The right part of Table 5.6 summarizes the extraction of technique identifiers from the downloaded threat reports. From the 4,414 Malpedia reports uniquely assigned to one group, we find 626 unique technique identifiers associated to 204 groups appearing in 541 (12.2%) reports. Of these 626 techniques, 248 are not associated with any groups in ATT&CK, i.e., are only mentioned in the Malpedia reports. This shows that focusing on a small set of reports causes under-reporting and that technique profiles extracted from ATT&CK are likely to miss techniques used by a group.

Then, we apply iocsearcher to the 920 reports downloaded from ATT&CK reference URLs and identify 470 unique technique identifiers from 122 (13.2%) reports. These 122 reports are associated to 63 groups.

ATT&CK has 451 techniques associated to 152 groups, while we find a larger technique set (470) mentioned for 63 groups in 13.2% of the same reports. This indicates that ATT&CK contributors may not be systematic in extracting all references of techniques in the report. Furthermore, we are only accounting for explicit references through technique identifiers. Threat reports may also include implicit references. However, extracting implicit references to techniques would reinforce the under-reporting trends we already observe.

The *Techniques** row in Table 5.5 captures the techniques profiles extended with the technique identifiers extracted from the threat reports. It shows that we can identify 69 groups with group-specific techniques from the Malpedia threat reports, compensating for the lack of techniques in Malpedia. Notably, when we extract technique identifiers directly from the ATT&CK reports and combine them with the existing ATT&CK technique profiles, the number of groups with group-specific techniques increases from 52 (34.2%)

to 60 (39.5%), despite analyzing the same set of threat reports. This highlights that the manual extraction of techniques by ATT&CK contributors is not always optimal.

Takeaway: The extraction of technique identifiers from threat reports increases the number of groups with group-specific techniques from 52 (6.4%) to 119 (14.7%) mostly due to additional techniques in the Malpedia reports. The comparison of technique identifiers extracted from ATT&CK reports with the techniques indexed in ATT&CK shows that the extraction process used by ATT&CK contributors may miss techniques, suggesting the need for an automated approach.

Implicit reference extraction. Previous work has proposed NLP techniques for recovering the ATT&CK technique identifiers implicitly mentioned in threat reports [6, 92, 205]. An alternative approach would be to use Large Language Models (LLMs), which have proved their flexibility in a number of security-related tasks involving natural language texts [47, 236]. We performed some preliminary experiments using large language models (LLMs) to extract techniques from the downloaded threat reports. Specifically, we used the commercial GPT-4 model, guided by a prompt shown in Figure 5.4 in the Appendix, which we experimentally identified as the most effective among other options. To ensure the model analyzed the report content, we removed any tables of techniques included at the end of the report.

Unfortunately, we obtained mixed results as the LLM frequently hallucinated techniques, introducing false positives (FPs). An example is a 2022 report on the Lyceum group [135]. For this report, iocsearcher identified 8 techniques (without sub-techniques), all of them in a table at the end of the report. These are the same 8 techniques that ATT&CK associates to Lyceum from this report, suggesting that the contributor relied directly on the table for extraction. When we provided the same report after removing the table of identifiers, the LLM returned 11 technique identifiers (7 techniques and 4 sub-techniques). Of these, only two overlapped with the original table. We manually reviewed the remaining 9, finding that while two were valid, seven were false positives (FP). An example of a correctly extracted implicit reference is T1547.001 *Boot or Logon Autostart Execution: Startup Folder* (part of the *Persistence* tactic) that was extracted from the following text: “written into the Startup folder in order to maintain persistence”.

One example of a false positive is T1018, *Remote System Discovery*. When prompted to justify its inclusion, the model responded, “While the report does not specifically mention remote system discovery, many backdoors and malware engage in network reconnaissance, which aligns with T1018”. In future work, we would like to compare LLM-based extraction with existing extraction tools [6, 92, 205].

```
I have a detailed threat report written in natural language. I need
   help identifying the TTPs (Tactics, Techniques, and Procedures)
   described in the report and mapping them to their corresponding
   MITRE ATT&CK Technique IDs. The output should include:
A list of tactics (high-level strategic goals) based on the threat
   actor's behavior.
A list of techniques (specific actions or behaviors), with their
   corresponding MITRE ATT&CK Technique IDs.
A description of procedures (the exact implementation or variation of
    a technique as described in the report).
For each technique, provide both the name and the MITRE ATT&CK
   Technique ID. Please ensure all identified TTPs are clearly mapped
    to the most relevant MITRE ATT&CK entries.
Here's the threat report:
[Insert threat report text here]
The output should be in this JSON format:
Example Output:

{
  "tactics": ["Initial Access", "C2"],
  "techniques": [
    {"name": "Spear Phishing", "MITRE ID": "T1193"},
    {"name": "C2 Channel Over HTTPS", "MITRE ID": "T1071"}
  ],
  "procedures": ["Use of malicious scripts"]
}
```

Figure 5.4: LLM prompt: An example prompt engineered to instruct LLM (GPT-4) to identify Tactics, Techniques, and Procedures (TTPs) from a natural language threat report and map them to their respective MITRE ATT&CK Technique IDs, formatted for JSON output.

5.5 Discussion

5.5.1 Implications of Results

Our work critically challenges the widespread notion that behavioral profiles for threat actor attribution can replace IOCs. We show that behavioral profiles are not as distinctive as expected, with many groups employing generic techniques, software, and vulnerabilities also used by other groups. Only a small fraction of threat groups have group-specific behaviors that uniquely identify them and thus could be used as behavioral signatures. Roughly two-thirds (65.8%) of groups in ATT&CK have no group-specific techniques, challenging the distinctiveness of TTPs. The software used by a group is more distinctive with 73% groups in ATT&CK using group-specific programs. However, once we consider the larger number of groups in Malpedia the percentage drops to 26.3%. Despite leveraging information from ATT&CK and Malpedia and extending the profiles with the exploited vulnerabilities and additional techniques, the fraction of threat groups without unique behaviors remains at 64%.

As the number of profiled threat groups increases from 152 in ATT&CK to 800 in Malpedia, the fraction of groups with distinctive behaviors declines, as behaviors that initially looked unique may, with broader coverage, be observed across multiple groups. Consequently, even for the roughly one-third of groups exhibiting group-specific behaviors, a key question remains: are these behaviors genuinely unique, or do they appear so due to incomplete visibility into other groups? In fact, we observe cases where group-specific techniques are not inherently unique to a group (e.g., network DoS), but rather unassociated with other groups due to under-reporting. This raises concerns about the confidence of behavior-based attribution, which may impact critical attribution tasks such as those performed by legal or law enforcement, e.g., the attribution may not stand examination in a judicial process.

We observe that wide coverage is critical for constructing truly distinctive behavioral profiles. A key factor impacting coverage is the number of threat reports analyzed. Building profiles from small datasets of threat reports (e.g., those written by a single vendor or a select group of trusted sources) is tempting because those reports may be more uniform and less "noisy", and it is easier to find behaviors that initially look unique. However, such behavioral profiles provide little confidence. Given the limited overlap we observe between ATT&CK and Malpedia, relying on a few sources may limit coverage too much. We argue that it is better to index more threat reports (as Malpedia does) rather than focusing narrowly on very selected sources (as ATT&CK seems to do) because it is hard and error-prone to predict which sources are the most accurate. Furthermore, such filtering for trusted sources can always be performed a posteriori on the indexed data, as long as the mapping from behaviors to original sources is maintained (as the examined datasets do). Reports from known low-confidence sources can be excluded later in the pipeline. In contrast, assuming only a small set of sources is trustworthy may lead to overly constrained coverage. This is particularly problematic given that each vendor produces a limited number of reports annually and that reporting tends to concentrate on high-profile threat groups. Another important factor influencing coverage is the methodology used to extract behaviors from threat reports (e.g., manual or automated). We found that we could extract more explicit technique references from ATT&CK reports than those indexed by ATT&CK itself, even when analyzing the same reports. This suggests that the manual extraction process used by ATT&CK contributors may not always be exhaustive, further motivating the adoption of automated approaches [6, 92, 205].

Beyond techniques and software already indexed in the examined datasets, we have shown that behavioral profiles can be further extended with the exploited vulnerabilities. Additionally, other behavioral features present in the threat reports such as, payment services for adversaries to receive ransom, communication channels through which victims and adversaries interact, and the textual content that adversaries present to victims (e.g., emails, ransom notes).

Our analysis reveals a number of potential improvements to the datasets. First, we observe that the addition of OS-integrated tools (e.g., net, ping) into the ATT&CK software taxonomy provides little value, as those tools are already available in most target systems. Having a complete software taxonomy is not realistic; the focus should be on tools that adversaries deploy, which captures intent and provides more behavioral information. Second, the classification of software into tools and malware in ATT&CK is useful for analysts to quickly filter generic software, but is missing in Malpedia. Furthermore, it is not clear where malicious kits should be placed, possibly indicating the need for a third category. We also observe some likely misclassified software, e.g., Cobalt Strike is arguably a tool rather than malware. Finally, the split of techniques into domains used by ATT&CK seems quite arbitrary, as techniques may apply to different domains, albeit with different implementations. For example, there is a Rootkit (T1014) technique in the Enterprise domain and another Rootkit (T0851) technique in the ICS domain. The latter includes in the description references to firmware rootkits and Stuxnet but having two equally named techniques is confusing and likely unnecessary. There is also an overlap between techniques. For example, the Enterprise domain includes Pre-OS Boot: System Firmware (T1542.001) for capturing adversaries modifying system firmware to persist on systems, which seems the same as a firmware rootkit. Given an observation of a rootkit in a device, different security vendors and analysts may assign any combination of the above 3 techniques to the observation, which would complicate understanding which group may be behind the attack. This split complicates usage as three different technique taxonomies, one per domain, need to be considered. This makes it tempting to focus on the Enterprise domain, comprising 76%

5.5.2 Threats to Validity

We discuss some potential threats to the validity of our results. First, our reliance on open-source threat intelligence (OSINT) introduces potential selection bias. Commercial CTI feeds could offer more detailed analysis of some threat groups. However, such commercial feeds are often limited to data from a single provider, significantly restricting their overall coverage. Second, a small number of threat reports could not be successfully downloaded, potentially leading to an underestimation of dataset coverage. Nonetheless, fewer than 5% of the URLs resulted in an error. To mitigate this, future work could incorporate archival sources such as the Wayback Machine [95] and AptNotes [13].

