

ENHANCING THREAT HUNTING AUTOMATION WITH LARGE LANGUAGE MODELS

by

William French

A thesis submitted to the faculty of
The University of North Carolina at Charlotte
in partial fulfillment of the requirements
for the degree of Master of Science in
Cyber Security

Charlotte

2024

Approved by:

Dr. Bill Chu

Dr. Chenglong Fu

Dr. Depeng Xu

ABSTRACT

WILLIAM FRENCH. Enhancing Threat Hunting Automation with Large Language Models. (Under the direction of DR. BILL CHU)

In response to the growing complexity of cyber security threats, threat hunting has become an essential proactive security measure. However, its adoption in security operations programs is often limited to larger organizations due to the myriad of resources required to support the activity. Transformer-based [1] Large Language Models (LLMs) present a new opportunity to democratize, automate, and enhance cyber security operations. This thesis seeks to contribute to this space in three ways: First, develop a demonstration of an LLM's ability to automate aspects of threat hunting. Second, produce a dataset that will assist with fine-tuning or training. Third, contributing to the development of a Retrieval Augmented Generation (RAG) system within AIThreatTrack [2].

ACKNOWLEDGEMENTS

I express my sincere gratitude to my supervisor Dr. Bill Chu for his guidance throughout this research, and to my thesis committee for their time and feedback.

TABLE OF CONTENTS

LIST OF FIGURES	vii
LIST OF ABBREVIATIONS	ix
CHAPTER 1: INTRODUCTION	1
1.1. Background of the Thesis	1
1.2. Problem Statement	4
1.3. Purpose of the Thesis	5
1.4. Research Objectives and Questions	7
1.5. Significance of the Thesis	8
CHAPTER 2: LITERATURE REVIEW	10
2.1. Cybersecurity Threat Hunting	10
2.1.1. Threat Hunting Definitions and Evolution	10
2.1.2. Current Challenges	13
2.2. Automated Threat Hunting Strategies	14
2.2.1. Overview of Existing Approaches	15
2.3. Large Language Models in Cybersecurity	19
2.3.1. Overview and Relevance	19
2.3.2. Gaps and Challenges	20
CHAPTER 3: METHODOLOGY	22
3.1. Research Design Overview	22
3.2. Dataset Creation	23
3.3. Demo Application Development	24

	vi
3.4. Retrieval Augmented Generation (RAG) Implementation	26
3.5. Expected Outcomes	27
CHAPTER 4: RESULTS	28
4.0.1. Retrieval Augmented Generation	28
4.0.2. IOC & KQL Dataset	29
4.0.3. Demonstration Application	31
CHAPTER 5: DISCUSSION	37
5.1. Limitations	37
5.2. Suggestions for Future Research	38
CHAPTER 6: CONCLUSIONS	41
6.1. Summary of the Research Proposal	41
REFERENCES	43
APPENDIX A: Threat Hunting Demo	44
APPENDIX B: KQL Dataset	50

LIST OF FIGURES

FIGURE 1.1: The threat hunting process [3]	2
FIGURE 1.2: The transformer - model architecture.	3
FIGURE 1.3: Retrieval Augmented Generation system	6
FIGURE 2.1: ThreatRaptor system	12
FIGURE 2.2: Survey population table	13
FIGURE 2.3: AUTOMA system	16
FIGURE 2.4: AIThreatTrack system	18
FIGURE 4.1: Threat Hunt Demo	32
FIGURE 4.2: TH Demo Workflow	33
FIGURE 4.3: Extracted IOCs	34
FIGURE 4.4: Brainstormed Events	34
FIGURE 4.5: Generated Elasticsearch DSL	35
FIGURE 4.6: Validation	36
FIGURE A.1: Import Appliance	44
FIGURE A.2: File Selection	45
FIGURE A.3: Application Settings	45
FIGURE A.4: Completed Import	46
FIGURE A.5: Booted VM	46
FIGURE A.6: Checking Elasticsearch Service	47
FIGURE A.7: Threat Hunt Demo Ready	48
FIGURE A.8: Extract IOC Code	48

FIGURE A.9: ponder_events

49

FIGURE A.10: generate_dsl_query

49

LIST OF ABBREVIATIONS

AI	Artificial Intelligence
CTI	Cyber Threat Intelligence
DSB	Defense Science Board
EK	Elasticsearch, Kibana
IOC	Indicator of Compromise
IR	Incident Response
IT	Information Technology
LLM	Large Language Model
P2OG	Proactive Preemptive Operations Group
RAG	Retrieval Augmented Generation
SIEM	Security Incident and Event Management
SOAR	Security Orchestration, Automation, and Response

CHAPTER 1: INTRODUCTION

1.1 Background of the Thesis

Threat hunting is the proactive process of searching for undetected or ongoing security risks within an environment. Unlike reactive threat detection, threat hunting is characterized by its hypothesis-driven nature. Effective threat hunting requires a team of analysts with expertise across various security domains, making it a resource-intensive activity often beyond the reach of smaller organizations due to budgetary, expertise, and organizational constraints.

Figure 1.1 illustrates the threat hunting cycle as a continuous loop of activity. The process begins with the Pre-hunt Plan, which is informed by various threat hunting methodologies. This is followed by Data Collection, designed to facilitate the planned hunting activities. The core of the cycle involves Hunting and Validation, where the actual threat hunting and analysis occur. The loop concludes with Remediation and Reporting before restarting. Notably, both the Remediation and Hunting phases provide valuable insights that feed back into the Pre-hunt Planning and Data Collection stages, creating a dynamic and self-improving process.

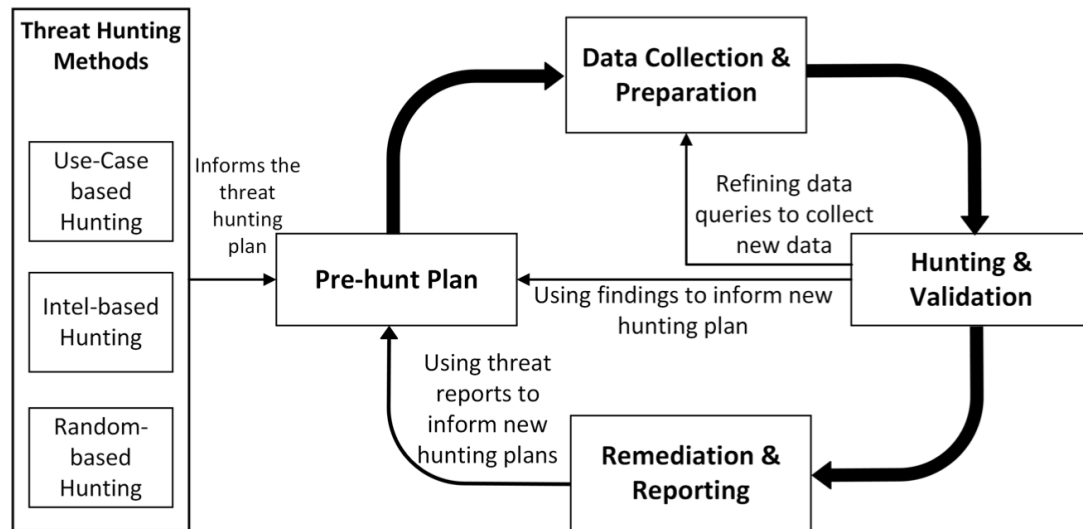


Figure 1.1: The threat hunting process [3]

As enterprise environments grow more complex, the demand for scalable threat hunting activities increases. While automation enhances threat hunting, full automation remains challenging due to its reliance on human intuition, expertise, and the development of hypotheses informed by CTI and IOCs [3]. These hypotheses require a deep understanding of the environment, current CTI, and the personal experiences of the analyst. The ongoing shortage of experienced cybersecurity professionals further complicates the scaling of threat hunting programs, as even the most qualified analysts must spend considerable time familiarizing themselves with the enterprise’s unique architecture and processes [3].

The advent of more powerful LLMs offer a new opportunity to enhance the automation of threat hunting. LLMs, such as ChatGPT, Claude, and LLaMA, leverage transformer architecture to process inputs in parallel, significantly improving contextual understanding [1]. These models are trained on extensive datasets to generate human-like text, making them highly adaptable for tasks such as generating complex queries and interpreting large volumes of data. Figure 1.2 illustrates the transformer architecture, which consists of fully connected encoder and decoder layers.

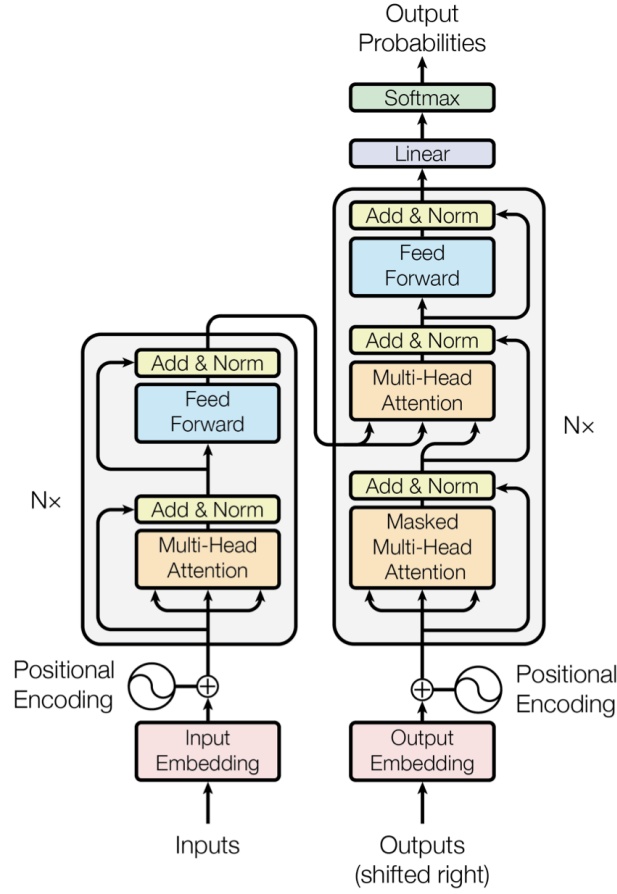


Figure 1.2: The transformer - model architecture. [1]

Training LLMs is resource-intensive, requiring vast, curated datasets, significant computational power, and specialized processors. Despite these challenges, advancements in training, fine-tuning, and prompt engineering have shown promise in improving LLM performance for specific tasks. This thesis aims to leverage these advancements to enhance LLM performance in generating operationally useful and syntactically valid Elasticsearch and Kibana queries for threat hunting within an Elasticsearch-Kibana (EK) stack. As cyber threats evolve in both volume and complexity, the ability to automate the generation of queries that can efficiently navigate and extract relevant information from an EK stack becomes increasingly critical. This thesis will explore methods to enhance LLM capabilities in automating threat

hunting, addressing a significant gap in current cybersecurity practices.

1.2 Problem Statement

Integrating LLMs into the threat hunting process presents several significant challenges:

1. **Syntactic Correctness:** A primary issue is ensuring that LLMs consistently generate syntactically correct output, particularly when creating Elasticsearch queries. This challenge extends beyond mere notation errors; it requires the LLM to produce outputs that are:
 - (a) Free from unnecessary conversational elements
 - (b) Aligned with the specific format and syntax expected by the system's configuration
 - (c) Correctly structured according to the specific Elasticsearch index being searched
2. **Contextual Understanding:** Effective query generation depends on the deep knowledge that human analysts typically bring to the task. This includes:
 - (a) Understanding which fields are available to search
 - (b) Knowing the origins of the logs
 - (c) Comprehending other contextual information that shapes the structure of a query
 - (d) Knowing which fields are most relevant to searching for specific behavior
3. **Operational Utility:** Generated queries must be not only syntactically correct but also operationally useful. A technically correct query that fails to yield actionable insights or support the analyst's objectives does not contribute meaningfully to the threat hunting process.

While LLMs can be guided through prompt engineering techniques such as N-shot prompting (providing background context or explicit examples of Cyber Threat Intelligence alongside corresponding analyst-created queries), these models still struggle to replicate the nuanced understanding possessed by human analysts.

Thus, the critical challenge in automating threat hunting lies in ensuring that LLM-generated queries are both syntactically valid and functionally relevant, effectively bridging the gap between machine-generated output and human-level analytical insight.

1.3 Purpose of the Thesis

The purpose of this thesis is to advance the integration of LLMs into the threat hunting process by addressing key challenges related to query generation, contextual understanding, and operational relevance. This thesis aims to contribute to ongoing efforts in cybersecurity automation through three primary outcomes:

1. **Retrieval Augmented Generation (RAG) System Development:** This thesis has developed a RAG system, as discussed in the paper "AIThreatTrack: Towards Automated End-to-End Threat Hunting with Generative AI" (AIThreatTrack) [2]. The system was designed to:

- (a) Compare the performance of AIThreatTrack in extracting IOCs from CTI
- (b) Accurately map these IOCs to MITRE ATT&CK technique IDs

Initial findings revealed that while the RAG system had a slightly higher rate of hallucinations (errors where the AI generates information not present in the input data), it also highlighted AIThreatTrack's strengths in this context. Figure 1.3 illustrates the RAG system's design, showcasing the various steps involved in its implementation.