Third, our focus on group-specific behaviors as behavioral signatures may overlook groups characterized by unique combinations of non-exclusive behaviors. Unfortunately, the more behaviors needed to identify a group, the greater the risk that an attack goes unattributed. Finally, inconsistencies in naming conventions across knowledge bases pose a challenge for accurate data comparison. Although we applied a normalization strategy to align group and software names, there remains a risk that some mappings are incorrect.

5.6 Related Work

Our research relates to the following prior CTI research.

Knowledge bases. Previous work has presented the design of the two knowledge bases we use [199, 243]. Other works have analyzed the usage of ATT&CK by systematically reviewing literature on its applications [101, 221, 224]. Oftentimes, works use the knowledge bases simply as a source of threat reports from where IOCs can be extracted [33, 104]. Our work differs in measuring the utility of ATT&CK and Malpedia for the specific case of adversary profiling.

Application-oriented studies. Several studies have examined the use of ATT&CK across different cybersecurity contexts. Oosthoek et al. [190] employed ATT&CK to map sandbox evasion techniques across 951 Windows malware families, offering insight into both commonly used and increasingly adopted techniques in recent years.

Virkud et al. [259] evaluate the ATT&CK framework in commercial endpoint detection products and assess its effectiveness as a security evaluation metric. They find that while these products typically cover between 48%–55% of ATT&CK techniques, much of this coverage consists of low-risk or less impactful rules. Their findings suggest that although ATT&CK is increasingly used to assess threat readiness, reported coverage frequently fails to reflect actual detection capabilities in real-world scenarios.

In another line of work, Rahman et al. [204] investigate challenges in implementing security controls (e.g., strong password policies) against ATT&CK techniques.

In simultaneous and independent work, yet to be presented, Horst et al. [86] examine the role of low-level IOCs (e.g., domains) and high-level IOCs (e.g., TTPs) in ransomware attribution. They use a mixed-methods approach, combining interviews of 15 ransomware attribution experts and analyzing 27 incident reports from two sources. They show that experts leverage low-level IOCs for attribution more frequently than high-level IOCs, which they regard as too generic. Our results match theirs in raising concerns about using behavioral traits for attribution. But our approaches are quite different. They examine 16 ransomware groups while we examine 807 threat groups covering different types of adversaries (e.g., APTs). We do not perform interviews but analyze over 15K threat reports from two popular knowledge bases. And, we measure for the first time the fraction of threat groups with unique behaviors.

Automated CTI extraction. Husari et al. [92] made early efforts to automate the extraction of TTPs from threat intelligence reports, using a context-aware, rule-based approach to identify and extract threat actions from both structured and unstructured CTI sources. The extracted TTPs are standardized using the STIX[179] format, with the tool achieving over 82% precision and recall on a proprietary dataset. Extending this work, Alam et al. [6] employed machine learning for automated extraction of attack patterns and IOCs. Their framework further mapped the extracted behaviors to the standardized ATT&CK framework and organized them in a knowledge graph to facilitate predictive analysis. Complementing these extraction-focused efforts, Rahman et al. [203] analyzed 667 CTI reports from the ATT&CK framework to study the prevalence and co-occurrence of TTPs used in APT campaigns, providing insights into adversary patterns. Our work builds upon these approaches by combining threat intelligence data from both the ATT&CK framework and Malpedia. We examine techniques and vulnerability usage across adversary groups, offering insights into building more comprehensive threat group profiles.

5.7 Conclusion

Our study critically evaluates the assumption that behavioral profiles can effectively replace Indicators of Compromise (IOCs) for threat actor attribution. By analyzing two open-source CTI knowledge bases, MITRE ATT&CK and Malpedia, we show significant limitations in the distinctiveness and completeness of group behavioral profiles. Specifically, only 34.2% of ATT&CK groups have group-specific techniques. Even after incorporating software and vulnerabilities from both ATT&CK and Malpedia, 64% of threat groups still lack unique behavioral signatures. As coverage expands from 152 groups in ATT&CK to 800 in Malpedia, the specificity of behaviors diminishes, with previously unique features proving to be more widespread. These findings highlight that group-specific behaviors are both rare and often overestimated.

CHAPTER 6

Conclusion and Future Work

Investigating and attributing advanced persistent threats remains a challenge at the intersection of technical complexity, operational workflow, and threat intelligence limitations. This thesis addresses the problem through complementary perspectives, including qualitative studies of practitioner workflows, machine-learning-based attribution with the ADAPT system, technical analysis of malicious documents, and large-scale empirical evaluation of CTI knowledge bases.

From a human perspective, our study of security practitioners revealed that attribution is not a monolithic task, but a layered decision process involving APT classification, TTP attribution, and, when feasible, country-level attribution. Practitioners focus on incident mitigation and threat prioritization over absolute actor identification. However, existing tools often fail to support their workflows, struggling with data heterogeneity, limited automation supporting diverse file formats, and poor internal and external collaboration. The study reinforces the need for attribution systems that are not only technically sound but also usable, interpretable, and aligned with real-world investigative practices.

To address these limitations, we developed ADAPT, a machine learning-based framework for campaign or TTP-level attribution across heterogeneous file types, including executables and documents. ADAPT uses clustering and static feature extraction to identify behavioral similarities, linking artifacts to campaigns and, when possible, to known threat groups. Practical case studies show that ADAPT supports analysts in navigating attribution complexity, reducing manual effort, and providing a foundation for automating parts of the attribution pipeline. Future directions include expanding ADAPT's dataset coverage, integrating network-based features, and improving clustering accuracy with dynamic analysis (e.g., sandbox-extracted IOCs only).

On the technical front, we showed that document-based malware, particularly those using Excel macros, obfuscation, and unconventional file formats such as RTF and OneNote, remains a prevalent yet understudied attack vector. Static analysis tools currently fail to reliably parse and extract meaningful indicators from malicious documents, especially in the presence of evasive Excel 4.0 macros. Future research should focus on advancing static analysis capabilities with grammar-aware parsers and large language models (LLMs) that can generalize to both standard and novel macro constructs.

Finally, we broaden our understanding of the APT landscape by evaluating open-source CTI knowledge bases, MITRE ATT&CK, and Malpedia, which reveal significant gaps in behavioral profiling. While indicators such as TTPs and software are often promoted as more stable than IOCs, our analysis shows that only a minority of groups have truly distinctive behaviors. Even after augmenting profiles with additional techniques and vulnerabilities, over 60% of groups remain indistinguishable. Furthermore, discrepancies between sources, underreporting, and inconsistent taxonomies hinder the reliability of behavior-based attribution. Future work should look into developing automated techniques, such as using LLMs to extract both explicit and implicit threat attack patterns, and unifying overlapping taxonomies across CTI platforms. These efforts enable reliable, scalable, and comprehensive threat group profiling from CTI data.

Toward a Unified Approach: Integrating Malware Behavior and Threat Intelligence for Attribution

The combined insights from this work, which include automated attribution using ADAPT, practitioner studies of attribution workflows, large-scale evaluations of document-based threats, and group profiles in CTI knowledge bases, highlight the need for a unified system for APT analysis and attribution. Such a system should integrate behavioral signals extracted directly from diverse malware artifacts with contextual intelligence derived from threat intelligence knowledge bases, enabling accurate and relevant decision-making. Future attribution systems should:

- Incorporate **human-centered design principles** to ensure that attribution tools align with the workflows, priorities, and decision-making processes of security practitioners. This includes understanding how analysts interpret attribution evidence and identifying which features provide actionable value during investigations.
- Leverage **scalable and interpretable machine learning techniques** to automate the clustering of diverse file types, including Android and iOS samples, and to provide robust tooling support for analysts. Future systems should integrate LLMs to extract implicit threat behaviors from unstructured CTI data and correlate these insights with malware code regions to expedite reverse engineering and analysis.
- Augment **behavioral data extracted from malware artifacts** with **enriched threat intelligence** to capture both low-level technical actions and high-level adversary context. Combining automated malware analysis with historical CTI data can improve attribution accuracy and support the development of comprehensive threat group profiles.

Overview of Generative AI Tools Used

AI-based coding assistants, particularly GitHub Copilot integrated within Visual Studio Code (VSCode), were used to assist with error handling, debugging, and generating graphs relevant to the research. These tools proved helpful during the experimental phases of the study. Additionally, generative AI tools for document editing, such as Grammarly and AI-powered LaTeX editors, were used to perform grammar checks, enhance formatting, and citation management throughout the manuscript. All generative tools were used critically, with outputs carefully reviewed to ensure accuracy, originality, and adherence to academic integrity standards.

Bibliography

[1] *7-Zip*. https://www.7-zip.org/. 2024.

[2] Hojjat Aghakhani, Fabio Gritti, Francesco Mecca, Martina Lindorfer, Stefano Ortolani, Davide Balzarotti, Giovanni Vigna, and Christopher Kruegel. "When Malware is Packing Heat; Limits of Machine Learning Classifiers based on Static Analysis Features". In: *Proceedings of the 27th Network and Distributed System Security Symposium (NDSS)*. 2020.

[3] Mansour Ahmadi, Dmitry Ulyanov, Stanislav Semenov, Mikhail Trofimov, and Giorgio Giacinto. "Novel Feature Extraction, Selection and Fusion for Effective Malware Family Classification". In: *Proceedings of the 6th ACM Conference on Data and Application Security and Privacy (CODASPY)*. 2016.

[4] Olusola Akinrolabu, Ioannis Agrafiotis, and Arnau Erola. "The Challenge of Detecting Sophisticated Attacks: Insights from SOC Analysts". In: *Proceedings of the 13th International Conference on Availability, Reliability and Security (ARES)*. 2018.

[5] Bushra A. Alahmadi, Louise Axon, and Ivan Martinovic. "99% False Positives: A Qualitative Study of SOC Analysts' Perspectives on Security Alarms". In: *Proceedings of the 31st USENIX Security Symposium (USENIX Sec)*. 2022.

[6] Md. Tanvirul Alam, Dipkamal Bhusal, Youngja Park, and Nidhi Rastogi. "Looking beyond IoCs: Automatically Extracting Attack Patterns from External CTI". In: *Proceedings of the 26th International Symposium on Research in Attacks, Intrusions and Defenses (RAID)*. 2023.

[7] *A Global Perspective of the SideWinder APT*. https://cdn-cybersecurity.att.com/docs/global-perspective-of-the-sidewinder-apt.pdf. 2021.