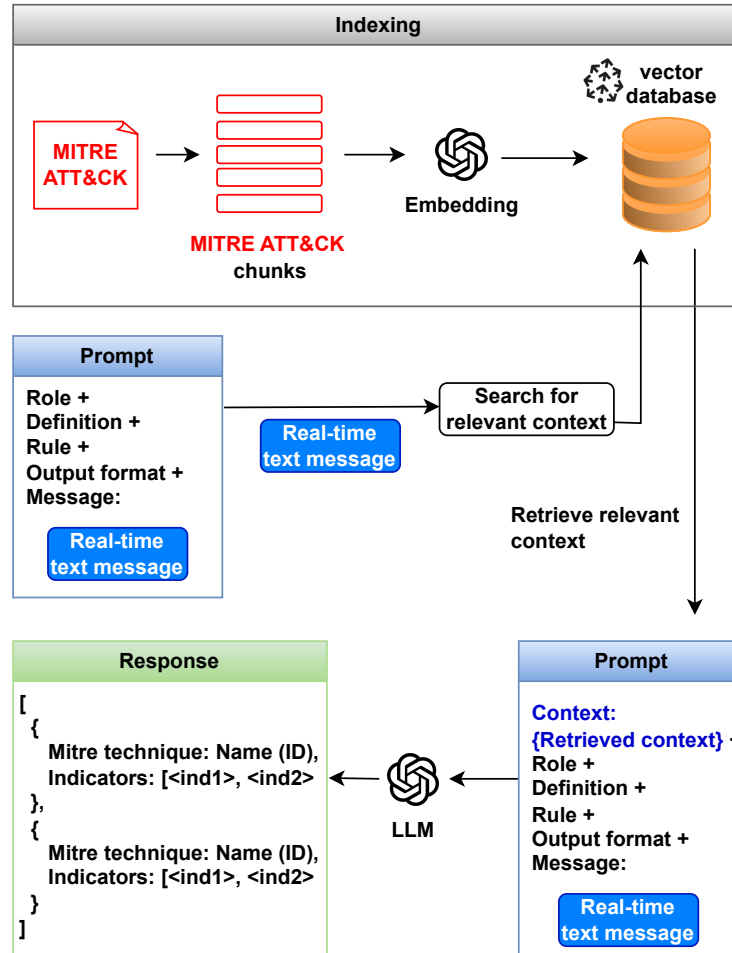


Figure 1.3: Retrieval Augmented Generation system [2]

2. **Curated Dataset Creation:** This thesis will produce a curated dataset pairing CTI-derived IOCs with Kibana queries that are:

- (a) Syntactically correct
- (b) Tailored to the specific configuration of the Elasticsearch-Kibana (EK) stack used to collect host logs

This dataset will serve as a benchmark for evaluating LLM performance in generating operationally useful queries within a real-world cybersecurity context, providing a valuable resource for future research and development.

3. **Working Prototype Development:** The thesis will culminate in the devel-

opment of a working prototype based on the system described in AIThreatTrack [2]. This prototype will:

- (a) Showcase the practical application of LLMs in automating query generation for threat hunting
- (b) Demonstrate how these models can be adapted to meet specific cybersecurity operational needs

Through these contributions, this thesis aims to enhance the effectiveness and reliability of LLMs in threat hunting, ultimately providing valuable tools and datasets that can be leveraged by the cybersecurity community to improve threat hunting and response. This prototype will be packaged as an Ubuntu Virtual Machine (VM). This VM will include Elasticsearch and Kibana environments, along with Windows Event log data that will enable the prototype to utilize the validation API.

1.4 Research Objectives and Questions

The objectives and questions of this thesis are closely aligned with the expected outcomes, providing a clear framework for advancing the integration of LLMs into the threat hunting process.

The first objective is to develop and evaluate a RAG system to improve the extraction of IOCs from CTI samples, and accurately map them to MITRE ATT&CK framework techniques. This objective is critical for enhancing the precision of threat hunting tools and reducing the reliance on manual analysis.

The second objective is to create a curated dataset of CTI-derived IOCs along with syntactically correct and useful Elasticsearch queries. This dataset will serve as a benchmark for evaluating LLM performance and provide a valuable resource for future research in cybersecurity.

The final objective is to develop and implement a prototype application of the system described in the AIThreatTrack paper. This prototype will demonstrate the

practical application of LLMs in automating query generation, showcasing their potential to enhance the efficiency and effectiveness of threat hunting operations.

To guide this thesis, the following questions have been formulated:

- **Question 1:** How effectively can a RAG system be used to align IOCs with their associated MITRE ATT&CK techniques compared to the AIThreatTrack system?
- **Question 2:** What are the key factors that influence the ability of an LLM to generate syntactically correct and operationally useful Elasticsearch queries?
- **Question 3:** How can LLMs be adapted to improve the automation of query generation in threat hunting, and what are the limitations of this approach?

These research questions are designed to explore the capabilities and limitations of LLMs in the context of threat hunting, guiding the thesis toward its intended outcomes.

1.5 Significance of the Thesis

The threat hunting process is an expensive and time-consuming endeavor that typically requires highly experienced professionals with specific insights into an enterprise IT environment. These experts manually interpret and craft queries to support their hypotheses, making the process resource-intensive and often out of reach for smaller organizations with limited resources. The potential of LLMs to accelerate a smaller team's ability to conduct effective threat hunting offers these organizations a way to compete with larger counterparts in identifying and remediating cybersecurity threats.

For smaller teams, LLMs could significantly enhance their capacity to cover necessary ground, enabling them to maintain the safety and security of their enterprise environments despite limited resources. Larger teams, which typically oversee more

complex environments, could further fine-tune LLMs to search for behaviors in specific contexts, such as differentiating between endpoint logging and network traffic logs. In both cases, LLMs offer the potential to scale threat hunting activities, making them more efficient and accessible.

In the realm of academic research, this thesis contributes to the foundational understanding of how LLMs can be evaluated and guided to generate content that is not traditionally "human-like" but remains operationally useful. This challenges the current paradigms of LLM usage, opening new avenues for their application in cybersecurity.

From a practical standpoint, the prototype developed in this thesis could serve as a valuable tool for cybersecurity teams, who may further refine and adapt it to their specific needs. Although the prototype is not intended to execute queries autonomously, it will be validated against a local Elasticsearch API to ensure syntactic correctness and alignment with the index structure. This validation process is crucial for demonstrating the prototype's practical applicability and its potential for seamless integration with existing EK stack deployments.

Looking toward the future, this thesis could play a role in the ongoing development of agentic AI—AI systems capable of interacting with external tools and environments beyond the scope of traditional chatbots. In the context of threat hunting, generating an Elasticsearch query could become one of several tools in an AI agent's toolkit, enable more dynamic and responsive cybersecurity operations.

In summary, this thesis has the potential to contribute significantly to both academic research and the practical operations of security teams. By building a foundation for further research and evaluation of LLMs in threat hunting, empowering AI agents to utilize these tools effectively, and enhancing the ability of security teams to manage risks, this thesis stands to make a lasting impact on the field of cybersecurity.

CHAPTER 2: LITERATURE REVIEW

2.1 Cybersecurity Threat Hunting

As cyber threats grow in complexity, cybersecurity threat hunting has emerged as a crucial part of modern defense strategies. This chapter outlines the origins and development of threat hunting, such as the establishment of the Proactive Preemptive Operations Group (P2OG) [4] and the creation of automated systems like ThreatRaptor [5], we explore how proactive strategies have transformed the field. This section also introduces the current challenges that cybersecurity professionals face as they work to stay ahead of increasingly sophisticated adversaries.

2.1.1 Threat Hunting Definitions and Evolution

Modern ideas of proactive defensive operations are exemplified in a 2002 report by the Defense Science Board (DSB), which recommended the establishment of the "Proactive Preemptive Operations Group" (P2OG) under the United States National Security Council (NSC)[4]. This group was designed to conduct counter-terrorism operations with a proactive and preemptive focus, aiming to anticipate and neutralize threats by stimulating adversary reactions and thereby enhancing intelligence collection and operational preparedness [4]. While the DSB's recommendations primarily targeted counter-terrorism activities in domains beyond cybersecurity, the underlying principle of preemptive action is highly relevant to modern cybersecurity practices.

The strategy employed by P2OG—anticipating and disrupting threats before they fully materialize—parallels the proactive defense strategies that are increasingly critical in the cybersecurity landscape. Just as P2OG sought to mitigate risks through preemptive operations at a national level, enterprises can apply similar concepts to

defend their networks against cyber threats. Although the scale and scope counter-terrorism operations exceed what is typically required for enterprise security, the core principle of identifying and neutralizing threats before they cause harm remains universally applicable. By adopting a proactive stance, cybersecurity teams can better anticipate potential threats and respond more effectively, reducing their overall vulnerability to attacks.

As cybersecurity threats have continued to evolve in both complexity and scale, the principles of proactive defense, once applied to national security, have increasingly been adapted to meet the needs of the digital domain. This alignment between proactive counter-terrorism strategies and proactive cybersecurity operations underscores the universality of preemptive defense as a cornerstone of comprehensive security strategies across various domains.

As the concept of proactive defense gained traction, cybersecurity practitioners began exploring ways to enhance the efficiency and effectiveness of threat hunting, a process traditionally marked by high costs and manual labor intensity. Historically, threat hunting required skilled analysts to manually interpret data and construct queries based on their deep understanding of the threat landscape. While thorough, this manual approach was time-consuming, resource-intensive, and susceptible to human error, making it increasingly challenging for organizations to keep pace with rapidly evolving threats.

The advent of advanced automation tools, coupled with the increasing availability of shared CTI, has driven significant progress toward automating the threat hunting process. In the paper "Enabling Efficient Cyber Threat Hunting With Cyber Threat Intelligence," the authors introduce "ThreatRaptor," a system designed to streamline and automate threat hunting by leveraging CTI and analyst input [5].

ThreatRaptor represents a significant advancement in the field by employing newer technologies such as knowledge graphs and Natural Language Processing (NLP) to

parse and understand unstructured CTI reports. By extracting structured threat behaviors from these reports, ThreatRaptor can automatically generate and execute queries to identify potential threats within an enterprise environment. These advanced technologies allow ThreatRaptor to significantly reduce the manual effort traditionally required in threat hunting, making the process more efficient and less prone to errors.

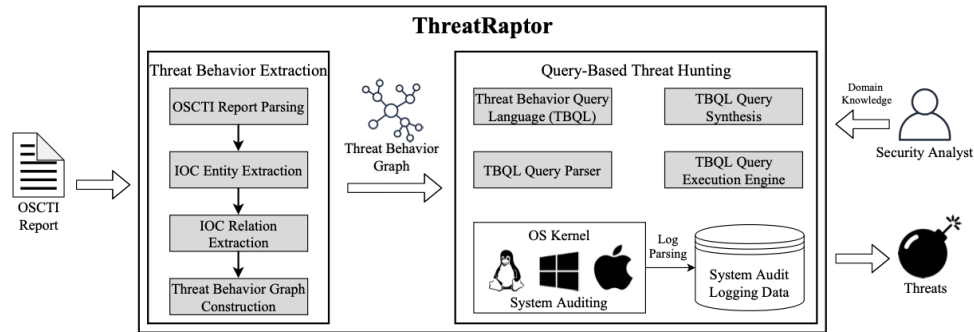


Figure 2.1: ThreatRaptor system [5]

Moreover, ThreatRaptor’s ability to integrate external threat knowledge with internal system auditing data marks a critical step forward in the automation of threat hunting. By bridging the gap between unstructured CTI and structured system logs, the system not only enhances the accuracy of threat detection but also offers a scalable solution adaptable to various enterprise environments. While ThreatRaptor is one of several systems advancing automated threat hunting, its innovative approach highlights the growing potential of AI and machine learning technologies to revolutionize cybersecurity practices.

This shift towards automation in threat hunting, exemplified by systems like ThreatRaptor, reflects a broader trend in cybersecurity: the transition from reactive to proactive, automated defense mechanisms. As these technologies continue to evolve, they promise to make threat hunting more accessible, efficient, and effective, enabling organizations of all sizes to better protect themselves against increasingly

sophisticated cyber threats.

2.1.2 Current Challenges

The continuous nature of threat hunting can be conceptualized as an ongoing cycle, where each phase presents unique challenges. A 2024 study[3], which surveyed a broad spectrum of cybersecurity professionals engaged in threat hunting, identified several persistent challenges in this process, categorizing them in to three main areas: Methodology, Data, and Organizational/People.

Table 1: Participants demographics details "-" : Prefer not to answer

ID	Job Role	Country	Experience	Education	Company	Type of Service	Recruitment Method
P01	Senior Cybersecurity Analyst	UK	10-15 years	PhD	Company 1	In-house	Industry Connection
P02	Security Consultant	Australia	15-20 years	MSc	Company 2	MSSP	Industry Connection
P03	Threat Intelligence Analyst	UK	5-10 years	Bachelor	Company 1	In-house	Industry Connection
P04	Associate Director Threat Hunt	US	10-15 years	-	Company 2	MSSP	Industry Connection
P05	Threat Hunting Team Lead	US	-	-	Company 2	MSSP	Industry Connection
P06	Digital Forensics Specialist	UK	5-10 years	Bachelor	Company 3	MSSP	Industry Connection
P07	SOC Analyst	US	10-15 year	Bachelor	Company 4	In-house + MSSP	Industry Connection
P08	Director for DFIR	Singapore	10-15 years	MSc	Company 2	MSSP	Industry Connection
P09	Consultant	US	15-20 years	Bachelor	Company 2	MSSP	Industry Connection
P10	Lead SOC Threat Hunter	US	5-10 years	High School	Company 2	MSSP	Industry Connection
P11	IT Security Engineer	US	15-20 years	Bachelor	Company 2	MSSP	Industry Connection
P12	SOC Analyst	India	10-15 years	Bachelor	Company 4	In-house + MSSP	Snowball
P13	Security Analyst L3	UAE	5-10 years	-	Company 5	In-house + MSSP	Slack
P14	Program Lead Adv Sec Analytics	US	15-20 years	-	Company 6	In-house	Slack
P15	Threat analyst	UK	5-10 years	High school	Company 7	MSSP	Slack
P16	Cybersecurity Technical Specialist	UK	10-15 year	Bachelor	Company 3	MSSP	Snowball
P17	SOC Head	UK	10-15 years	MSc	Company 8	MSSP	Industry Connection
P18	Security Research Lead	UK	15-20 years	Bachelor	Company 9	In-house + MSSP	Snowball
P19	Cybersecurity Engineer	Germany	15-20 years	-	Company 10	In-house	Snowball
P20	Lead Cybersec Engineer	US	10-15 years	MSc	Company 5	In-house + MSSP	Slack
P21	Manager, Incident Handling	US	10-15 years	MSc	Company 11	In-house	Snowball
P22	Senior Incident Response Consultant	Qatar	10-15 years	MSc	Company 9	In-house + MSSP	Snowball

Figure 2.2: Survey population table [3]

Methodology Challenges: One of the most significant methodological challenges is the prevalence of false alerts. These require substantial time and effort to investigate, often diverting resources from more critical tasks. Additionally, building effective use cases and hypotheses is complicated by the constantly evolving tactics and techniques employed by adversaries. This fluidity demands that threat hunters continuously adapt their strategies, which can be hindered by the limitations and occasional failures of automated tools and systems designed to assist in these tasks. These tools, while beneficial, can introduce new issues when they fail to perform as expected, leading to disruptions in the hunting process.

Data Challenges: Data-related issues are pervasive in threat hunting, with the complexity and quality of data being paramount concerns. Threat hunters often grapple with complex datasets that are difficult to standardize and analyze, leading to inefficiencies. Incomplete or low-quality data further exacerbates these challenges, as it can result in blind spots that leave threats undetected. Additionally, the overwhelming volume of data generated by modern IT environments can reduce the signal-to-noise ratio, making it harder for analysts to identify actionable insights. Storage limitations also pose a significant challenge, as organizations may lack the capacity to retain the necessary data for extended periods, thus hindering long-term threat analysis.

Organizational and People Challenges: On the organizational side, the shortage of skilled personnel remains a critical issue. Threat hunting requires not only technical expertise but also strategic thinking and the ability to adapt to rapidly changing threat landscapes. However, the high demand for such professionals often outstrips supply, making recruitment and retention difficult. Communication issues within teams and across departments can also impede the effectiveness of threat hunting efforts, as misunderstandings or miscommunications can lead to delayed responses or overlooked threats. Budget constraints further compound these issues, limiting the resources available for hiring, training, and deploying advanced threat hunting tools.

2.2 Automated Threat Hunting Strategies

This section explores the distinct approaches to automating cybersecurity activities, with a focus on comparing the well-established automation in Incident Response (IR) with the more challenging domain of Threat Hunting. While IR automation, facilitated by platforms like Security Orchestration, Automation and Response and Security Incident Event Management (SIEM), thrives on repeatable processes and known threats, Threat Hunting demands a higher level of flexibility and human intuition, making full automation more complex and elusive. Recent advancements

in threat hunting automatic systems, such as AUTOMA and AIThreatTrack, aim to automate the generation and refinement of attack hypotheses using a variety of methods. Although these innovations show promise, they still face limitations, particularly in fully automating the validation and execution phases of threat hunting.

2.2.1 Overview of Existing Approaches

IR and Threat Hunting are both critical to cybersecurity, but they operate on fundamentally different principles—IR being reactive and Threat Hunting proactive. These differences extend to the automation tools that support each activity. Automation in IR is more advanced, largely due to IR processes being well-documented, repeatable, and based on known threats. These characteristics make IR well-suited for automation through SOAR and SIEM platforms which can manage a series of predefined responses to specific incidents.[6].

In contrast, Threat Hunting is inherently more complex and less predictable, making it more challenging to automate. Threat Hunting involves the formulation of hypotheses about potential threats that have not yet been detected, requiring a high degree of flexibility and adaptability—qualities that are difficult to codify into automated processes. While some aspects of Threat Hunting, such as data collection and initial analysis, can be supported by automation tools, the core activities of hypothesis generation and testing still rely heavily on human intuition and expertise. This reliance on human judgement highlights a significant gap in the current state of cybersecurity automation, where the proactive nature of Threat Hunting demands a level of adaptability that current technologies are only beginning to address [6].

Recent research in cybersecurity has introduced advanced systems aimed at automating proactive, hypothesis driven operations. One such system, AUTOMA, leverages system telemetry and a comprehensive knowledge base from the MITRE ATT&CK framework to automate the generation and refinement of attack hypotheses. AUTOMA operates through two key agents: the Hypothesis Generator (HG) and

the Hypothesis Examiner (HE). These agents collaboratively generate potential attack scenarios based on observed system activities and known attack patterns stored in the knowledge base [7].

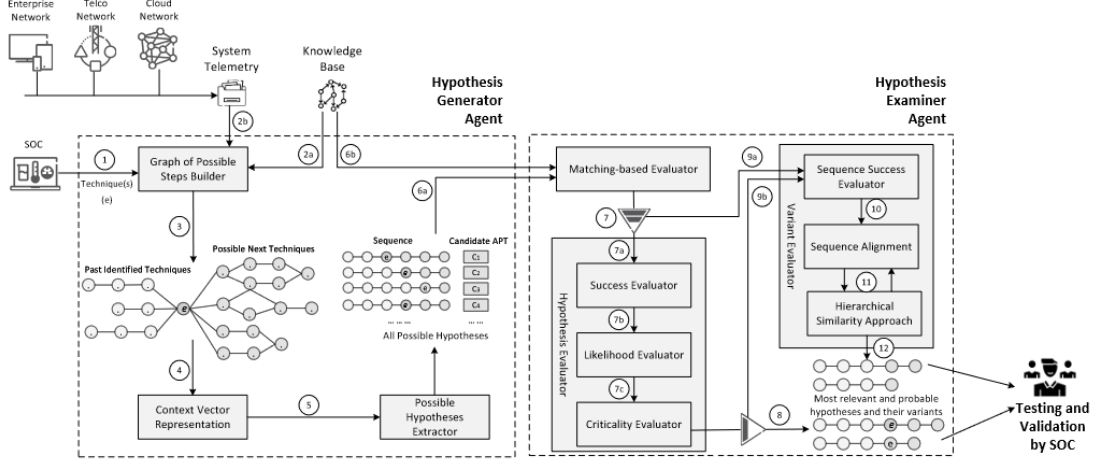


Figure 2.3: AUTOMA system [7]

The Hypothesis Generator uses system telemetry to construct a graph of possible attack steps, while the knowledge base provides contextual information about known threats, allowing the system to generate relevant hypotheses. The Hypothesis Examiner then evaluates these hypotheses, pruning less likely scenarios to focus on those with the highest probability of success based on historical data and attack patterns [7].

While AUTOMA represents a significant advancement in automating the hypothesis generation phase of threat hunting, it is currently limited by its inability to carry out the actual hunting for these hypothesized threats. This limitation underscores a critical gap in the threat hunting cycle, where the system’s generated hypotheses must still be validated and acted upon by human analysts. Despite this, AUTOMA’s ability to bridge the gap between system-specific telemetry and general cybersecurity knowledge offers a powerful tool for enhancing proactive defense measures in enterprise environments [7].

AIThreatTrack is another system designed to automate the threat hunting pro-

cess, but it distinguishes itself from AUTOMA by utilizing LLMs like ChatGPT-4 at multiple stages of the process. In its current implementation, ChatGPT-4 plays a central role in extracting IOCs from CTI, generating templates, identifying relationships between IOCs, and finally producing Kibana queries for threat detection [2]. While ChatGPT-4 is employed in this iteration, future versions of AIThreatTrack could benefit from more specialized models, potentially fine-tuned to specific threat-hunting tasks for improved accuracy.

One of the key innovations in AIThreatTrack is its use of vector embedding to assess the similarity of extracted elements. By embedding IOCs into a vector database, the system can determine the relationship between concepts based on similarity scores. If two elements surpass a defined threshold of similarity, they are considered related and used in query generation. If not, they are categorized as hallucinations and discarded, helping to mitigate the inclusion of irrelevant or inaccurate data in the final analysis. This threshold-based mechanism plays a crucial role in ensuring that the system remains reliable in its outputs.

AIThreatTrack’s focus on reducing hallucinations is critical to maintaining the accuracy and usefulness of the generated queries. Hallucinations—when the LLM introduces non-existent or inaccurate information—can significantly undermine the effectiveness of threat hunting. One method of addressing this is implementing a RAG approach, allowing ChatGPT-4 to not only extract the relevant IOCs but also assign them appropriate MITRE ATT&CK techniques by using a corpus of the techniques with their definitions and IDs. By combining retrieved knowledge with real-time LLM outputs, AIThreatTrack ensures that the context provided to the model is accurate and relevant, further reducing the risk of hallucinations and improving the overall quality of the queries generated [2].

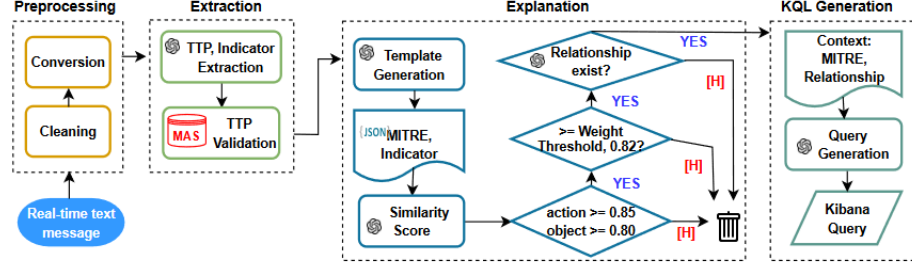


Figure 2.4: AIThreatTrack system [2]

The combination of these methods has led to a significant reduction in hallucinations, lowering the rate from 47% to just 1.5% [2]. In addition to improving accuracy, the system encourages transparency by having the LLM "explain" its reasoning when extracting content. This explanation not only serves as additional context that the LLM can reference in subsequent steps but also allows analysts to trace the model's thought process. By making the system's reasoning visible, analysts can better understand how the LLM arrives at its conclusions, which fosters greater trust in the system and helps users diagnose performance issues—whether the system is functioning well or under performing.

This approach aligns with the "chain-of-thought" reasoning strategy, where the LLM explicitly outputs its thought process to guide its reasoning in later stages. By maintaining a clear record of how it interprets and analyzes data, the system enhances its ability to make more informed decisions. For analysts, this transparency not only builds confidence in the system's outputs but also provides valuable insights into the underlying logic that drives threat hunting operations. This method of reasoning allows the LLM to consistently refine its outputs, improving overall performance and reducing errors such as hallucinations.

2.3 Large Language Models in Cybersecurity

2.3.1 Overview and Relevance

The application of LLMs in cybersecurity is rapidly gaining traction, driven by the need for more intelligent and scalable solutions to counter increasingly sophisticated cyber threats. LLMs, which have shown remarkable success in various NLP tasks, are now being leveraged to address a wide range of cybersecurity challenges, such as their use in AIThreatTrack [2]. These models, such as GPT-4, Claude, and their variants, are capable of understanding and generating human-like text, making them suitable for tasks that involve analyzing large volumes of security-related data, detecting vulnerabilities, and even responding to cyber incidents [8].