[8] Saed Alrabaee, Paria Shirani, Mourad Debbabi, and Lingyu Wang. "On the Feasibility of Malware Authorship Attribution". In: *Proceedings of 9th International Symposium on Foundations and Practice of Security (FPS)*. 2016.

[9] Hyrum S. Anderson and Phil Roth. "EMBER: An Open Dataset for Training Static PE Malware Machine Learning Models". In: *arXiv preprint arXiv:1804.04637* (2018).

[10] Md. Monowar Anjum, Shahrear Iqbal, and Benoit Hamelin. "ANUBIS: A Provenance Graph-Based Framework for Advanced Persistent Threat Detection". In: *Proceedings of the 37th ACM/SIGAPP Symposium on Applied Computing (SAC)*. 2022.

[11] John Annett. *Hierarchical Task Analysis*. CRC Press, 2003.

[12] Simone Aonzo, Yufei Han, Alessandro Mantovani, and Davide Balzarotti. "Humans vs. Machines in Malware Classification". In: *Proceedings of the 32nd USENIX Security Symposium (USENIX Sec)*. 2023.

[13] *APTnotes repository*. https://github.com/aptnotes/data/. 2025.

[14] Ionut Arghire. *Malicious RTF Persistently Asks Users to Enable Macros*. https://web.archive.org/web/20240516132100/https://www.securityweek.com/malicious-rtf-persistently-asks-users-enable-macros/. 2018.

[15] Daniel Arp, Erwin Quiring, Feargus Pendlebury, Alexander Warnecke, Fabio Pierazzi, Christian Wressnegger, Lorenzo Cavallaro, and Konrad Rieck. "Dos and don'ts of machine learning in computer security". In: *Proceedings of the 31st USENIX Security Symposium (USENIX Sec)*. 2022.

[16] Mohammed Asiri, Neetesh Saxena, Rigel Gjomemo, and Pete Burnap. "Understanding Indicators of Compromise against cyber-attacks in Industrial Control Systems: A Security Perspective". In: *ACM Transactions on Cyber-Physical Systems* 7.2 (2023).

[17] Priyanka Badva, Kopo M. Ramokapane, Eleonora Pantano, and Awais Rashid. "Unveiling the Hunter-Gatherers: Exploring Threat Hunting Practices and Challenges in Cyber Defense". In: *Proceedings of the 33rd USENIX Security Symposium (USENIX Sec)*. 2024.

[18] Michael Barnhart, Austin Larsen, Jeff Johnson, Taylor Long, Michelle Cantos, and Adrian Hernandez. *Assessed Cyber Structure and Alignments of North Korea in 2023*. https://www.mandiant.com/resources/blog/north-korea-cyber-structure-alignment-2023. 2023.

[19] Brian Bartholomew and Juan Andres Guerrero-Saade. "Wave Your False Flags! Deception Tactics Muddying Attribution in Targeted Attacks". In: *Virus Bulletin Conference*. 2016.

[20] Ulrich Bayer, Paolo Milani Comparetti, Clemens Hlauschek, Christopher Kruegel, and Engin Kirda. "Scalable, Behavior-based Malware Clustering". In: *Proceedings of the 16th Network and Distributed System Security Symposium (NDSS)*. 2009.

[21] Ruth Bearden and Dan Chai-Tien Lo. "Automated Microsoft Office Macro Malware Detection Using Machine Learning". In: *Proceedings of the IEEE International Conference on Big Data (Big Data)*. 2017.

[22] David Bianco. *The Pyramid of Pain*. https://detect-respond.blogspot.com/2013/03/the-pyramid-of-pain.html. 2014.

[23] *Binary Ninja*. https://binary.ninja/. 2024.

[24] *Bitcoin Address*. https://www.blockchain.com/explorer/addresses/btc/1QLDYEyeo8c6CFHdcEB5yBjQw6pcRiTdN5. 2016.

[25] Frank Boldewin. *OfficeMalScanner*. https://github.com/fboldewin/reconstructer.org. 2019.

[26] Marcus Botacin. “What do Malware Analysts want from Academia? A Survey on the State-of-the-practice to Guide Research Developments”. In: *Proceedings of the 27th International Symposium on Research in Attacks, Intrusions and Defenses (RAID)*. 2024.

[27] Marcus Botacin, Hojjat Aghakhani, Stefano Ortolani, Christopher Kruegel, Giovanni Vigna, Daniela Oliveira, Paulo Lício De Geus, and André Grégio. “One Size Does Not Fit All: A Longitudinal Analysis of Brazilian Financial Malware”. In: *ACM Transactions on Privacy and Security (TOPS)* (2021).

[28] Xander Bouwman, Harm Griffioen, Jelle Egbers, Christian Doerr, Bram Klievink, and Michel Van Eeten. “A Different Cup of TI? The Added Value of Commercial Threat Intelligence”. In: *Proceedings of the 29th USENIX Security Symposium (USENIX Sec)*. 2020.

[29] Xander Bouwman, Victor Le Pochat, Pawel Foremski, Tom Van Goethem, Carlos H. Ganan, Giovane C. M. Moura, Samaneh Tajalizadehkhoob, Wouter Joosen, and Michel van Eeten. “Helping Hands: Measuring the Impact of a Large Threat Intelligence Sharing Community”. In: *Proceedings of the 31st USENIX Security Symposium (USENIX Sec)*. 2022.

[30] Guillaume Brogi and Valerie Viet Triem Tong. “TerminAPTor: Highlighting Advanced Persistent Threats through Information Flow Tracking”. In: *Proceedings of the 8th IFIP International Conference on New Technologies, Mobility and Security (NTMS)*. 2016.

[31] Rebekah Brown and Pasquale Stirparo. *SANS 2022 Cyber Threat Intelligence Survey*. https://www.sans.org/white-papers/sans-2022-cyber-threat-intelligence-survey/. 2022.

[32] Rufus Brown, Van Ta, Douglas Bienstock, Geoff Ackerman, and John Wolfram. *Does This Look Infected? A Summary of APT41 Targeting U.S. State Governments*. https://www.mandiant.com/resources/blog/apt41-us-state-governments. 2022.

[33] Juan Caballero, Gibran Gomez, Srdjan Matic, Gustavo Sánchez, Silvia Sebastián, and Arturo Villacañas. “The Rise of GoodFATR: A Novel Accuracy Comparison Methodology for Indicator Extraction Tools”. In: *Future Generation Computer Systems* 144 (2023).

[34] Aylin Caliskan-Islam, Richard Harang, Andrew Liu, Arvind Narayanan, Clare Voss, Fabian Yamaguchi, and Rachel Greenstadt. “De-anonymizing Programmers via Code Stylometry”. In: *Proceedings of the 24th USENIX Security Symposium (USENIX Sec)*. 2015.

[35] Canadian Center for Cyber Security. *Joint Report on Publicly Available Hacking Tools*. https://www.cyber.gc.ca/sites/default/files/cyber/publications/joint_report_on_publicly_available_hacking_tools.pdf. 2018.

[36] Steve Canny. *python-docx (v1.1.0)*. https://pypi.org/project/python-docx/. 2023.

[37] Fran Casino, Nikolaos Totosis, Theodoros Apostolopoulos, Nikolaos Lykousas, and Constantinos Patsakis. “Analysis and Correlation of Visual Evidence in Campaigns of Malicious Office Documents”. In: *Digital Threats: Research and Practice (DTRAP)* (2023).

[38] Microsoft Security Response Center. *Tracking the Cross-Domain Solorigate Attack from Endpoint to the Cloud*. https://www.microsoft.com/en-us/security/blog/2020/12/28/using-microsoft-365-defender-to-coordinate-protection-against-solorigate/. 2020.

[39] Tencent Security Threat Intelligence Center. *Cyber Warfare in the Shadow of the India-Pakistan War - A Summary of Recent Indo-Pakistani APT Attack Activities*. https://mp.weixin.qq.com/s/pJ-rnzB7VMZ0feM2X0ZrHA. 2019.

[40] Fabrício Ceschin, Marcus Botacin, Albert Bifet, Bernhard Pfahringer, Luiz S. Oliveira, Heitor Murilo Gomes, and André Grégio. “Machine Learning (In) Security: A Stream of Problems”. In: *Digital Threats: Research and Practice (DTRAP)* (2024).

[41] *Emotet Malware*. https://www.cisa.gov/news-events/cybersecurity-advisories/aa20-280a. 2020.

[42] CISA. *AppleJeus: JMT Trading*. https://www.cisa.gov/news-events/cybersecurity-advisories/aa21-048a. 2021.

[43] Aviad Cohen, Nir Nissim, Lior Rokach, and Yuval Elovici. “SFEM: Structural Feature Extraction Methodology for the Detection of Malicious Office Documents Using Machine Learning Methods”. In: *Expert Systems with Applications* (2016).

[44] Eric Cole. *Advanced Persistent Threat: Understanding the Danger and How to Protect your Organization*. Syngress Publishing, 2012.

[45] Juliet Corbin and Anselm Strauss. *Basics of Qualitative Research: Techniques and Procedures for Developing Grounded Theory*. Sage Publications, 2014.

[46] Emanuele Cozzi, Mariano Graziano, Yanick Fratantonio, and Davide Balzarotti. “Understanding Linux Malware”. In: *Proceedings of the 39th IEEE Symposium on Security & Privacy (S&P)*. 2018.

[47] Hoang Cuong Nguyen, Shahroz Tariq, Mohan Baruwal Chhetri, and Bao Quoc Vo. "Towards Effective Identification of Attack Techniques in Cyber Threat Intelligence Reports using Large Language Models". In: *Companion Proceedings of the ACM on Web Conference (WWW)*. 2025.

[48] *Microsoft Word Security Vulnerabilities*. https://www.cvedetails.com/vulnerability-list/vendor_id-26/product_id-529/Microsoft-Word.html. 2024.

[49] *APT Malware Dataset*. https://github.com/cyber-research/APTMalware. 2019.

[50] *Sidewinder APT Targets with Futuristic Tactics and Techniques*. https://blog.cyble.com/2020/09/26/sidewinder-apt-targets-with-futuristic-tactics-and-techniques/. 2020.

[51] *DoNot Team APT Updates its Malware Arsenal*. https://cyware.com/news/donot-team-apt-updates-its-malware-arsenal-a5a76e92. 2020.

[52] Mike Czumak. *Abusing Microsoft Office DDE*. https://web.archive.org/web/20240515161640/https://www.securitysift.com/abusing-microsoft-office-dde/. 2017.