LLMs are particularly relevant in cybersecurity due to their ability to process unstructured data, such as security logs, incident reports, and threat intelligence feeds, which are often rich in critical information but challenging to analyze manually. By applying LLMs, cybersecurity professionals can automate the extraction of insights from these data sources, enabling faster and more accurate identification of potential threats. This capability is essential in an era where the sheer volume of data can overwhelm traditional analysis methods [9].

The versatility of LLMs in cybersecurity is further demonstrated by their application across various domains, including network security, software vulnerability detection, malware analysis, and threat intelligence. For instance, LLMs have been used to enhance the detection of network intrusions by analyzing traffic patterns and identifying anomalies that may indicate malicious activity. Similarly, in the realm of software security, LLMs assist in identifying vulnerabilities within code bases by analyzing both source code and natural language descriptions of potential threats [8].

Moreover, the adaptability of LLMs allows them to be fine-tuned for specific cybersecurity tasks, ensuring that they can handle the unique challenges presented by different security domains. This customization is often achieved through techniques

such as transfer learning, where a pre-trained LLM is further trained on domain-specific data to enhance its performance in a particular area of cybersecurity.

Overall, the integration of LLMs into cybersecurity practices represents a significant advancement in the field, offering a powerful tool set for automating and enhancing the effectiveness of security operations. As research in this area continues to evolve, LLMs are expected to play an increasingly central role in defending against the ever-growing landscape of cyber threats [9].

2.3.2 Gaps and Challenges

While LLMs have demonstrated significant potential in enhancing cybersecurity operations, their integration into this domain is not without challenges. These challenges can be broadly categorized into issues related to the security of LLMs themselves, the quality of data they rely on, and the inherent limitations of applying NLP models to cybersecurity tasks.

One of the primary challenges is the security of the LLMs themselves. LLMs are susceptible to various adversarial attacks, such as prompt injection, where malicious actors manipulate the input to elicit specific, often harmful, outputs. These vulnerabilities can be exploited to bypass security measures, introduce bias, or even generate misleading or dangerous content, thereby undermining the security systems they are meant to protect [9]. Ensuring the robustness in cybersecurity could introduce new avenues for exploitation if not adequately safeguarded.

Another significant challenge is the quality and diversity of the data used to train and fine-tune LLMs for cybersecurity tasks. LLMs require vast amounts of data to perform effectively, but in the cybersecurity domain, obtaining high-quality, representative datasets is particularly difficult. Much of the data in cybersecurity is sensitive, fragmented, or proprietary, limiting the availability of comprehensive training sets. This scarcity of data can lead to models that are not fully equipped to handle the wide variety of threats encountered in real-world scenarios, resulting in reduced ef-

fectiveness [9].

Moreover, LLMs, while powerful in natural language processing, may struggle with the highly technical and context-dependent nature of cybersecurity tasks. For instance, understanding and generating code, interpreting complex network traffic patterns, or analyzing cryptographic algorithms often require specialized knowledge that goes beyond what current LLMs can provide. The application of LLMs in these areas is further complicated by the need for precise and unambiguous outputs, where even minor errors can lead to significant security vulnerabilities [9].

Finally, there is a broader challenge in integrating LLMs into existing cybersecurity frameworks and workflows. Enterprise environments are complex, with many moving parts, including diverse software, hardware, and human elements. Integrating LLMs in a way that complements rather than complicates these environments requires careful consideration of how these models interact with other security tools and how their outputs are interpreted and acted upon by human analysts.

In summary, while LLMs hold great promise in advancing cybersecurity, addressing these gaps and challenges is essential to ensure their safe and effective development. Ongoing research is needed to develop more robust, secure, and context-aware LLMs that can truly enhance the security landscape without introducing new risks.

CHAPTER 3: METHODOLOGY

3.1 Research Design Overview

This thesis explores the application of LLMs in cybersecurity, focusing on their ability to generate valid Elasticsearch DSL queries. The research design includes developing a demo application, creating a specialized dataset, and implementing a RAG system in a version of the larger AIThreatAttack system.

The demo application, to be built with Python and Streamlit, will allow users to input a CTI sample, which the LLM will process in order to extract IOCs and generate a query. This query is then validated against a local Elasticsearch environment, showcasing how LLMs can automate query generation in real time.

The second component involves creating a dataset consisting of 216 IOCs extracted from 71 CTI tweet samples. For each IOC, a Kibana query will be manually created to ensure the syntactical accuracy and utility within a winlogbeat index. This dataset will serve as a benchmark for evaluating and tuning LLMs in generating precise and operationally useful queries.

The final component integrates a RAG system into AIThreatTrack. The system embeds the MITRE ATT&CK dataset into a vector database, retrieving relevant information to assist the LLM in assigning correct ATT&CK techniques to IOCs. While the RAG system did not outperform the base system in all areas, it provided valuable insights into the use of retrieval-augmented models to improve contextual understanding.

This multi-faceted research design evaluates LLM application in cybersecurity, contributing new tools, datasets and retrieval-based techniques to enhance threat hunting activity.

3.2 Dataset Creation

Using 216 IOCs extracted from 71 tweets as a part of [2], a Kibana query will be created for each of them that would be considered syntactically accurate and operationally useful. This relates back to the objective to better tune LLMs to creating such searches, although with a different query language.

To support the development and tuning of LLMs for generating accurate and operationally useful search queries, a dataset consisting of CTI in the form of tweets will be constructed. This dataset will include 336 IOCs extracted from 71 unique tweets, previously identified as part of the research outline in [2].

The important fields of the dataset are:

1. **Tweet:** Provides the raw CTI in a condensed format, simulating real-world intelligence sharing scenarios on platforms like X, formerly Twitter.
2. **IOC Extraction:** Each tweet is processed by an LLM to extract the relevant IOCs, showcasing the model's ability to analyze unstructured data and identify potential threats [2].
3. **Kibana Query:** For each IOC, a Kibana query that is both syntactically correct and operationally useful will be written. These queries will target winlogbeat/sysmon indices, ensuring that the dataset is tailored for practical use in threat hunting within enterprise environments.

This dataset not only aids in evaluating LLM performance in generating accurate queries but also provides a valuable benchmark for tuning these models to work with different query languages. By ensuring that each Kibana query is correct, the dataset emphasizes real-world utility, helping bridge the gap between machine-generated outputs and human-level operational effectiveness.

The associated Kibana Query Language searches will be based on a Windows Event

log index within Elasticsearch and Kibana, in order to validate that the searches are syntactically correct.

3.3 Demo Application Development

To demonstrate the capability of an LLM to generate valid Elasticsearch Domain Specific Language (DSL) queries, a prototype application with focused functionality will be developed. The application will be written in Python, a language chosen for its simplicity and support for rapid development through a variety of modules. The application interface will be built using Streamlit, which allows for quick web-based UI development, enabling users to upload CTI data in the form of PDFs or plain text for analysis.

The application interacts with several key modules:

- **OpenAI's Python Module** to simplify activities related to LLM interactions like calling the completion API to get responses from ChatGPT.
- **Elasticsearch's Python Module** to validate generated queries from the local Elasticsearch instance.
- **pdfminer, logging, sys and json** to handle PDF parsing, application logging, and data management.

Application Workflow

1. **Application Setup and User Interaction:** When running the Streamlit app, users are prompted to input their OpenAI API key, allowing the app to connect to ChatGPT. Users can then upload a PDF containing cyber threat intelligence or paste CTI into an input box.
2. **Elasticsearch Connection:** In the background, the application establishes a connection with the local Elasticsearch instance, retrieving a list of available

indices. This index list is crucial for validating DSL queries. For this prototype, the focus is on the winlogbeat index, which stores Windows event logs.

3. **Data Extraction and Processing:** Once the CTI data is submitted, the application moves through three processing steps:

- **IOC Extraction:** Using pdfminer to extract text from the uploaded PDF, the app sends this text to the LLM. A system prompt instructs the model to extract Indicators of Compromise (IOCs) as a cybersecurity analyst would. These IOCs are compiled into a list.
- **Possible Relevant Windows Event Log Generation:** The extracted IOCs are then passed back to the LLM, with instructions to generate corresponding Windows event logs that would be relevant to the IOCs and behaviors. These are not actual logs, simply listing out log types (based on event ID) that could relate.
- **DSL Query Generation:** In the final step, the LLM is provided with both the IOCs and the relevant event log Ids, alongside the original CTI data, and instructed to produce an Elasticsearch DSL query that conforms to the fields in the winlogbeat index.

4. **Query Validation:** After generating the DSL query, the application performs a validation step by sending the query to the selected index on the Elasticsearch instance. The validation result, which checks for syntax correctness and index comparability, is displayed to the user.

By following this workflow, the demo application serves as a practical demonstration of how LLMs can automate portions of the threat hunting process, from parsing CTI data to generating actionable search queries. This prototype would provide a glimpse into the potential of integrating LLMs into cybersecurity tools to enhance efficiency in threat hunting.

This application relies on several other components that must be present in order to function. For ease of use, a VM with the required infrastructure will be created. This VM will be based on the Ubuntu Operating system, and have Elasticsearch and Kibana installed as services. Additionally, a Windows Event log index will be created to validate the generated queries against. Python, as well as all of the dependencies, will be pre-installed as well. Users will simply need to run the VM, login, let the Elasticsearch and Kibana start, then click the desktop shortcut to launch the app.

3.4 Retrieval Augmented Generation (RAG) Implementation

A significant contribution of this thesis is the implementation of a RAG system for AIThreatTrack, as part of previous research efforts in collaborations with another student’s doctoral dissertation [2]. This system was designed to enhance the performance of LLMs in extracting and assigning MITRE ATT&CK Technique IDs to IOCs extracted from provided CTI.

The RAG system leverages the MITRE ATT&CK Framework corpus, embedding its extensive dataset—containing technique IDs, names, descriptions, and sub-techniques—into a vector database hosted by Pinecone. The data is chunked into manageable segments for efficient retrieval. When a new piece of CTI is submitted to AIThreatTrack, the system queries the vector database to retrieve the top K (at the time, 5) relevant chunks of the MITRE ATT&CK data. These retrieved chunks, along with the original CTI, are passed to the LLM to extract IOCs and assign them MITRE ATT&CK techniques.

This RAG implementation aimed to address limitations in prior AIThreatTrack iterations, particularly by improving the system’s ability to accurately map IOCs to specific ATT&CK techniques. By enhancing the context available to the LLM, the RAG system was intended to reduce hallucinations and improve accuracy in technique attribution.

3.5 Expected Outcomes

Several key outcomes are anticipated from this thesis, each directly aligned with the research objectives. The first expected outcome is the creation of a dataset that can be leveraged by future researchers to fine-tune or evaluate LLM models for cybersecurity tasks, particularly in generating operationally useful search queries. This dataset provides a valuable benchmark for testing the accuracy and effectiveness of LLMs in a domain-specific context, laying the groundwork for future experimentation and model refinement.

The second outcome relates to the RAG system, which, while not fully optimized in its current form, offers a foundation for future researchers to analyze and iteratively improve upon. This implementation serves as a starting point for refining the integration of retrieval-based systems into threat hunting platforms, with the potential to enhance the accuracy and relevance of LLM-generated results.

Finally, the prototype application developed in this thesis is expected to serve dual purposes. It will act as an educational tool, demonstrating the practical capabilities of LLMs in the threat hunting process. Additionally, it can be extended or further developed into software that directly assists analysts in real-world cybersecurity environments. This prototype showcases how LLMs can be integrated into operational workflows, highlighting their potential to streamline tasks and provide actionable insights in threat hunting.

These outcomes, while specific to the research objectives, also have broader implications for the ongoing development of AI-driven threat hunting tools and the future of LLM applications in cybersecurity.

CHAPTER 4: RESULTS

4.0.1 Retrieval Augmented Generation

4.0.1.1 System Implementation

The RAG system was implemented in AIThreatTrack to reduce LLM hallucinations by providing contextual information through vector space retrieval. The implementation utilized Python with OpenAI, Langchain, and Pinecone libraries, with the MITRE ATT&CK dictionary serving as the knowledge corpus. The system's key implementation features included text chunking using `recursiveCharacterTextSplitter()` with 6000-token chunks and OpenAI's `text-embedding-ada-002` model for embedding generation. Additionally, the system employed modified prompts to leverage the retrieved MITRE dictionary context and utilized a Pinecone vector database for efficient similarity search.

4.0.1.2 Performance Metrics

The system was evaluated across three key metrics:

1. **Extract Hallucination Rate:** The RAG system demonstrated a rate of 35.52% (119 out of 335), showing a slight increase of 0.9% compared to the AIThreatTrack baseline. This metric represents incorrect extractions from CTI samples.
2. **MITRE Hallucination Rate:** The system achieved a rate of 2.99% (10 out of 335), which represents extractions with non-existent MITRE ATT&CK techniques. This is a significant improvement from AIThreatTrack's rate of 47.72%.
3. **Error Rate:** The system showed an error rate of 45.73%, marking an 8.83% increase compared to the AIThreatTrack baseline. This metric represents missing

critical IOCs as determined by the AIThreatTrack baseline.