[53] Constanze Dietrich, Katharina Krombholz, Kevin Borgolte, and Tobias Fiebig. "Investigating System Operators' Perspective on Security Misconfigurations". In: *Proceedings of the 25th ACM SIGSAC Conference on Computer and Communications Security (CCS)*. 2018.

[54] Ditekshen. *Detection Yara*. https://github.com/ditekshen/detection/tree/master/yara. 2024.

[55] *Royal Road Is Still in Use!* https://web.archive.org/web/20240515155837/https://www.docguard.io/royal-road-malware-rtf/. 2023.

[56] *DPRK Hacking Indictment*. https://www.justice.gov/d9/press-releases/attachments/2021/02/17/dprk_hacking_-_indictment_1_0.pdf. 2021.

[57] Zakir Durumeric, David Adrian, Ariana Mirian, Michael Bailey, and J. Alex Halderman. "A Search Engine Backed by Internet-Wide Scanning". In: *Proceedings of the 22nd ACM SIGSAC Conference on Computer and Communications Security (CCS)*. 2015.

[58] *ECMA-376 – Office Open XML File Formats*. https://ecma-international.org/publications-and-standards/standards/ecma-376/. 2021.

[59] *Elastic Security Detection Content for Endpoint*. https://github.com/elastic/protections-artifacts. 2024.

[60] *Empire Post-exploitation Framework*. https://github.com/EmpireProject/Empire. 2025.

[61] *Exploit Database*. https://www.exploit-db.com/. 2025.

[62] Facundo Muñoz. *njRAT: A Remote Access Trojan Widely Used by Various Cybercriminals*. https://www.welivesecurity.com/la-es/2021/09/29/que-es-njrat-troyano-acceso-remoto-utilizado-cibercriminales/. 2021.

[63] Mohamed Fahmy and Mahmoud Zohdy. *APT 34 Deploys Phishing Attack with New Malware*. https://web.archive.org/web/20240517103343/https://www.trendmicro.com/en_us/research/23/i/apt34-deploys-phishing-attack-with-new-malware.html. 2023.

[64] *APT 41*. https://www.fbi.gov/wanted/cyber/apt-41-group. 2020.

[65] *APT 10*. https://www.fbi.gov/wanted/cyber/apt-10-group. 2018.

[66] *Word Processing File Formats - RTF*. https://docs.fileformat.com/word-processing/rtf/. 2023.

[67] Nicole Fishbein. *How to Analyze Malicious Microsoft Office Files*. https://intezer.com/blog/malware-analysis/analyze-malicious-microsoft-office-files/. 2023.

[68] FORTRA. *Cobal Strike - Adversary Simulations and Red Team Operations*. https://www.cobaltstrike.com/. 2025.

[69] Yanick Fratantonio, Elie Bursztein, Luca Invernizzi, Marina Zhang, Giancarlo Metitieri, Thomas Kurt, Francois Galilee, Alexandre Petit-Bianco, and Ange Albertini. *Magika Content-type Scanner*. https://github.com/google/magika. 2023.

[70] *Malpedia*. https://malpedia.caad.fkie.fraunhofer.de/. 2025.

[71] *gh0st RAT*. https://github.com/sin5678/gh0st. 2025.

[72] Ibrahim Ghafir, Mohammad Hammoudeh, Vaclav Prenosil, Liangxiu Han, Robert Hegarty, Khaled Rabie, and Francisco J Aparicio-Navarro. "Detection of Advanced Persistent Threat using Machine-Learning Correlation Analysis". In: *Future Generation Computer Systems* 89 (2018).

[73] John R. Goodall, Wayne G. Lutters, and Anita Komlodi. "I Know my Network: Collaboration and Expertise in Intrusion Detection". In: *Proceedings of the 17th ACM Conference on Computer Supported Cooperative Work (CSCW)*. 2004.

[74] Dan Goodin. *Windows Vulnerability Reported by the NSA Exploited to Install Russian Malware*. https://arstechnica.com/security/2024/04/kremlin-backed-hackers-exploit-critical-windows-vulnerability-reported-by-the-nsa/. 2024.

[75] Jason Gray, Daniele Sgandurra, Lorenzo Cavallaro, and Jorge Blasco. "Identifying Authorship in Malicious Binaries: Features, Challenges & Datasets". In: *ACM Computing Surveys (CSUR)* 56.8 (2024).

[76] Jonathan Greig. *CISA Warns of Attacks using Microsoft Word, Adobe Bugs*. https://web.archive.org/web/20240519175623/https://therecord.media/microsoft-adobe-bugs-cisa-kev-list. 2023.

[77] Harm Griffioen, Tim Booij, and Christian Doerr. "Quality Evaluation of Cyber Threat Intelligence Feeds". In: *Proceedings of the 18th International Conference of Applied Cryptography and Network Security (ACNS)*. 2020.

[78] Greg Guest, Arwen Bunce, and Laura Johnson. "How Many Interviews are Enough? An Experiment with Data Saturation and Variability". In: *Field Methods* 18.1 (2006).

[79] Weijie Han, Jingfeng Xue, Yong Wang, Fuquan Zhang, and Xianwei Gao. "APT-MalInsight: Identify and Cognize APT Malware Based on System Call Information and Ontology Knowledge Framework". In: *Information Sciences* 546 (2021).

[80] Xueyuan Han, Thomas Pasquier, Adam Bates, James Mickens, and Margo Seltzer. "UNICORN: Runtime Provenance-Based Detector for Advanced Persistent Threats". In: *Proceedings of the 27th Network and Distributed System Security Symposium (NDSS)*. 2020.

[81] Irfan Ul Haq, Sergio Chica, Juan Caballero, and Somesh Jha. "Malware Lineage in the Wild". In: *Computers & Security* 78 (2018).

[82] Phil Harvey. *Exiftool (v12.4)*. https://exiftool.org/. 2022.

[83] Mehadi Hassen, Marco M. Carvalho, and Philip K. Chan. "Malware Classification using Static Analysis-based Features". In: *Proceedings of the IEEE Symposium Series on Computational Intelligence (SSCI)*. 2017.

[84] *The Evolution of Sidewinder APT and their Modus Operandi*. https://www.hawk-eye.io/2022/12/the-evolution-of-sidewinder-apt-and-their-modus-operandi/. 2022.

[85] Andrew F Hayes and Klaus Krippendorff. "Answering the Call for a Standard Reliability Measure for Coding Data". In: *Communication Methods and Measures* 1.1 (2007).

[86] Max van der Horst, Ricky Kho, Olga Gadyatskaya, and Michel Mollema. "High Stakes, Low Certainty: Evaluating the Efficacy of High-Level Indicators of Compromise in Ransomware Attribution". In: *Proceedings of the 34th USENIX Security Symposium (USENIX Sec)*. 2025.

[87] Md Nahid Hossain, Sanaz Sheikhi, and R. Sekar. "Combating Dependence Explosion in Forensic Analysis using Alternative Tag Propagation Semantics". In: *Proceedings of the 41st IEEE Symposium on Security & Privacy (S&P)*. 2020.

[88] Zheng Hu, Jiaojiao Zhang, and Yun Ge. "Handling Vanishing Gradient Problem using Artificial Derivative". In: *IEEE Access* 9 (2021).

[89] *multi-qa-mpnet-base-dot-v1*. https://huggingface.co/sentence-transformers/multi-qa-mpnet-base-dot-v1. 2022.

[90] *all-MiniLM-L12-v2*. https://huggingface.co/sentence-transformers/all-MiniLM-L12-v2. 2022.

[91] Adam Hupp. *python-magic (v0.4.27)*. https://pypi.org/project/python-magic/. 2022.

[92] Ghaith Husari, Ehab Al-Shaer, Mohiuddin Ahmed, Bill Chu, and Xi Niu. “TTP-Drill: Automatic and Accurate Extraction of Threat Actions from Unstructured Text of CTI Sources”. In: *Proceedings of the 33rd Annual Computer Security Applications Conference (ACSAC)*. 2017.

[93] *IDA Pro*. https://hex-rays.com/ida-pro/. 2024.

[94] *yara-rules-vt*. https://github.com/InQuest/yara-rules-vt. 2024.

[95] Internet Archive. *Wayback Machine*. https://web.archive.org/. 2025.

[96] *ISO/IEC 29500-1:2016 - Office Open XML File Formats*. https://www.iso.org/standard/71691.html. 2016.

[97] Anil K Jain, M Narasimha Murty, and Patrick J Flynn. “Data Clustering: A Review”. In: *ACM Computing Surveys (CSUR)* 31.3 (1999).

[98] Jamie. *zloader and XLM 4.0: Making Evasion Great Again*. https://web.archive.org/web/20240515190020/https://clickallthethings.wordpress.com/2020/05/13/zloader-and-xlm-4-0-making-evasion-great-again/. 2020.

[99] Jiyong Jang, Maverick Woo, and David Brumley. “Towards Automatic Software Lineage Inference”. In: *Proceedings of the 22nd USENIX Security Symposium (USENIX Sec)*. 2013.

[100] Zian Jia, Yun Xiong, Yuhong Nan, Yao Zhang, Jinjing Zhao, and Mi Wen. “MAGIC: Detecting Advanced Persistent Threats via Masked Graph Representation Learning”. In: *Proceedings of the 32nd USENIX Security Symposium (USENIX Sec)*. 2023.

[101] Yuning Jiang, Qiaoran Meng, Feiyang Shang, Nay Oo, Le Thi Hong Minh, Hoon Wei Lim, and Biplab Sikdar. “MITRE ATT&CK Applications in Cybersecurity and The Way Forward”. In: *arXiv preprint arXiv:2502.10825* (2025).

[102] Beomjin Jin, Eunsoo Kim, Hyunwoo Lee, Elisa Bertino, Doowon Kim, and Hyoungshick Kim. “Sharing Cyber Threat Intelligence: Does it Really Help?” In: *Proceedings of the 31st Network and Distributed System Security Symposium (NDSS)*. 2024.

[103] *JoeSandbox*. https://joesandbox.com/. 2024.

[104] Robert J Joyce, Dev Amlani, Charles Nicholas, and Edward Raff. “Motif: A Malware Reference Dataset with Ground Truth Family Labels”. In: *Computers & Security* 124 (2023).

[105] Gerhard Jungwirth, Aakanksha Saha, Michael Schröder, Tobias Fiebig, Martina Lindorfer, and Jürgen Cito. “Connecting the .dotfiles: Checked-In Secret Exposure with Extra (Lateral Movement) Steps”. In: *Proceedings of the 20th International Conference on Mining Software Repositories (MSR)*. 2023.

[106] Vaibhavi Kalgutkar, Natalia Stakhanova, Paul Cook, and Alina Matyukhina. “Android Authorship Attribution through String Analysis”. In: *Proceedings of the 13th International Conference on Availability, Reliability and Security (ARES)*. 2018.

[107] Anthony Kasza and Dominik Reiche. *The Gamaredon Group Toolset Evolution*. https://unit42.paloaltonetworks.com/unit-42-title-gamaredon-group-toolset-evolution/. 2017.

[108] Sangwoo Kim, Seokmyung Hong, Jaesang Oh, and Heejo Lee. “Obfuscated VBA Macro Detection using Machine Learning”. In: *Proceedings of the 48th International Conference on Dependable Systems and Networks (DSN)*. 2018.

[109] Faris Bugra Kokulu, Ananta Soneji, Tiffany Bao, Yan Shoshitaishvili, Ziming Zhao, Adam Doupé, and Gail-Joon Ahn. “Matched and Mismatched SOCs: A Qualitative Study on Security Operations Center Issues”. In: *Proceedings of the 26th ACM SIGSAC Conference on Computer and Communications Security (CCS)*. 2019.

[110] Jesse Kornblum. “Identifying Almost Identical Files using Context Triggered Piecewise Hashing”. In: *Digital Investigation* 3 (2006).

[111] Vasilios Koutsokostas, Nikolaos Lykousas, Theodoros Apostolopoulos, Gabriele Orazi, Amrita Ghosal, Fran Casino, Mauro Conti, and Constantinos Patsakis. “Invoice# 31415 Attached: Automated Analysis of Malicious Microsoft Office Document”. In: *Computers & Security* (2022).

[112] Marc Kührer, Christian Rossow, and Thorsten Holz. “Paint it Black: Evaluating the Effectiveness of Malware Blacklists”. In: *Proceedings of the 17th International Symposium on Research in Attacks, Intrusions and Defenses (RAID)*. 2014.

[113] David Kushner. *The Real Story of Stuxnet*. https://spectrum.ieee.org/the-real-story-of-stuxnet. 2013.

[114] Philippe Lagadec. *Anti-Analysis Tricks in Weaponized RTF*. http://decalage.info/rtf_tricks. 2016.

[115] Philippe Lagadec. *Tools to Extract VBA Macro Source Code from MS Office Documents*. https://www.decalage.info/en/vba_tools. 2017.

[116] Philippe Lagadec. *msodde*. https://github.com/decalage2/oletools/blob/master/oletools/msodde.py. 2019.

[117] Philippe Lagadec. *oletools (v0.60.1)*. https://github.com/decalage2/oletools. 2022.

[118] Selena Larson, Daniel Blackford, and Garrett G. *The First Step: Initial Access Leads to Ransomware*. https://www.proofpoint.com/us/blog/threat-insight/first-step-initial-access-leads-ransomware. 2021.

[119] Pavel Laskov and Nedim Šrndić. “Static Detection of Malicious JavaScript-bearing PDF Documents”. In: *Proceedings of the 27th Annual Computer Security Applications Conference (ACSAC)*. 2011.

[120] Giuseppe Laurenza and Riccardo Lazzeretti. “ dAPTaset: A Comprehensive Mapping of APT-Related Data”. In: *Proceedings of the International Workshop on Security for Financial Critical Infrastructures and Services (FINSEC)*. 2019.

[121] Robert Layton and Ahmad Azab. “Authorship Analysis of the Zeus Botnet Source Code”. In: *Proceedings of the 5th Cybercrime and Trustworthy Computing Conference (CTC)*. 2014.

[122] Stevens Le Blond, Cédric Gilbert, Utkarsh Upadhyay, Manuel Gomez-Rodriguez, and David R. Choffnes. “A Broad View of the Ecosystem of Socially Engineered Exploit Documents”. In: *Proceedings of the 24th Network & Distributed System Security Symposium (NDSS)*. 2017.

[123] Clement Lecigne and Maddie Stone. *Active North Korean Campaign Targeting Security Researchers*. https://blog.google/threat-analysis-group/active-north-korean-campaign-targeting-security-researchers/. 2023.

[124] *RTF Files*. https://web.archive.org/web/20240516081339/https://www.lenovo.com/us/en/glossary/rtf/. 2024.

[125] Shaofei Li, Feng Dong, Xusheng Xiao, Haoyu Wang, Fei Shao, Jiedong Chen, Yao Guo, Xiangqun Chen, and Ding Li. “NODLINK: An Online System for Fine-Grained APT Attack Detection and Investigation”. In: *Proceedings of the 31st Network and Distributed System Security Symposium (NDSS)*. 2024.

[126] Vector Guo Li, Matthew Dunn, Paul Pearce, Damon McCoy, Geoffrey M. Voelker, and Stefan Savage. “Reading the Tea Leaves: A Comparative Analysis of Threat Intelligence”. In: *Proceedings of the 28th USENIX Security Symposium (USENIX Sec)*. 2019.

[127] Xiaojing Liao, Kan Yuan, XiaoFeng Wang, Zhou Li, Luyi Xing, and Raheem Beyah. “Acing the IoC Game: Toward Automatic Discovery and Analysis of Open-Source Cyber Threat Intelligence”. In: *Proceedings of the 23rd ACM SIGSAC Conference on Computer and Communications Security (CCS)*. 2016.

[128] *Microsoft Office Word 97-2003 Binary File Format (.doc)*. https://www.loc.gov/preservation/digital/formats/fdd/fdd000509.shtml. 2023.

[129] Kevin van Liebergen, Gibran Gomez, Srdjan Matic, and Juan Caballero. “All your (data)base are belong to us: Characterizing Database Ransom(ware) Attacks”. In: *Proceedings of Network and Distributed Systems Security Symposium (NDSS)*. 2025.

[130] Martina Lindorfer, Alessandro Di Federico, Federico Maggi, Paolo Milani Comparetti, and Stefano Zanero. “Lines of Malicious Code: Insights Into the Malicious Software Industry”. In: *Proceedings of the 28th Annual Computer Security Applications Conference (ACSAC)*. 2012.

[131] Martina Lindorfer, Clemens Kolbitsch, and Paolo Milani Comparetti. "Detecting Environment-Sensitive Malware". In: *Proceedings of the 14th International Symposium on Recent Advances in Intrusion Detection (RAID)*. 2011.

[132] Martina Lindorfer, Bernhard Miller, Matthias Neugschwandtner, and Christian Platzer. "Take a Bite - Finding the Worm in the Apple". In: *Proceedings of the 9th International Conference on Information, Communications and Signal Processing (ICICS)*. 2013.

[133] Daiping Liu, Haining Wang, and Angelos Stavrou. "Detecting Malicious Javascript in PDF through Document Instrumentation". In: *Proceedings of the 44th International Conference on Dependable Systems and Networks (DSN)*. 2014.

[134] Matthew Lombard, Jennifer Snyder-Duch, and Cheryl Campanella Bracken. "Content Analysis in Mass Communication: Assessment and Reporting of Intercoder Reliability". In: *Human Communication Research* 28.4 (2002).

[135] *Lyceum .NET DNS Backdoor*. https://www.zscaler.com/blogs/security-research/lyceum-net-dns-backdoor. 2022.

[136] Tommy Madjar, Corsin Camichel, Joe Wise, Selena Larson, and Chris Talib. *OneNote Documents Increasingly used to Deliver Malware*. https://www.proofpoint.com/us/blog/threat-insight/onenote-documents-increasingly-used-to-deliver-malware. 2023.

[137] Moustafa Mahmoud, Mohammad Mannan, and Amr Youssef. "APTHunter: Detecting Advanced Persistent Threats in Early Stages". In: *Digital Threats: Research and Practice* 4 (2022).

[138] *Malcat Binary Analysis Software (version 0.8.3)*. https://malcat.fr/. 2024.

[139] Dean Malmgren. *textract (v1.6.4)*. https://textract.readthedocs.io/en/stable/. 2017.

[140] *Lazarus Group*. https://malpedia.caad.fkie.fraunhofer.de/actor/lazarus_group. 2024.

[141] *Silent Chollima*. https://malpedia.caad.fkie.fraunhofer.de/actor/silent_chollima. 2025.

[142] MalwareLab. *On the Royal Road*. https://blog.malwarelab.pl/posts/on_the_royal_road/. 2020.

[143] *Assembling the Russian Nesting Doll: UNC2452 Merged into APT29*. https://www.mandiant.com/resources/blog/unc2452-merged-into-apt29. 2022.

[144] *APT1: Exposing One of China's Cyber Espionage Units*. https://www.mandiant.com/resources/apt1-exposing-one-of-chinas-cyber-espionage-units. 2021.

[145] *Uncategorized (UNC) Threat Groups*. https://www.mandiant.com/resources/insights/uncategorized-unc-threat-groups. 2024.

[146] *Advanced Persistent Threats (APTs).* https://www.mandiant.com/resources/insights/apt-groups. 2023.

[147] *FLOSS - FLARE Obfuscated String Solver (v2.2.0).* https://github.com/mandiant/flare-floss. 2023.

[148] *Supply Chain Analysis: From Quartermaster to Sunshop.* https://www.mandiant.com/resources/supply-chain-analysis-from-quartermaster-to-sunshop. 2022.

[149] *APT42: Crooked Charms, Cons and Compromises.* https://www.mandiant.com/media/17826. 2022.

[150] Alessandro Mantovani, Simone Aonzo, Yanick Fratantonio, and Davide Balzarotti. "RE-Mind: A First Look inside the Mind of a Reverse Engineer". In: *Proceedings of the 31st USENIX Security Symposium (USENIX Sec)*. 2022.

[151] Alessandro Mantovani, Simone Aonzo, Xabier Ugarte-Pedrero, Alessio Merlo, and Davide Balzarotti. "Prevalence and Impact of Low-Entropy Packing Schemes in the Malware Ecosystem". In: *Proceedings of the 27th Network and Distributed System Security Symposium (NDSS)*. 2020.

[152] Morgan Marquis-Boire, Marion Marschalek, and Claudio Guarnieri. "Big Game Hunting: The Peculiarities in Nation-State Malware Research". In: *BlackHat USA*. 2015.