4.0.1.3 Analysis

The implementation utilized a larger chunk size of 6000 tokens to maximize the likelihood of capturing complete technique descriptions and associated sub-techniques within single chunks. While this approach helped maintain contextual integrity, the performance metrics revealed positive, albeit mixed results.

The minimal increase in extract hallucinations (+0.9%) suggests that the RAG system maintained similar accuracy to the baseline in terms of extraction quality.

Additionally, the low MITRE hallucination rate of 2.99% demonstrates strong accuracy in technique attribution. This performance is only slightly higher (+1.41%) than AIThreatTrack’s final validation step rate of 1.58%. However, AIThreatTrack’s initial extract rate is 47.72% (-44.73% reduction). Further refinement steps bring this down to 1.58%. This large improvement in initial extraction could allow AIThreatTrack’s refinement steps to further eliminate errors.

However, the significant increase in error rate (+8.83%) suggests that the RAG system may be more conservative in its extractions, potentially missing valid IOCs.

These findings indicate that while the RAG system successfully maintains extraction accuracy and technique attribution, it may need refinement to improve its coverage of relevant IOCs.

4.0.2 IOC & KQL Dataset

4.0.2.1 Dataset Overview

The dataset was developed by manually creating Kibana Query Language (KQL) queries for 216 Indicators of Compromise (IOCs) extracted from 71 unique CTI samples. Each query was designed to detect specific behaviors within Windows event logs collected through WinLogBeat and Sysmon. The dataset exhibits several key characteristics:

- Coverage across multiple MITRE ATT&CK techniques
- Event source diversity (Security, Sysmon, PowerShell, etc.)
- Both individual IOC queries and aggregated threat profile searches
- Standardized query structure and format

4.0.2.2 Query Development Methodology

The queries were developed following a systematic approach that consisted of three main phases. The first phase involved individual IOC analysis, where each IOC was carefully analyzed in the context of its source CTI. During this analysis, relevant Windows event types were identified, and queries were tailored to specific behaviors. For instance, when dealing with PowerShell-based threats, queries were designed to target both PowerShell provider logs and Sysmon process creation events.

The second phase focused on establishing query patterns across four key areas:

1. Process Execution: Monitoring specific process names and command lines
2. Network Connections: Tracking specific IPs, domains, and ports
3. File Operations: Monitoring file creations and modifications
4. Registry Changes: Tracking persistence mechanisms and system modifications

In the final phase of aggregate query creation, individual IOC queries were combined into comprehensive threat detection rules while preserving logical relationships between different IOC types. For example, this involved combining PowerShell execution, network connections, and file operations to detect a complete attack chain. All of the queries created are validated against a winlogbeat index in Kibana configured locally.

4.0.2.3 Query Structure Characteristics

The dataset exhibits consistent query patterns across three main areas. First, event provider selection typically includes "Microsoft-Windows-Sysmon/Operational" and "Microsoft-Windows-PowerShell*". Second, event type filtering is implemented through specific winlog.event_id values. Third, message content matching follows the pattern of message: (*pattern1* OR *pattern2*).

Example: ((event.provider : "Microsoft-Windows-PowerShell*" AND message : (*Convert* OR *FromBase64* OR *-enc* OR *hidden* OR *-w* OR *bypass* OR *IEX* OR *Invoke-Expression*)) OR (event.provider : "Microsoft-Windows-Sysmon/Operational" AND (winlog.event_id : "3" AND message : (*445* OR *139* OR SMB*)) OR (winlog.event_id : "10" AND message : (*lsass.exe* OR *mimikatz*))) OR (winlog.channel : "Security" AND winlog.event_id : ("4624" OR "4625" OR "4648" OR "4776" OR "4778")))

4.0.2.4 Dataset Applications

The dataset serves multiple purposes across three key areas. For threat detection, it provides ready-to-use queries for common attack patterns and serves as a baseline for customizing to specific environments. In terms of training and evaluation, the dataset functions as a benchmark for evaluating LLM query generation and provides examples of effective threat hunting queries. Finally, from a research perspective, the dataset enables analysis of IOC-to-query mapping patterns and facilitates the study of threat behavior manifestation in logs.

4.0.3 Demonstration Application

The research implementation culminates in a Python-based web application designed to transform CTI samples into actionable threat detection queries. This demonstration platform accepts input through two primary methods: PDF file upload

and direct text entry. The system processes these inputs to generate validated Elasticsearch DSL queries specifically crafted to identify described behaviors within Windows Event log indices. The application is packaged in an Ubuntu Virtual Machine that includes all of the required infrastructure such as Python and its dependencies, Elasticsearch and Kibana for portability.

The application's architecture leverages several key technologies:

1. Streamlit for web interface development
2. OpenAI for behavior extraction and query generation
3. PDFMiner for document processing
4. Elasticsearch for query validation

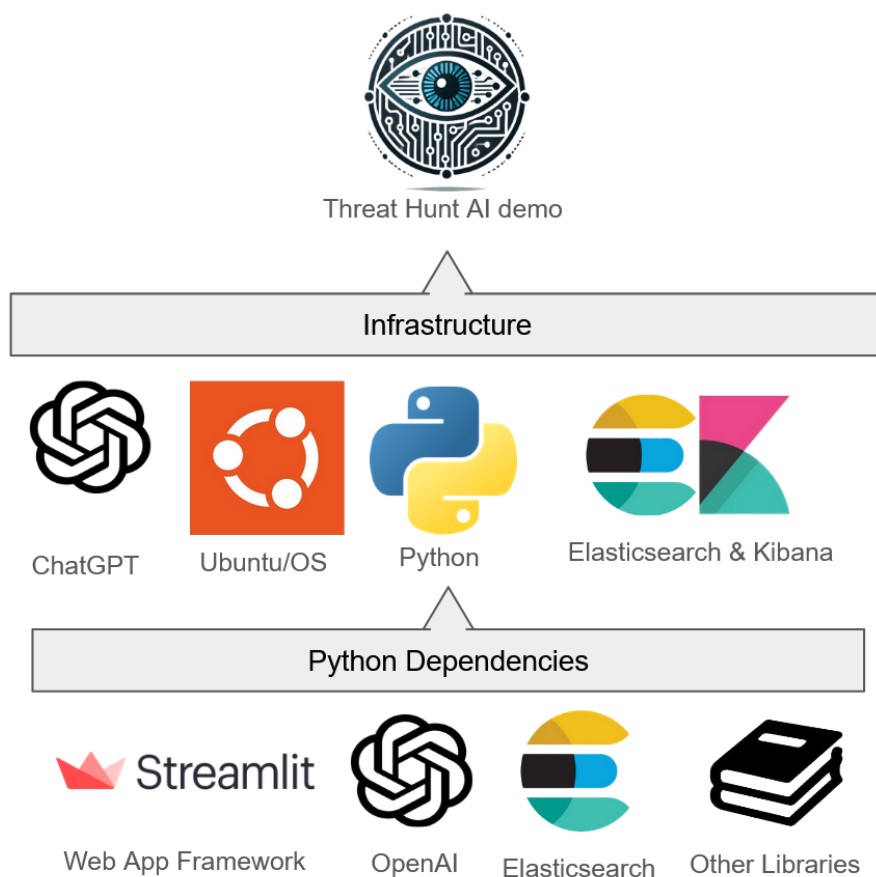


Figure 4.1: Threat Hunting Demo infrastructure

The main application leverages several Python libraries in order to streamline development. Streamlit, Elasticsearch, OpenAI, and PDFMiner all contribute to the operation of the application. Additionally, in order to operate as intended, there is an Elasticsearch and Kibana environment configured with a Windows event log index to validate against. These platforms are installed as services on the host, and will automatically start when the machine is started. This enables the application to connect to the Elasticsearch API as intended.

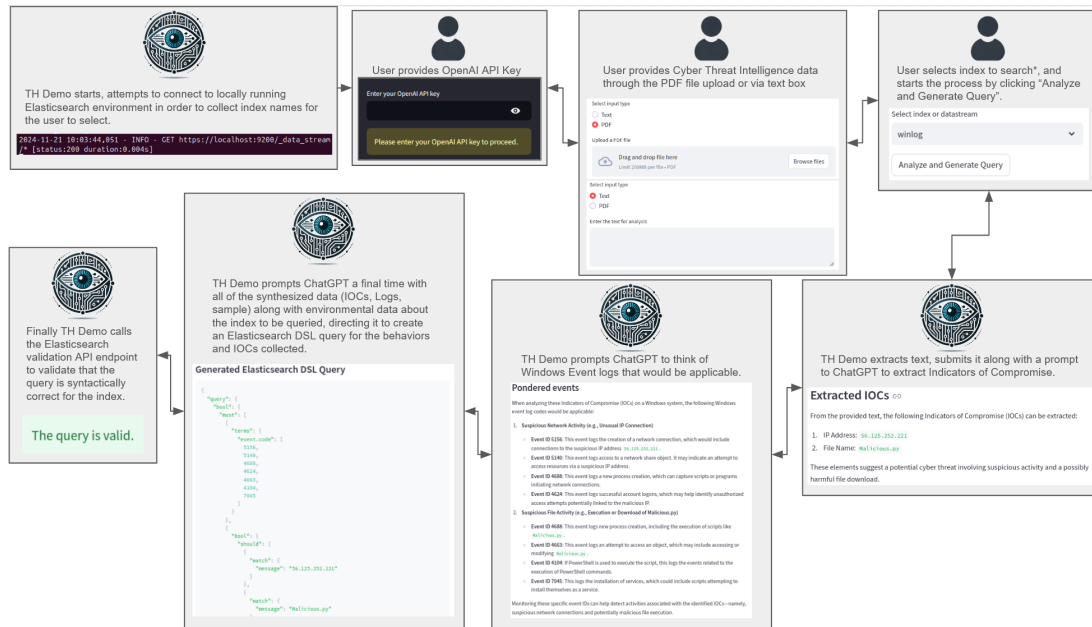


Figure 4.2: TH Demo workflow

By following the application's workflow, users can generate Elasticsearch DSL queries based off of their Cyber Threat Intelligence samples. Even a simple threat intelligence sample can yield complex Elastic DSL queries. As an example, we provide the demo with text CTI: "A suspicious account logged into a host late at night. They connected to: 56.125.252.221 and downloaded a file named: Malicious.py". We select the windows event log index that was found, and click the analyze and generate button. The first prompt to LLM directs it to extract any important indicators of compromise. The LLM returns 2 IOCs:

Extracted IOCs

From the provided text, the following Indicators of Compromise (IOCs) can be extracted:

1. IP Address: `56.125.252.221`
2. File Name: `Malicious.py`

These elements suggest a potential cyber threat involving suspicious activity and a possibly harmful file download.

Figure 4.3: Extracted IOCs

The application immediately continues to the next step, which prompts the LLM to brainstorm what Windows Event log event IDs would be applicable to the IOCs provided.

Pondered events

When analyzing these Indicators of Compromise (IOCs) on a Windows system, the following Windows event log codes would be applicable:

1. **Suspicious Network Activity (e.g., Unusual IP Connection)**
 - **Event ID 5156:** This event logs the creation of a network connection, which would include connections to the suspicious IP address `56.125.252.221`.
 - **Event ID 5140:** This event logs access to a network share object. It may indicate an attempt to access resources via a suspicious IP address.
 - **Event ID 4688:** This event logs a new process creation, which can capture scripts or programs initiating network connections.
 - **Event ID 4624:** This event logs successful account logons, which may help identify unauthorized access attempts potentially linked to the malicious IP.
2. **Suspicious File Activity (e.g., Execution or Download of Malicious.py)**
 - **Event ID 4688:** This event logs new process creation, including the execution of scripts like `Malicious.py`.
 - **Event ID 4663:** This event logs an attempt to access an object, which may include accessing or modifying `Malicious.py`.
 - **Event ID 4104:** If PowerShell is used to execute the script, this logs the events related to the execution of PowerShell commands.
 - **Event ID 7045:** This logs the installation of services, which could include scripts attempting to install themselves as a service.

Monitoring these specific event IDs can help detect activities associated with the identified IOCs—namely, suspicious network connections and potentially malicious file execution.

Figure 4.4: Brainstormed Events

Once again, the application proceeds to the next step, which is the creation of the Elasticsearch DSL query. The LLM is prompted for a final time with all of the contextual information available: context about the Elasticsearch index's set up and configuration, the CTI sample, extracted IOCs, and guidance on what to produce.

Generated Elasticsearch DSL Query

```
{
  "query": {
    "bool": {
      "must": [
        {
          "terms": {
            "event.code": [
              5156,
              5148,
              4688,
              4624,
              4663,
              4184,
              7845
            ]
          }
        },
        {
          "bool": {
            "should": [
              {
                "match": {
                  "message": "56.125.252.221"
                }
              },
              {
                "match": {
                  "message": "Malicious.py"
                }
              }
            ]
          }
        }
      ]
    }
  }
}
```

Figure 4.5: Generated Elasticsearch DSL

In the final step of the process, the application utilizes the local Elasticsearch environment's API to validate that this query is syntactically correct not only for Elasticsearch's DSL syntax, but also validating that the fields that the query filters on are present. The status of this validation is displayed to the user.