[153] James Mattei, Madeline McLaughlin, Samantha Katcher, and Daniel Votipka. "A Qualitative Evaluation of Reverse Engineering Tool Usability". In: *Proceedings of the 38th Annual Computer Security Applications Conference (ACSAC)*. 2022.

[154] William P Maxam III and James C Davis. "An Interview Study on Third-Party Cyber Threat Hunting Processes in the US Department of Homeland Security". In: *Proceedings of the 33rd USENIX Security Symposium (USENIX Sec)*. 2024.

[155] *MAXQDA.* https://www.maxqda.com/automatic-transcription. 2024.

[156] Francesco Meloni, Alessandro Sanna, Davide Maiorca, and Giorgio Giacinto. "Effective Call Graph Fingerprinting for the Analysis and Classification of Windows Malware". In: *Proceedings of 19th International Conference on Detection of Intrusions and Malware, and Vulnerability Assessment (DIMVA)*. 2022.

[157] Leigh Metcalf and Jonathan M. Spring. "Blacklist Ecosystem Analysis". In: *Proceedings of the 2nd ACM Workshop on Information Sharing and Collaborative Security*. 2015.

[158] *Manage Exclusions for Microsoft Defender.* https://learn.microsoft.com/en-us/microsoft-365/security/defender-endpoint/defender-endpoint-antivirus-exclusions. 2023.

[159] *New Feature in Office 2016 can Block Macros and Help Prevent Infection.* https://www.microsoft.com/en-us/security/blog/2016/03/22/new-feature-in-office-2016-can-block-macros-and-help-prevent-infection/. 2016.

[160] *Office VBA Reference.* https://learn.microsoft.com/en-us/office/vba/api/overview/#vba-programming-in-office. 2021.

[161] *Hide Sheets and use the xlVeryHidden Constant in a Macro.* https://learn.microsoft.com/en-us/office/troubleshoot/excel/hide-sheet-and-use-xlveryhidden. 2022.

[162] *HAFNIUM targeting Exchange Servers with 0-day exploits.* https://www.microsoft.com/en-us/security/blog/2021/03/02/hafnium-targeting-exchange-servers/. 2021.

[163] *Microsoft Shifts to a New Threat Actor Naming Taxonomy.* https://www.microsoft.com/en-us/security/blog/2023/04/18/microsoft-shifts-to-a-new-threat-actor-naming-taxonomy/. 2023-04.

[164] Sadegh M Milajerdi, Rigel Gjomemo, Birhanu Eshete, Ramachandran Sekar, and V.N. Venkatakrishnan. “Holmes: Real-Time APT Detection through Correlation of Suspicious Information Flows”. In: *Proceedings of the 40th IEEE Symposium on Security & Privacy (S&P).* 2019.

[165] Mamoru Mimura. “Using Fake Text Vectors to Improve the Sensitivity of Minority Class for Macro Malware Detection”. In: *Journal of Information Security and Applications* (2020).

[166] Mamoru Mimura and Hiroya Miura. “Detecting Unseen Malicious VBA Macros with NLP Techniques”. In: *Journal of Information Processing* (2019).

[167] Mamoru Mimura and Taro Ohminami. “Towards Efficient Detection of Malicious VBA Macros with LSI”. In: *Proceedings of the 14th International Workshop on Security (IWSEC).* 2019.

[168] Jaron Mink, Hadjer Benkraouda, Limin Yang, Arridhana Ciptadi, Ali Ahmadzadeh, Daniel Votipka, and Gang Wang. “Everybody’s Got ML, Tell Me What Else You Have: Practitioners’ Perception of ML-Based Security Tools and Explanations”. In: *Proceedings of the 44th IEEE Symposium on Security & Privacy (S&P).* 2023.

[169] Omid Mirzaei, Roman Vasilenko, Engin Kirda, Long Lu, and Amin Kharraz. “Scrutinizer: Detecting Code Reuse in Malware via Decompilation and Machine Learning”. In: *Proceedings of the 18th Detection of Intrusions and Malware, and Vulnerability Assessment (DIMVA).* 2021.

[170] *MISP Threat Actors Galaxy.* https://github.com/MISP/misp-galaxy/blob/main/clusters/threat-actor.json. 2025.

[171] *MITRE Groups.* https://attack.mitre.org/groups/. 2023.

[172] *MITRE Campaigns.* https://attack.mitre.org/campaigns/. 2022.

[173] *Sidewinder*. https://attack.mitre.org/groups/G0121/. 2020.

[174] *Transparent Tribe*. https://attack.mitre.org/groups/G0134/. 2021.

[175] *MITRE ATT&CK*. https://attack.mitre.org/. 2025.

[176] *Babuk*. https://attack.mitre.org/software/S0638/. 2025.

[177] *Conficker*. https://attack.mitre.org/software/S0608/. 2025.

[178] *SimBad*. https://attack.mitre.org/software/S0419/. 2025.

[179] *MITRE ATT&CK STIX Data*. https://github.com/mitre-attack/attack-stix-data. 2024.

[180] Jens Müller, Fabian Ising, Christian Mainka, Vladislav Mladenov, Sebastian Schinzel, and Jörg Schwenk. "Office Document Security and Privacy". In: *Proceedings of the 14th USENIX Workshop on Offensive Technologies (WOOT)*. 2020.

[181] Aleksandr Nahapetyan, Sathvik Prasad, Kevin Childs, Adam Oest, Yeganeh Ladwig, Alexandros Kapravelos, and Brad Reaves. "On SMS Phishing Tactics and Infrastructure". In: *Proceedings of IEEE Security & Privacy Symposium (S&P)*. 2024.

[182] *Advisory: APT29 Targets COVID-19 Vaccine Development*. https://media.defense.gov/2020/Jul/16/2002457639/-1/-1/0/NCSC_APT29_ADVISORY-QUAD-OFFICIAL-20200709-1810.PDF. 2020.

[183] Amirreza Niakanlahiji and Pedram Amini. *Extracting "Sneaky" Excel XLM Macros*. https://inquest.net/blog/extracting-sneaky-excel-xlm-macros/. 2019.

[184] Amirreza Niakanlahiji and Pedram Amini. *Getting Sneakier: Hidden Sheets, Data Connections, and XLM Macros*. https://inquest.net/blog/getting-sneakier-hidden-sheets-data-connections-and-xlm-macros/. 2020.

[185] *Advanced Persistent Threats*. https://csrc.nist.gov/topics/security-and-privacy/risk-management/threats/advanced-persistent-threats. 2024.

[186] *Threat Actor*. https://csrc.nist.gov/glossary/term/threat_actor. 2024.

[187] *Threat Scenario*. https://csrc.nist.gov/glossary/term/threat_scenario. 2024.

[188] *CVE-2017-11882*. https://nvd.nist.gov/vuln/detail/CVE-2017-11882. 2017.

[189] Sean Oesch, Robert Bridges, Jared Smith, Justin Beaver, John Goodall, Kelly Huffer, Craig Miles, and Dan Scofield. "An Assessment of the Usability of Machine Learning Based Tools for the Security Operations Center". In: *Proceedings of the International Conferences 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) and IEEE Congress on Cybermatics (Cybermatics)*. 2020.

[190] Kris Oosthoek and Christian Doerr. "SoK: ATT&CK Techniques and Trends in Windows Malware". In: *International Conference on Security and Privacy in Communication Systems (SecureComm)*. 2019.

[191] Yuhei Otsubo, Mamoru Mimura, and Hidehiko Tanaka. "O-Checker: Detection of Malicious Documents Through Deviation from File Format Specifications". In: *BlackHat USA* (2016).

[192] *LevelBlue Open Threat Exchange*. https://otx.alienvault.com/dashboard/new. 2025.

[193] Outflank. *Old School: Evil Excel 4.0 Macros (XLM)*. https://web.archive.org/web/20240515191958/https://www.outflank.nl/blog/2018/10/06/old-school-evil-excel-4-0-macros-xlm/. 2018.

[194] Pierluigi Paganini. *Russia-Linked APT 28 Group Observed using DDE Attack to Deliver Malware*. https://web.archive.org/web/20240511190941/https://securityaffairs.com/65318/hacking/dde-attack-apt28.html. 2017.

[195] Seongsu Park and Vitaly Kamluk. *The BlueNoroff Cryptocurrency Hunt is Still on*. https://web.archive.org/web/20240519163632/https://securelist.com/the-bluenoroff-cryptocurrency-hunt-is-still-on/105488/. 2022.

[196] Fabian Pedregosa, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, Jake Vanderplas, Alexandre Passos, David Cournapeau, Matthieu Brucher, Matthieu Perrot, and Édouard Duchesnay. "Scikit-learn: Machine Learning in Python". In: *Journal of Machine Learning Research* 12 (2011).

[197] Roberto Perdisci, Wenke Lee, and Nick Feamster. "Behavioral Clustering of Http-Based Malware and Signature Generation using Malicious Network Traces". In: *Proceedings of the 7th USENIX Conference on Networked Systems Design and Implementation (NSDI)*. 2010.

[198] Avi Pfeffer, Catherine Call, John Chamberlain, Lee Kellogg, Jacob Ouellette, Terry Patten, Greg Zacharias, Arun Lakhotia, Suresh Golconda, John Bay, Robert Hall, and Daniel Scofield. "Malware Analysis and Attribution using Genetic Information". In: *Proceedings of the 7th International Conference on Malicious and Unwanted Software (MALWARE)*. 2012.

[199] Daniel Plohmann, Martin Clauss, Steffen Enders, and Elmar Padilla. “Malpedia: A Collaborative Effort to Inventorize the Malware Landscape”. In: *The Journal on Cybercrime & Digital Investigations* 3.1 (2018).

[200] Emilee Rader, Samantha Hautea, and Anjali Munasinghe. ““I Have a Narrow Thought Process”: Constraints on Explanations Connecting Inferences and Self-Perceptions”. In: *Proceedings of the 16th Symposium On Usable Privacy and Security (SOUPS)*. 2020.

[201] Michael Raggi. *Chinese APT TA413 Resumes Targeting of Tibet Following COVID-19 Themed Economic Espionage Campaign Delivering Sepulcher Malware Targeting Europe*. https://www.proofpoint.com/us/blog/threat-insight/chinese-apt-ta413-resumes-targeting-tibet-following-covid-19-themed-economic. 2020.

[202] Michael Raggi. *Injection is the New Black: Novel RTF Template Inject Technique Poised for Widespread Adoption Beyond APT Actors*. https://www.proofpoint.com/uk/blog/threat-insight/injection-new-black-novel-rtf-template-inject-technique-poised-widespread. 2021.

[203] Md Rayhanur Rahman, Setu Kumar Basak, Rezvan Mahdavi Hezaveh, and Laurie Williams. “Attackers Reveal their Arsenal: An Investigation of Adversarial Techniques in CTI Reports”. In: *arXiv preprint arXiv:2401.01865* (2024).

[204] Md Rayhanur Rahman, Brandon Wroblewski, Mahzabin Tamanna, Imranur Rahman, Andrew Anufryienak, and Laurie Williams. “Towards a Taxonomy of Challenges in Security Control Implementation”. In: *Proceedings of the 40th Annual Computer Security Applications Conference (ACSAC)*. 2024.

[205] Nanda Rani, Bikash Saha, Vikas Maurya, and Sandeep Kumar Shukla. “TTPX-Hunter: Actionable Threat Intelligence Extraction as TTPs from Finished Cyber Threat Reports”. In: *Digital Threats: Research and Practice (DTRAP)* 5.4 (2024).

[206] Chris Ray. *Intro to ImpHash for DFIR: “Fuzzy” Malware Matching*. https://www.cybertriage.com/blog/intro-to-imphash-for-dfir-fuzzy-malware-matching/. 2024.

[207] *A Consumer Cybersecurity Trends Report*. https://reasonlabs.com/research/consumer-cybersecurity-trends-report-2023. 2023.

[208] *Triage*. https://tria.ge/. 2024.

[209] RedxorBlue. *Executing Macros from a DOCX with Remote Template Injection*. https://web.archive.org/web/20240515180936/https://blog.redxorblue.com/2018/07/executing-macros-from-docx-with-remote.html. 2018.

[210] Nils Reimers and Iryna Gurevych. “Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks”. In: *Proceedings of the 9th International Joint Conference on Natural Language Processing (IJCNLP)*. 2019.

[211] Yitong Ren, Yanjun Xiao, Yinghai Zhou, Zhiyong Zhang, and Zhihong Tian. “CSKG4APT: A Cybersecurity Knowledge Graph for Advanced Persistent Threat Organization Attribution”. In: *IEEE Transactions on Knowledge and Data Engineering* 35 (2022).

[212] *SolarWinds Hack was Largest and Most Sophisticated Attack Ever: Microsoft President*. https://www.reuters.com/article/us-cyber-solarwinds-microsoft-idUSKBN2AF03R. 2021.

[213] *Rewterz Threat Alert – SideWinder APT Group aka Rattlesnake – Active IOCs*. https://www.rewterz.com/rewterz-news/rewterz-threat-alert-sidewinder-apt-group-aka-rattlesnake-active-iocs-2/. 2024.

[214] Thomas Rid and Ben Buchanan. “Attributing Cyber Attacks”. In: *Journal of Strategic Studies* 38.1-2 (2015).

[215] Konrad Rieck, Thorsten Holz, Carsten Willems, Patrick Düssel, and Pavel Laskov. “Learning and Classification of Malware Behavior”. In: *Proceedings of the 5th International Conference on Detection of Intrusions and Malware, and Vulnerability Assessment (DIMVA)*. 2008.

[216] Md Omar Faruk Rokon, Risul Islam, Ahmad Darki, Evangelos E Papalexakis, and Michalis Faloutsos. “SourceFinder: Finding Malware Source-Code from Publicly Available Repositories in GitHub”. In: *Proceedings of the 23rd International Symposium on Research in Attacks, Intrusions and Defenses (RAID)*. 2020.

[217] Ishai Rosenberg, Guillaume Sicard, and Eli Omid David. “DeepAPT: Nation-State APT Attribution Using End-to-End Deep Neural Networks”. In: *Proceedings of the 26th International Conference on Artificial Neural Networks (ICANN)*. 2017.

[218] Nathan Rosenblum, Xiaojin Zhu, and Barton P Miller. “Who Wrote this Code? Identifying the Authors of Program Binaries”. In: *Proceedings of the 16th European Symposium on Research in Computer Security (ESORICS)*. 2011.

[219] Florian Roth. *The Newcomer’s Guide to Cyber Threat Actor Naming*. https://cyb3rops.medium.com/the-newcomers-guide-to-cyber-threat-actor-naming-7428e18ee263. 2018.

[220] Florian Roth. *Signature-Base*. https://github.com/Neo23x0/signature-base/tree/master/yara. 2021.

[221] Shanto Roy, Emmanouil Panaousis, Cameron Noakes, Aron Laszka, Sakshyam Panda, and George Loukas. “SoK: The MITRE ATT&CK Framework in Research and Practice”. In: *arXiv preprint arXiv:2304.07411* (2023).

[222] Nicola Ruaro, Fabio Pagani, Stefano Ortolani, Christopher Kruegel, and Giovanni Vigna. “SYMBEXCEL: Automated Analysis and Understanding of Malicious Excel 4.0 Macros”. In: *Proceedings of the 43rd IEEE Security & Privacy Symposium (S&P)*. 2022.

[223] Vinay Sachidananda, Rajendra Patil, Akshay Sachdeva, Kwok-Yan Lam, and Liu Yang. "APTer: Towards the Investigation of APT Attribution". In: *Proceedings of the 6th IEEE Conference on Dependable and Secure Computing (DSC)*. 2023.

[224] Bader Al-Sada, Alireza Sadighian, and Gabriele Oligeri. "MITRE ATT&CK: State of the Art and Way Forward". In: *ACM Computing Surveys* 57.1 (2024).

[225] Aakanksha Saha, Jorge Blasco, Lorenzo Cavallaro, and Martina Lindorfer. "ADAPT it! Automating APT Campaign and Group Attribution by Leveraging and Linking Heterogeneous Files". In: *Proceedings of the 27th International Symposium on Research in Attacks, Intrusions and Defenses (RAID)*. 2024.

[226] Aakanksha Saha, Jorge Blasco, and Martina Lindorfer. "Exploring the Malicious Document Threat Landscape: Towards a Systematic Approach to Detection and Analysis". In: *Proceedings of the 3rd Workshop on Rethinking Malware Analysis (WoRMA)*. 2024.

[227] Aakanksha Saha, Martina Lindorfer, and Juan Caballero. "From IOCs to Group Profiles: On the Specificity of Threat Group Behaviors in CTI Knowledge Bases". In: *Proceedings of the 41st Annual Computer Security Applications Conference (ACSAC) (under review)*. 2025.

[228] Aakanksha Saha, James Mattei, Jorge Blasco, Lorenzo Cavallaro, Daniel Votipka, and Martina Lindorfer. "Expert Insights into Advanced Persistent Threats: Analysis, Attribution, and Challenges". In: *Proceedings of the 34th USENIX Security Symposium (USENIX Sec)*. 2025.

[229] *Hierarchical Clustering*. https://scikit-learn.org/stable/modules/clustering.html#hierarchical-clustering. 2024.

[230] *Silhouette Score*. https://scikit-learn.org/stable/modules/generated/sklearn.metrics.silhouette_score.html. 2024.

[231] Alex Scroxton. *IT Leaders Fear 'Trickle-Down' of Nation-State Cyber Attacks*. https://web.archive.org/web/20240515181415/https://www.computerweekly.com/news/252505571/IT-leaders-fear-trickle-down-of-nation-state-cyber-attacks. 2021.

[232] Silvia Sebastián and Juan Caballero. "AVclass2: Massive Malware Tag Extraction from AV Labels". In: *Proceedings of the 36th Annual Computer Security Applications Conference (ACSAC)*. 2020.

[233] Sentinel Labs. *ShadowPad: A Masterpiece of Privately Sold Malware RAT*. https://assets.sentinelone.com/c/shadowpad?x=p42eqa. 2021.

[234] Chintan Shah. *An Inside Look into Microsoft Rich Text Format and OLE Exploits*. https://web.archive.org/web/20240517151249/https://www.mcafee.com/blogs/other-blogs/mcafee-labs/an-inside-look-into-microsoft-rich-text-format-and-ole-exploits/. 2020.

[235] Chintan Shah. *The Tale of Two Exploits - Breaking Down CVE-2023-36884 and the Infection Chain*. https://www.trellix.com/about/newsroom/stories/research/breaking-down-cve-2023-36884-and-the-infection-chain/. 2023.

[236] Sayuj Shah and Vijay K Madisetti. “MAD-CTI: Cyber Threat Intelligence Analysis of the Dark Web Using a Multi-Agent Framework”. In: *IEEE Access* 13 (2025).

[237] Krzysztof Siwek and Stanislaw Osowski. “Autoencoder versus PCA in Face Recognition”. In: *Proceedings of the 18th International Conference on Computational Problems of Electrical Engineering (CPEE)*. 2017.

[238] Florian Skopik and Timea Pahi. “Under False Flag: Using Technical Artifacts for Cyber Attack Attribution”. In: *Cybersecurity* 3 (2020).

[239] Charles Smutz and Angelos Stavrou. “Malicious PDF Detection using Metadata and Structural Features”. In: *Proceedings of the 28th Annual Computer Security Applications Conference (ACSAC)*. 2012.

[240] Nedim Šrndic and Pavel Laskov. “Detection of Malicious PDF Files based on Hierarchical Document Structure”. In: *Proceedings of the 20th Network & Distributed System Security Symposium (NDSS)*. 2013.

[241] Nedim Šrndić and Pavel Laskov. “Hidost: A Static Machine-Learning-Based Detector of Malicious Files”. In: *EURASIP Journal on Information Security* (2016).

[242] Etienne Stalmans and Saif El-Sherei. *Macro-less Code Exec in MS Word*. https://sensepost.com/blog/2017/macro-less-code-exec-in-msword/. 2017.

[243] Lake E. Strom, Andy Applebaum, Doug P. Miller, Kathryn C. Nickels, Adam G. Pennington, and Cody B Thomas. *MITRE ATT&CK: Design and Philosophy*. https://attack.mitre.org/docs/ATTACK_Design_and_Philosophy_March_2020.pdf. 2018.

[244] Ting Su and Jennifer Dy. “A Deterministic Method for Initializing K-means Clustering”. In: *Proceedings of the 16th IEEE International Conference on Tools with Artificial Intelligence (ICTAI)*. 2004.

[245] Sathya Chandran Sundaramurthy, Alexandru G Bardas, Jacob Case, Xinming Ou, Michael Wesch, John McHugh, and S Raj Rajagopalan. “A Human Capital Model for Mitigating Security Analyst Burnout”. In: *Proceedings of the 11th Symposium On Usable Privacy and Security (SOUPS)*. 2015.