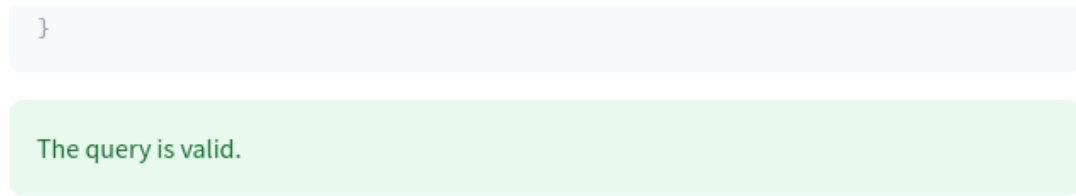


Figure 4.6: Validation

This application, along with the VM that packages all of its dependencies, serves as a useful resource for iterating development of similar projects, swapping out other LLMs to evaluate performance, but also to educate audiences that may be less familiar with this research. This demonstration distills theory into tangible outcomes for users.

CHAPTER 5: DISCUSSION

5.1 Limitations

While this thesis contributes valuable insights and tools to the cybersecurity field, several limitations should be acknowledged.

First, there is no definitive or "perfect" Kibana query for a given IOC. The ideal query depends heavily on various contextual factors, such as the specific CTI being analyzed, the structure of the data being searched, and the environment in which the queries are executed. This inherent variability means that the Kibana queries in the dataset, while accurate and operationally useful, are not universally applicable to all situations.

Second, the Kibana queries created for this research are limited to being executed on a winlogbeat/sysmon index. While winlogbeat is widely used for logging Windows event data, this constraint narrows the generalizability of the queries. Different datasets or indices would require substantial adjustments to the query structure and content, limiting the direct applicability of these queries outside the specific use case.

Additionally, the prototype application developed for this thesis was configured to generate DSL queries specifically for the winlogbeat index, which restricts its flexibility. The current system prompts are tailored to this particular index, and adapting the app to generate queries for other indices would require significant modifications to the underlying prompt structure. This lack of dynamism limits the broader applicability of the tool without further adjustments.

Another limitation is that the application does not execute the generated queries or retrieve search results. While it validates the syntax of the DSL queries against Elasticsearch, it does not assess the operational effectiveness of the queries or the

accuracy of the results they would generate. This limits the scope of the application to syntactical correctness, leaving out the important step of real-world query execution and result evaluation.

Finally, the RAG system implemented in this thesis should not be considered the "definitive" or optimal way to incorporate retrieval-augmented techniques into AI-based threat hunting systems like AIThreatTrack. While it serves as a useful foundation, future iterations could explore alternative architectures, enhanced retrieval methods, or different ways to integrate relevant context into the model. The limitations of the current RAG implementation provide opportunities for further refinement and experimentation.

These limitations provide important context for interpreting the results and highlight areas for improvement in future research, where addressing these constraints could lead to more dynamic, adaptable, and effective cybersecurity tools.

5.2 Suggestions for Future Research

Several avenues for future research emerge from the limitations and outcomes of this thesis, particularly in enhancing LLM performance and expanding the functionality of the demo application. These suggestions offer practical steps for building on the foundation established by this work:

First, one important direction for future research is the development of more robust benchmarks for LLMs as they relate to query generation, similar to existing benchmarks for code generation tasks. A key challenge in cybersecurity is creating datasets that are large, diverse, and accurate enough to reflect real-world scenarios. Building better datasets would involve curating a wider variety of CTI sources, extracting IOCs from them, and ensuring these datasets are representative of various types of cyber threats and environments. By improving benchmarks, future researchers can better evaluate and fine-tune LLMs to generate more effective and operationally useful queries.

Second, future work could focus on iterating on the RAG system within AIThreatTrack. While the current RAG implementation focused on assigning MITRE ATT&CK techniques to IOCs, applying RAG at different stages of the threat hunting process could yield new insights. For example, future research could experiment with alternative chunking methods or adjust the retrieval mechanism to improve relevance and accuracy. Iterating on how RAG is applied, and potentially using it for other purposes within AIThreatTrack, could enhance the system’s overall performance.

Third, extending the functionality of the demo application offers another promising direction. Instead of solely validating DSL queries against the Elasticsearch API, future versions of the application could run the queries and return real search results. Additionally, enhancing the system to dynamically generate queries for different indices—beyond the current winlogbeat index—would significantly expand its utility. Such extensions would make the application more adaptable to a variety of cybersecurity environments, providing a more comprehensive tool for threat hunting.

Fourth, improving the research environment itself could lead to more meaningful results. Currently, the EK stack used for this research collects logs from a personal PC, which limits the realism and applicability of the test cases. Future work could set up a more comprehensive environment, incorporating logs that represent risky or malicious behaviors. Custom CTI describing these behaviors could be generated, providing a more effective testbed for LLMs in creating searches that match real-world threats.

Finally, a broader area for future exploration is incorporating the developed tools into a more complete AI cybersecurity agent’s toolkit. By integrating query generation and retrieval-augmented models into an AI agent, researchers could expand the agent’s capabilities in real-time threat hunting and analysis. This would represent a significant step forward in automating more complex aspects of cybersecurity, further enhancing the role of AI in protecting enterprise environments.

By addressing these areas, future research could build upon the foundations of this thesis, further advancing the efficiency and accuracy of automated threat hunting systems.

CHAPTER 6: CONCLUSIONS

6.1 Summary of the Research Proposal

Cyber threat hunting is a complex and resource-intensive process, requiring skilled analysts and significant time investment. While existing automation tools support aspects of this work, the emergence of more advanced LLMs presents new opportunities for enhancing automation within the threat hunting process. This thesis aims to explore these possibilities by developing tools and methods that harness LLM capabilities in cybersecurity.

The core contributions of this thesis include the creation of a specialized threat hunting dataset, consisting of CTI paired with IOCs and corresponding Kibana queries. This dataset will serve as a valuable benchmark for evaluating and fine-tuning LLMs in generating operationally useful Elasticsearch DSL queries. Additionally, a prototype application has been developed to demonstrate the practical use of LLMs in generating these queries, providing a proof of concept for how LLMs can assist cybersecurity professionals in real-world threat detection scenarios. This prototype is packaged in a Linux-based Virtual Machine that has all necessary infrastructure and dependencies. Lastly, the thesis has implemented a RAG system within AIThreatTrack, a threat hunting system, to further explore how retrieval-based techniques can improve LLM performance in mapping IOCs to MITRE ATT&CK techniques. Most significantly, reducing the rate of first-pass hallucinated MITRE ATT&CK techniques by 44.72%.

Although this research offers promising contributions, it is not without limitations. The complexity of cybersecurity, combined with the challenge of providing LLMs with the extensive context required for accurate query generation, means that fully

automating the threat hunting process remains an ongoing challenge. The current work is constrained by the scope of its dataset and the focused nature of the application, leaving room for future improvements and more comprehensive solutions.

Future research could build on this foundation by creating larger and more diverse benchmark datasets, refining RAG system designs, extending the functionality of the application, and integrating these tools into more dynamic, agent-based workflows. These expansions would further enhance the role of LLMs in automating and optimizing threat hunting processes, ultimately contributing to more effective cybersecurity defense strategies.

REFERENCES

- [1] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, “Attention is all you need.” 31st Conference on Neural Information Processing Systems (NIPS 2017), 2017. Available at <https://arxiv.org/pdf/1706.03762> (accessed 15 August 2024).
- [2] M. Pruba, “Aithreattrack: Towards automated end-to-end threat hunting with generative ai.” PhD Dissertation, 2024. Available by request.
- [3] P. Badva, K. M. Ramokapane, E. Pantano, and A. Rashid, “Unveiling the hunter-gatherers: Exploring threat hunting practices and challenges in cyber defense.” 33rd USENIX Security Symposium, 2024. Available at <https://www.usenix.org/system/files/usenixsecurity24-badva.pdf> (accessed 28 Aug. 2024).
- [4] D. S. Board, “Dsb summer study on special operation and joint forces in support of countering terrorism.” Defense Science Board, 2002. Available at <http://www.fas.org/irp/agency/dod/dsbbrief.ppt> (accessed 25 Aug. 2024).
- [5] P. Gao, F. Shao, X. Liu, X. Xiao, Z. Qin, F. Xu, P. Mittal, S. R. Kulkarni, and D. Song, “Enabling efficient cyber threat hunting with cyber threat intelligence.” IEEE 37th International Conference on Data Engineering (ICDE), 2021. Available at <https://ieeexplore.ieee.org/abstract/document/9458828> (accessed 24 August 2024).
- [6] B. Nour, M. Pourzandi, and M. Debbabi, “A survey on threat hunting in enterprise networks,” 2023. Available at <https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=10216378> (accessed 9 August 2024).
- [7] B. Nour, M. Pourzandi, R. K. Qureshi, and M. Debbabi, “Automa: Automated generation of attack hypotheses and their variants for threat hunting using knowledge discovery.” IEE Transactions on Network and Service Management, 2024. Available at <https://ieeexplore.ieee.org/abstract/document/10477575> (accessed 30 Aug. 2024).
- [8] M. A. Ferrag, F. Alwahedi, A. Battah, B. Cherif, A. Mechri, and N. Tihanyi, “Generative ai and large language models for cyber security: All insights you need.” arXiv, 2024. Available at <https://arxiv.org/abs/2405.12750> (accessed 30 Aug. 2024).
- [9] H. Xu, S. Wang, N. Li, K. Wang, Y. Zhao, K. Chen, T. Yu, Y. Liu, and H. Wang, “Large language models for cyber security: A systematic literature review.” arXiv, 2024. Available at <https://arxiv.org/pdf/2405.04760> (accessed 30 Aug. 2024).

APPENDIX A: Threat Hunting Demo

A.1 Demo Virtual Machine

This appendix outlines specific instructions on how to utilize the Virtual Machine. Provided is an Open Virtualization Format Archive file named LLM_CTH_Demo.ova (7.8 GB). While this format is compatible with many virtualization platforms and architectures, these instructions are specific to VirtualBox on an x86 Windows 10 host.

After downloading and starting VirtualBox, select "File" from the top ribbon. In the menu that appears, select "import appliance".

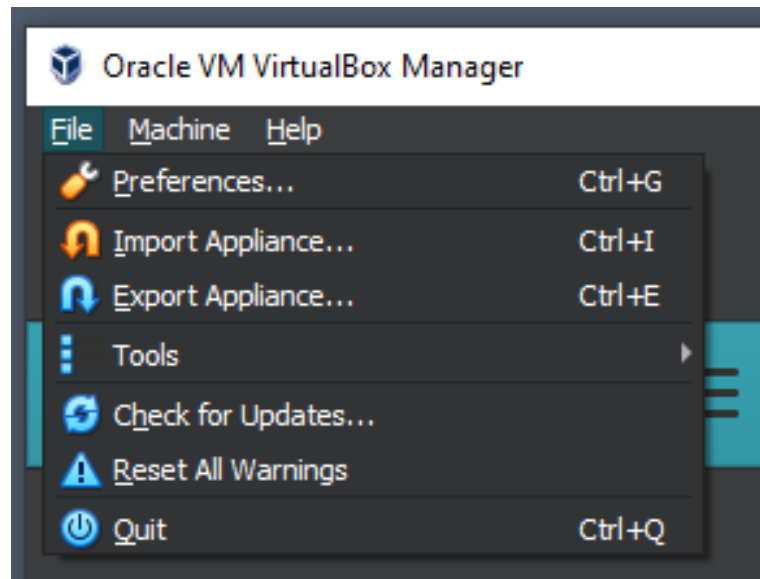


Figure A.1: Import Appliance

A new window will appear. For the "source" field, if it is not already "Local File System", select it. Next, copy and paste the path to the .ova file to the "file" input, or use the file icon to locate the .ova file via the file explorer. Click "next" to proceed.

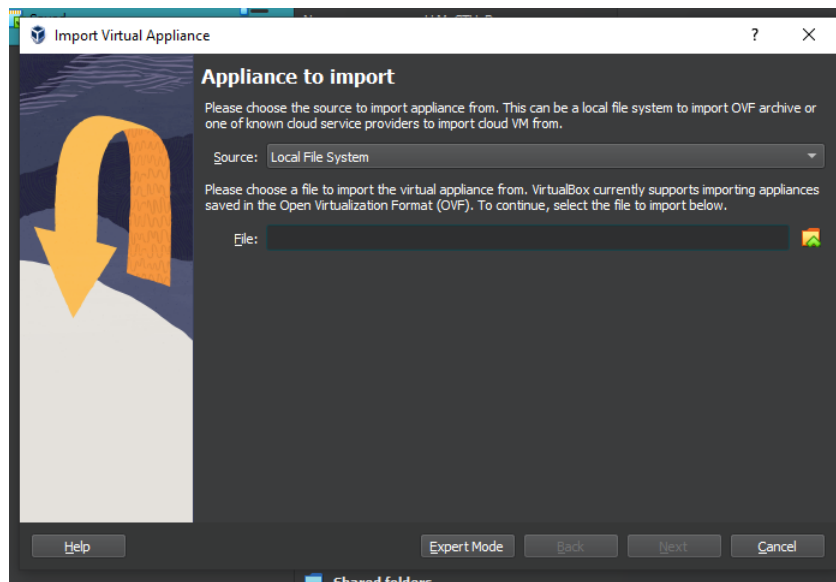


Figure A.2: File Selection

The next page is the "Appliance Settings" page. This will be populated with the settings from the .ova file. Changing these settings may impact operations. Click "Finish" to proceed with importing the VM.

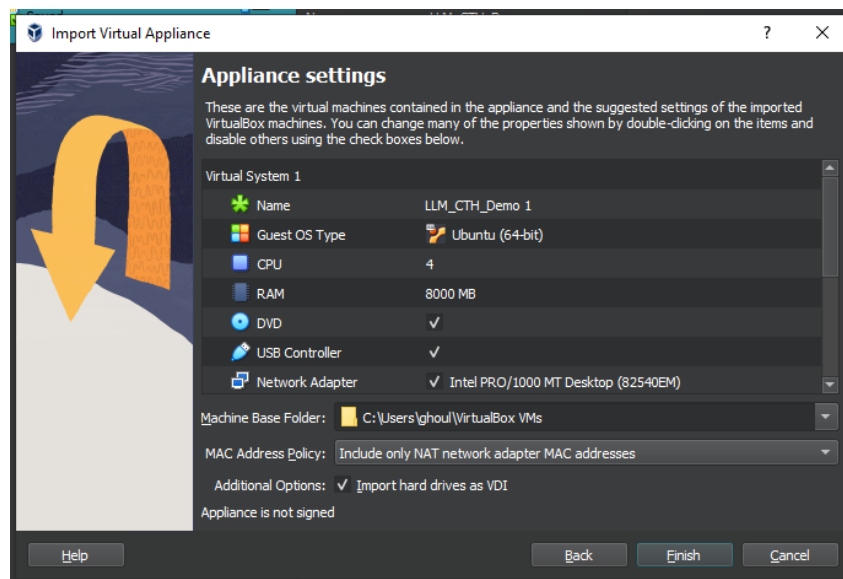


Figure A.3: Application Settings

Once the import is completed, the VM will appear in the left hand column, as an available host. Select the VM, and click the "Start" Green arrow button.

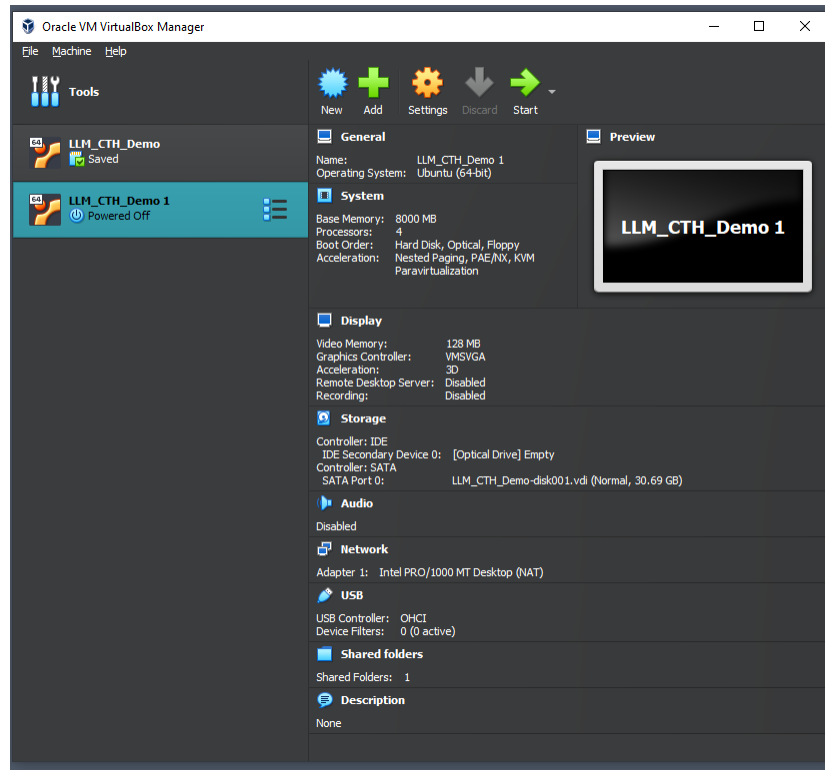


Figure A.4: Completed Import

The VM will start, and you will see the desktop of the host. The password for the user "uncc-ai-cth" is "cyberuncc49". This is the user that is currently logged in when the VM starts.

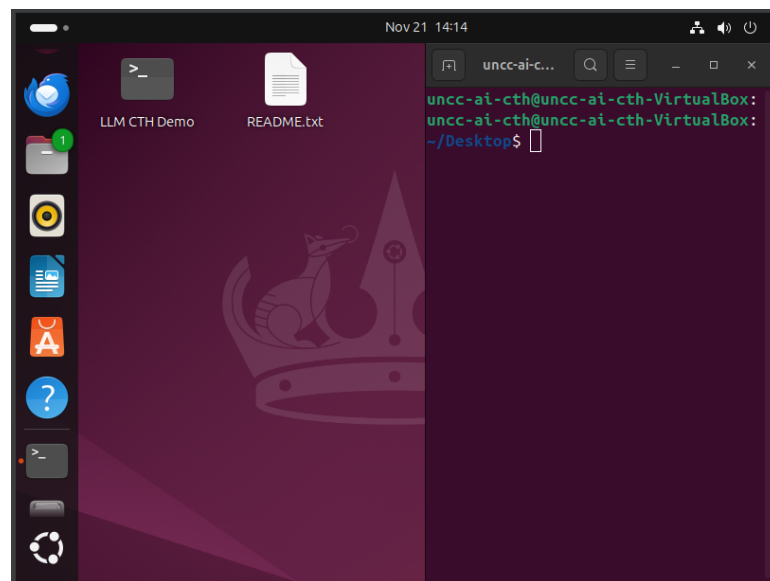
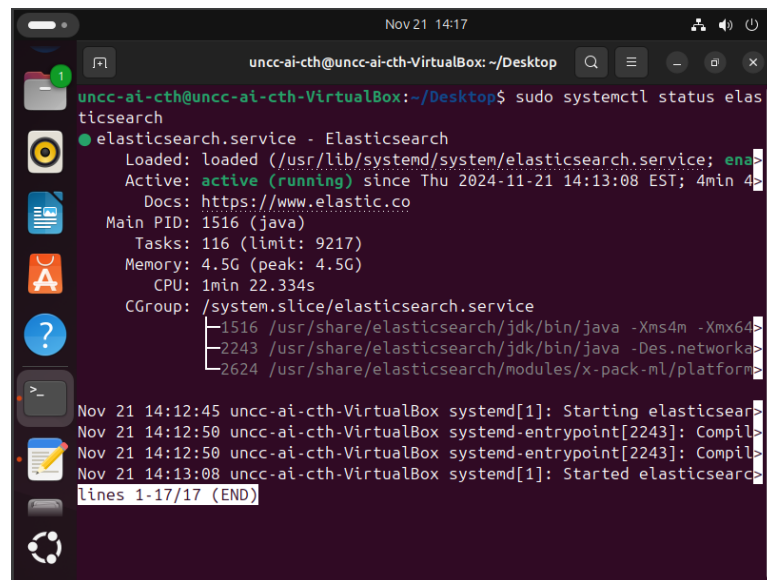


Figure A.5: Booted VM

There are 2 desktop files in the top left corner that are important. First is a .txt file with more instructions, although their content is more than covered here. Second is a terminal shortcut. Double-clicking this icon will run the command to start the Streamlit application. Essentially, this shortcut will start the Threat Hunting Demo application. It is important to let the system start both the Elasticsearch and Kibana services before starting the application. They are both installed as services, and will automatically start on boot. To check on these services, `sudo systemctl status elasticsearch/kibana` will show the current status.



```

Nov 21 14:17
uncc-ai-cth@uncc-ai-cth-VirtualBox: ~/Desktop
uncc-ai-cth@uncc-ai-cth-VirtualBox:~/Desktop$ sudo systemctl status elasticsearch
● elasticsearch.service - Elasticsearch
   Loaded: loaded (/usr/lib/systemd/system/elasticsearch.service; enabled)
   Active: active (running) since Thu 2024-11-21 14:13:08 EST; 4min 4s
     Docs: https://www.elastic.co
   Main PID: 1516 (java)
    Tasks: 116 (limit: 9217)
   Memory: 4.5G (peak: 4.5G)
      CPU: 1min 22.334s
   CGroup: /system.slice/elasticsearch.service
           └─1516 /usr/share/elasticsearch/jdk/bin/java -Xms4m -Xmx64m
             2243 /usr/share/elasticsearch/jdk/bin/java -Des.networkaddress.cache.ttl=60
             2624 /usr/share/elasticsearch/modules/x-pack-ml/platform/linux-x86_64/bin/java

Nov 21 14:12:45 uncc-ai-cth-VirtualBox systemd[1]: Starting elasticsearch.service: Elasticsearch
Nov 21 14:12:50 uncc-ai-cth-VirtualBox systemd-entrypoint[2243]: Compiling search plugins
Nov 21 14:12:50 uncc-ai-cth-VirtualBox systemd-entrypoint[2243]: Compiling plugins
Nov 21 14:13:08 uncc-ai-cth-VirtualBox systemd[1]: Started elasticsearch.service: Elasticsearch
lines 1-17/17 (END)

```

Figure A.6: Checking Elasticsearch Service

After double-clicking the desktop icon, a terminal will appear and will show the console output of the Streamlit app. Additionally, the default web browser Firefox will open to the app's page. The terminal is only for reading logs. The web page is the important part. From this web page you can follow the workflow outlined in the thesis to insert or upload CTI, and get Elasticsearch DSL queries created.

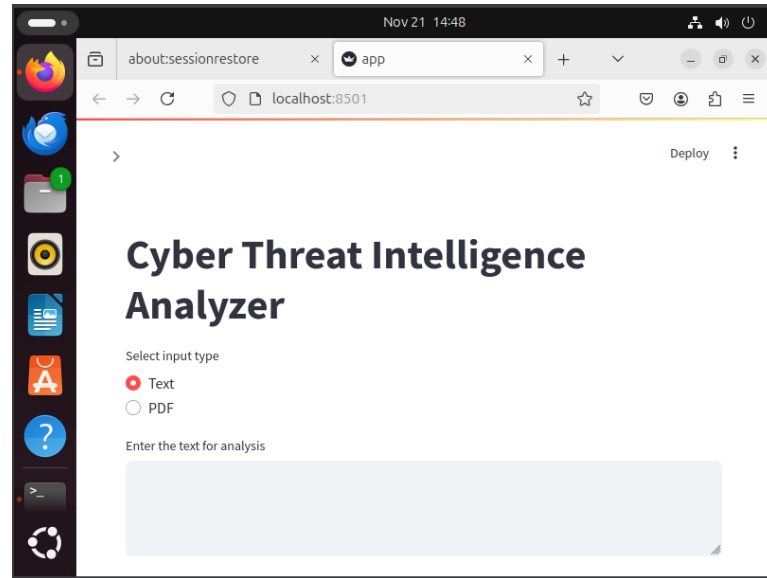


Figure A.7: Threat Hunt Demo Ready

A.2 Demo Application Modification

The current implementation of LLM-based Elasticsearch DSL query generation can be extended and modified. An important theme for this application's development was to be a platform to develop further. This section outline the changes required in order to change what LLM is being interacted with. There are 3 main functions in the application that require interactions with an LLM. They are: `extract_iocs`, `ponder_events`, and `generate_dsl_query`.

```
def extract_iocs(text):
    logging.info("Extracting IOCs using LLM")

    if not client:
        st.error("Please enter a valid OpenAI API key to proceed.")
        return ""

    try:
        response = client.chat.completions.create(
            model="gpt-4o",
            messages=[
                {"role": "system", "content": "You are a helpful assistant skilled in cybersecurity and cyber threat intelligence analysis."},
                {"role": "user", "content": f"Extract Indicators of Compromise (IOCs) from the following text:\n{text}"}
            ]
        )
        iocs = response.choices[0].message.content
        logging.info("IOC extraction successful")
        return iocs
    except Exception as e:
        logging.error(f"Error interacting with LLM: {str(e)}")
        st.error(f"Error interacting with LLM: {str(e)}")
        return ""
```