[246] Vanja Svajcer and Vitor Ventura. *What’s with the shared VBA code between Transparent Tribe and other threat actors?* https://blog.talosintelligence.com/2022/02/whats-with-shared-vba-code.html. 2022.

[247] *Fact Sheet: Imposing Costs for Harmful Foreign Activities by the Russian Government.* https://www.whitehouse.gov/briefing-room/statements-releases/2021/04/15/fact-sheet-imposing-costs-for-harmful-foreign-activities-by-the-russian-government/. 2021.

[248] Romain Thomas. *LIEF - Library to Instrument Executable Formats (version 0.13.1).* https://lief.quarkslab.com/. 2024.

[249] *ThreatMiner: Data Mining for Threat Intelligence.* https://www.threatminer.org/. 2025.

[250] *SideWinder APT Targets Nepal, Afghanistan in Wide-Ranging Spy Campaign.* https://threatpost.com/sidewinder-apt-nepal-afghanistan-spy-campaign/162086/. 2020.

[251] Shusei Tomonaga. *Malware "WellMess" Targeting Linux and Windows.* https://blogs.jpcert.or.jp/en/2018/07/malware-wellmes-9b78.html. 2018.

[252] Umut Tosun. *How to Analyze RTF Template Injection Attacks.* https://www.letsdefend.io/blog/how-to-analyze-rtf-template-injection-attacks. 2022.

[253] Wiem Tounsi and Helmi Rais. "A Survey on Technical Threat Intelligence in the Age of Sophisticated Cyber Attacks". In: *Computers & Security* 72 (2018).

[254] *Trellix Insights: FireEye Red Team Tools Stolen in Cyber Attack.* https://kcm.trellix.com/corporate/index?page=content&id=KB93880. 2022.

[255] *Spear-Phishing Email: Most Favored APT Attack Bait.* https://documents.trendmicro.com/assets/wp/wp-spear-phishing-email-most-favored-apt-attack-bait.pdf. 2012.

[256] *SideWinder Uses South Asian Issues for Spear Phishing, Mobile Attacks.* https://www.trendmicro.com/de_de/research/20/l/sidewinder-leverages-south-asian-territorial-issues-for-spear-ph.html. 2020.

[257] Unit42. *Russia's Gamaredon aka Primitive Bear APT Group Actively Targeting Ukraine.* https://unit42.paloaltonetworks.com/gamaredon-primitive-bear-ukraine-update-2021/. 2022.

[258] Kelli Vanderlee. *DebUNCing Attribution: How Mandiant Tracks Uncategorized Threat Actors.* https://www.mandiant.com/resources/blog/how-mandiant-tracks-uncategorized-threat-actors. 2020.

[259] Apurva Virkud, Muhammad Adil Inam, Andy Riddle, Jason Liu, Gang Wang, and Adam Bates. "How does Endpoint Detection use the MITRE ATT&CK Framework?" In: *Proceedings of the 33rd USENIX Security Symposium (USENIX Sec).* 2024.

[260] *VirusTotal.* https://www.virustotal.com/. 2024.

[261] Daniel Votipka, Seth Rabin, Kristopher Micinski, Jeffrey S. Foster, and Michelle L. Mazurek. “An Observational Investigation of Reverse Engineers’ Processes”. In: *Proceedings of the 29th USENIX Security Symposium (USENIX Sec)*. 2020.

[262] Daniel Votipka, Rock Stevens, Elissa Redmiles, Jeremy Hu, and Michelle Mazurek. “Hackers vs. Testers: A Comparison of Software Vulnerability Discovery Processes”. In: *Proceedings of the 39th IEEE Symposium on Security & Privacy (S&P)*. 2018.

[263] Qinqin Wang, Hanbing Yan, and Zhihui Han. “Explainable APT Attribution for Malware using NLP Techniques”. In: *Proceedings of the 21st IEEE International Conference on Software Quality, Reliability and Security (QRS)*. 2021.

[264] Chad Warner. *Diamond Model in Cyber Threat Intelligence*. https://warnerchad.medium.com/diamond-model-for-cti-5aba5ba5585. 2021.

[265] Adam Weidemann. *New Campaign Targeting Security Researchers*. https://blog.google/threat-analysis-group/new-campaign-targeting-security-researchers/. 2021.

[266] Georg Wicherski. “peHash: A Novel Approach to Fast Malware Clustering”. In: *Proceedings of the 2nd USENIX Workshop on Large-Scale Exploits and Emergent Threats (LEET)*. 2009.

[267] Miuyin Yong Wong, Matthew Landen, Frank Li, Fabian Monrose, and Mustaque Ahamad. “Comparing Malware Evasion Theory with Practice: Results from Interviews with Expert Analysts”. In: *Proceedings of the 20th Symposium On Usable Privacy and Security (SOUPS)*. 2024.

[268] Yiming Wu, Qianjun Liu, Xiaojing Liao, Shouling Ji, Peng Wang, Xiaofeng Wang, Chunming Wu, and Zhao Li. “Price Tag: Towards Semi-Automatically Discovery Tactics, Techniques and Procedures of e-commerce Cyber Threat Intelligence”. In: *IEEE Transactions on Dependable and Secure Computing* (2021).

[269] Khaled Yakdan, Sergej Dechand, Elmar Gerhards-Padilla, and Matthew Smith. “Helping Johnny to Analyze Malware: A Usability-Optimized Decompiler and Malware Analysis User Study”. In: *Proceedings of the 37th IEEE Symposium on Security & Privacy (S&P)*. 2016.

[270] Jia Yan, Ming Wan, Xiangkun Jia, Lingyun Ying, Purui Su, and Zhanyi Wang. “DitDetector: Bimodal Learning based on Deceptive Image and Text for Macro Malware Detection”. In: *Proceedings of the 38th Annual Computer Security Applications Conference (ACSAC)*. 2022.

[271] Junfeng Yang. *How RTF Malware Evades Static Signature-Based Detection*. https://www.mandiant.com/resources/blog/how-rtf-malware-evad. 2024.

[272] *Repository of Yara Rules*. https://github.com/Yara-Rules/rules. 2022.

[273] Miuyin Yong Wong, Matthew Landen, Manos Antonakakis, Douglas M Blough, Elissa M Redmiles, and Mustaque Ahamad. “An Inside Look into the Practice of Malware Analysis”. In: *Proceedings of the 28th ACM SIGSAC Conference on Computer and Communications Security (CCS)*. 2021.

[274] Xiaopeng Zhang. *New Remcos RAT Variant is Spreading by Exploiting CVE-2017-1188*. https://web.archive.org/web/20240516131624/https://www.fortinet.com/blog/threat-research/new-remcos-rat-variant-is-spreading-by-exploiting-cve-2017-11882. 2018.

[275] Xin Zhou and Jianmin Pang. “Expdf: Exploits Detection System Based on Machine-Learning”. In: *International Journal of Computational Intelligence Systems* (2019).

Appendix to Chapter 3

Survey Questionnaire

Consent and GDPR Consent. In this section, we obtain informed consent from participants, ensuring they understand the study's purpose, procedures, data protection measures, and their rights.

Participant Information. In this section, we ask a few questions about your background, name, age, and email address.

Job Description.

1. What is your current job role and job title?

2. Please specify your highest level of education. Less than high school, High School graduate (high school diploma or equivalent such as GED), Some college, but no degree, Associate Degree, Bachelor's Degree, Master's Degree, Doctorate Degree (MD, PhD, JD, etc.), Prefer not to answer.

3. Please rate your expertise on the scale of Novice (basic knowledge), Fundamental Awareness (limited experience), Intermediate (practical application), Advanced (applied theory), Expert (recognized authority), None in the area of Malware analysis, APT tracking and attribution, Threat intelligence research, and Incidence response.

4. What is the end goal of your APT threat analysis work? (Please check all that apply) Forensics, Attribution, Classification and clustering, Signature creation, Indicators of Compromise, Research, writing threat reports, and others.

Work Experience. Please list any tools you use when performing APT malware analysis, APT incident response, or threat intelligence research. Please continue listing tools, as you have a new line for each tool, and continue to list tools until you cannot think of any more. These can be any tools you have used; you do not need to regularly use them.

Interview Questions

Background and Experience.

- Could you tell me a little bit about your job role and experience?
- How did you get into the field of investigating APTs? Could you describe any interesting or recent APT incident that you worked on? What steps did you take to investigate the suspected APT attack?
- What is the end goal of your work pipeline—collecting IoCs, malware analysis, threat intelligence research, or writing threat reports?

APT Analysis Objectives.

- What is the end goal of analyzing an APT sample?
  - Is it to identify tactics, observe behavior, attribute the attack, or something else?
- Where do you source your APT samples from?
  - How do you ensure that the samples are relevant to your study?
  - Do you use in-house SIEM systems, collect samples from VT or through clients/repositories, or use other methods?
- When do you start considering a malicious sample as part of a suspected APT campaign or operation?
  - How do you prioritize samples for further analysis based on their potential association with known APT campaigns?
  - Are there specific indicators or techniques you use for this prioritization?

Process and Pipeline.

- What is your digital forensics and incident response protocol when you identify malicious activity as part of an APT campaign?
  - Do you have separate workflows for dealing with APTs versus traditional malware threats?
- What is your process for attributing a sample to a specific threat group?
  - What features do you consider when making this attribution?

- For a newly identified APT campaign, what is your process for correlating samples with previously established campaigns?
  - How do you identify similar patterns or connections between APT campaigns?
  - Do you use ML-based automation for sample correlation?
  - If yes, do you see any challenges in working with ML-based tools?
- What is your approach to gathering comprehensive information about APT groups?
  - How do you keep track of potentially related campaigns over time, especially when they are spread across multiple sources?
- Are there challenges in incorporating threat intelligence into your analysis of APT campaigns?
  - What challenges do you face when attributing attacks to specific groups?
  - Are there any other challenges when grouping attacks that may have been carried out by APT groups with aliases or changing names?
- How do you effectively aggregate and consolidate data from diverse OSINT sources?
  - Can you provide an example of this process?
- What techniques do you use to manage publicly available information about APT campaigns?
  - How do you detect and eliminate redundancies?

Final Remarks.

- What are the primary concerns when dealing with APT incidents?
  - Are accuracy and precision prioritized, or is there more focus on speed, automation, or developing a generic framework for file types?
- What processes, tools, and policies currently work best in your team?