Figure A.8: `extract_iocs`

```
def ponder_events(ioc):
    logging.info("Pondering Windows event IDs for IOCs")

    if not client:
        st.error("Please enter a valid OpenAI API key to proceed.")
        return ""

    try:
        response = client.chat.completions.create(
            model="gpt-4o",
            messages=[
                {"role": "system", "content": "You are a cybersecurity expert with specific knowledge of windows events, their codes and meanings"},
                {"role": "user", "content": f"Given these indicators of compromise, list the windows event log codes that would be most applicable to"}
            ]
        )
        brainstormed_events = response.choices[0].message.content
        logging.info("events pondering extraction successful")
        return brainstormed_events
    except Exception as e:
        logging.error(f"Error interacting with LLM: {str(e)}")
        st.error(f"Error interacting with LLM: {str(e)}")
        return ""
```

Figure A.9: ponder_events

```
def generate_dsl_query(iocs, pondered_events, index_name, sample_text):
    logging.info("Generating DSL query")

    if not client:
        st.error("Please enter a valid OpenAI API key to proceed.")
        return ""

    prompt = (

    try:
        response = client.chat.completions.create(
            model="gpt-4o",
            messages=[
                {"role": "system", "content": "You are an expert in cybersecurity, Windows Event logs, and Elasticsearch DSL (Domain specific Language)"},
                {"role": "user", "content": prompt}
            ]
        )
        dsl_query = response.choices[0].message.content
        logging.info("DSL query generation successful")
        cleaned_dsl_query = dsl_query.strip('```json').strip('```')
        return cleaned_dsl_query
    except Exception as e:
        logging.error(f"Error generating DSL query: {str(e)}")
        st.error(f"Error generating DSL query: {str(e)}")
        return ""
```

Figure A.10: generate_dsl_query

The current implementation uses OpenAI's Python library to access the API. This connection is made after providing the API key. In each of these functions, the model to use is set in the "model" variable. To change the model to another OpenAI LLM, simply updating these values to the specific model is all that is needed. To utilize other models, these functions will need to be updated with new calls. These new models may not be compatible with OpenAI's Python library. In that case, the changes to the code will extend to the authentication mechanisms. These functions return the outputs of the LLMs and displays them to the user via Streamlit. This allows the specific calls to the LLMs to be changed, while the front end remains flexible.

APPENDIX B: KQL Dataset

The following is the content of the KQL dataset in a listed form in this format:

1. Cyber Threat Intel Sample

- (a) Extracted Indicator of Compromise 1:

<IOC KQL search 1>

- (b) Extracted Indicator of Compromise 2:

<IOC KQL search 2>

- (c) Extracted Indicator of Compromise N:

<IOC KQL search N>

- (d) Aggregate Cyber Threat Intel Search:

<Aggregate search>

- (e) Explanation of search methods

It is made up of the Cyber Threat Intelligence sample, then separated out by the extracted Indicator of Compromises and their associated attributes like the KQL query that would search for them, along with an explanation. Finally, the entire CTI sample has an aggregate KQL query.

1. Another sophisticated #heavily #obfuscated #powershell: contains three unseen #embedded #PE (mimi, mon, vcp (msvcp120.dll?) all UNDETECTED), uses #SMB exploit (#MS17_010) Extract user creds, attempts to con to neighbors (established cons). If not possible, uses a #SMB #exploit

- (a) PowerShell (PowerShell (T1059.001)):

(event.provider : "Microsoft-Windows-PowerShell*" AND message : (*Convert* OR *FromBase64* OR *-enc* OR *hidden* OR *-w* OR *bypass* OR *IEX* OR *Invoke-Expression*))

- (b) extract user creds (Exploitation for Credential Access (T1212)):
- ```
(event.provider : "Microsoft-Windows-Sysmon/Operational"
AND (winlog.event_id : "3" AND message : (*445* OR *139* OR
SMB*)) OR (winlog.event_id : "10" AND message : *lsass.exe* OR
mimikatz))
```
- (c) attempts to con to neighbors (Valid Accounts (T1078)):
- ```
(winlog.channel : "Security" AND winlog.event_id : ("4624" OR "4625"
OR "4648" OR "4776" OR "4778"))
```
- (d) Aggergate KQL Query:
- ```
((event.provider : "Microsoft-Windows-PowerShell*" AND
message : (*Convert* OR *FromBase64* OR *-enc* OR *hidden* OR *-
w* OR *bypass* OR *IEX* OR *Invoke-Expression*)) OR (event.provider
: "Microsoft-Windows-Sysmon/Operational"
AND (winlog.event_id : "3" AND message : (*445* OR *139* OR
SMB*)) OR (winlog.event_id : "10" AND message : (*lsass.exe* OR
mimikatz))) OR (winlog.channel : "Security" AND winlog.event_id :
("4624" OR "4625" OR "4648" OR "4776" OR "4778")))
```
- (e) For the powershell IOC, the search is focused on that event provider. Given the high obfuscation described, we focus further on attempting to match against some commonly used methods of obfuscation like base64 encoding. Credential extraction is more focused on capturing attempts to use common cred extract tools like mimikatz. Searching for connections to neighbors looks for several events that all relate to users logging on and connections being made. Finally, these partial searches are joined via OR operators in order to get a wholistic view of the CTI IOCs.

2. #docx contains #simple #vba script [bare minimum obfuscation]. Runs a fresh

#malware instance that is not detected by AVs #PE:

hxxp://185.121.139.238/~payments/background.png

(a) hxxp://185.121.139.238/ payments/background.png

(Deobfuscate/Decode Files or Information (T1140)):

event.provider : "Microsoft-Windows-Sysmon/Operational"

AND winlog.event\_id : "3" AND message : "185.121.139.238"

(b) hxxp://185.121.139.238/ payments/background.png

(User Execution (T1204)):

event.provider : "Microsoft-Windows-Sysmon/Operational"

AND winlog.event\_id : "3" AND message : (\*payments\* AND \*background.png\*)

(c) Aggergate KQL Query:

( (event.provider : "Microsoft-Windows-Sysmon/Operational"

AND ( (winlog.event\_id : "3" AND message : ("185.121.139.238" OR (\*payments\* AND \*background.png\*))) OR (winlog.event\_id : "1" AND

message : (\*winword.exe\* OR \*WINWORD.EXE\*)) ) ) OR

(event.provider : "Microsoft-Windows-Windows Defender/Operational"

AND message : ("185.121.139.238" OR \*payments/background.png\*) ) )

(d) The queries focus on detecting network connections to the malicious URL and monitoring for specific file patterns in the payload. The aggregate query also includes process monitoring for Word execution and Windows Defender alerts related to the suspicious URL and file patterns.

3. complex #malware: #docx (external #Relationship - type #oleObject) -> #rtf contains 10 #ole -> each has #package (#xlsx) -> #vba macro dls #vbs -> #powershell -> #NET. It compiles embedded c# code dynamically invokes.

The code decrypts data stored as a #resource and calls it

(a) docx (Obfuscated Files or Information (T1027)):

event.provider : "Microsoft-Windows-Sysmon/Operational"

AND

winlog.event\_id : ("1" OR "11") AND message : \*.docx\*

(b) rtf (Obfuscated Files or Information (T1027)):

event.provider : "Microsoft-Windows-Sysmon/Operational"

AND

winlog.event\_id : ("1" OR "11") AND message : \*.rtf\*

(c) ole (Obfuscated Files or Information (T1027)):

event.provider : "Microsoft-Windows-Sysmon/Operational"

AND

winlog.event\_id : ("1" OR "11") AND message : \*.ole\*

(d) xlsx (Obfuscated Files or Information (T1027)):

event.provider : "Microsoft-Windows-Sysmon/Operational"

AND

winlog.event\_id : ("1" OR "11") AND message : \*.xlsx\*

(e) vba (Obfuscated Files or Information (T1027)):

event.provider : "Microsoft-Windows-Sysmon/Operational"

AND

winlog.event\_id : ("1" OR "11") AND message : \*.vba\*

(f) vbs (Obfuscated Files or Information (T1027)):

event.provider : "Microsoft-Windows-Sysmon/Operational"

AND

winlog.event\_id : ("1" OR "11") AND message : \*.vbs\*

(g) powershell (Obfuscated Files or Information (T1027)):

event.provider : ("Microsoft-Windows-PowerShell\*" OR  
 "Windows PowerShell") AND message : (\*FromBase64\* OR \*encoded\*  
 OR \*hidden\*)

- (h) NET (Obfuscated Files or Information (T1027)):

event.provider : "Microsoft-Windows-Sysmon/Operational"  
 AND  
 winlog.event\_id : ("1" OR "11") AND message : (\*.dll\* OR \*.exe\*)

- (i) c# (Obfuscated Files or Information (T1027)):

event.provider : "Microsoft-Windows-Sysmon/Operational"  
 AND  
 winlog.event\_id : ("1" OR "11") AND message : \*.cs\*

- (j) docx (Resource Hijacking (T1496)):

event.provider : "Microsoft-Windows-Sysmon/Operational"  
 AND  
 winlog.event\_id : "1" AND message :  
 (\*winword.exe\* OR \*WINWORD.EXE\*)

- (k) vba (Resource Hijacking (T1496)):

event.provider : "Microsoft-Windows-Sysmon/Operational"  
 AND  
 winlog.event\_id : "1" AND message : (\*winword.exe\* OR  
 WINWORD.EXE\* OR \*excel.exe\* OR \*EXCEL.EXE\*)

- (l) resource (Obfuscated Files or Information (T1027)):

event.provider : "Microsoft-Windows-Sysmon/Operational"  
 AND  
 winlog.event\_id : ("11" OR "7") AND message : \*resource\*

- (m) vba (Command and Scripting Interpreter (T1059)):

event.provider : "Microsoft-Windows-Sysmon/Operational"

AND

winlog.event\_id : "1" AND message : (\*winword.exe\* OR  
WINWORD.EXE\* OR \*excel.exe\* OR \*EXCEL.EXE\*)

(n) vbs (Command and Scripting Interpreter (T1059)):

event.provider : "Microsoft-Windows-Sysmon/Operational"

AND winlog.event\_id : "1" AND message : (\*wscript.exe\* OR  
cscript.exe\* OR

WSSCRIPT.EXE\* OR \*CSCRIPT.EXE\*)

(o) powershell (Command and Scripting Interpreter (T1059)):

event.provider : ("Microsoft-Windows-PowerShell\*" OR "Windows Pow-  
erShell") AND message : (\*IEX\* OR \*Invoke-Expression\* OR \*From-  
Base64\* OR \*Convert\*)

(p) c# (Command and Scripting Interpreter (T1059)):

event.provider : "Microsoft-Windows-Sysmon/Operational"

AND

winlog.event\_id : "1" AND message : (\*csc.exe\* OR \*CSC.EXE\*)

(q) NET (Native API (T1106)):

event.provider : "Microsoft-Windows-Sysmon/Operational"

AND

winlog.event\_id : ("7" OR "8") AND message :

(\*System.Runtime.InteropServices\* OR \*mscorlib.dll\* OR \*mscorlib.dll\*  
OR \*clr.dll\*)

(r) Aggergate KQL Query:

( (event.provider : "Microsoft-Windows-Sysmon/Operational"

AND ( (winlog.event\_id : ("1" OR "11") AND message : ( \*.docx\* OR



```
.rtf OR *.ole* OR *.xlsx* OR *.vba* OR *.vbs* OR *.dll* OR *.exe* OR
.cs)) OR (winlog.event_id : "1" AND message : (*winword.exe* OR
WINWORD.EXE OR *excel.exe* OR *EXCEL.EXE* OR *wscript.exe*
OR
WSCRIPT.EXE* OR *cscript.exe* OR *CSCRIPT.EXE* OR *csc.exe*
OR *CSC.EXE* OR *InstallUtil* OR *RegAsm* OR *RegSvcs* OR *MS-
Build*)) OR (winlog.event_id : ("7" OR "8") AND message : (*Sys-
tem.Runtime.InteropServices* OR *mscorlib.dll* OR *mscorlib.dll* OR
clr.dll)) OR (winlog.event_id : ("11" OR "7") AND message : *re-
source*))) OR (event.provider : ("Microsoft-Windows-PowerShell*" OR
"Windows PowerShell") AND message : (*FromBase64* OR *encoded*
OR *hidden* OR *IEX* OR *Invoke-Expression* OR *Convert*)))
```

- (s) The queries systematically track each stage of this sophisticated malware chain. Starting with document handling, we monitor for various Microsoft Office processes and file operations related to DOCX, RTF, and XLSX files. For the scripting stages, we look for VBA execution in Office processes, VBScript execution via wscript/cscript, and PowerShell activity with special attention to encoding and obfuscation indicators. The .NET and C components are tracked through relevant DLL loading, process creation, and resource manipulation, particularly focusing on dynamic compilation and assembly loading. Registry monitoring helps detect persistence attempts, while the aggregate query combines all these elements to provide full visibility across the entire attack chain.

4. A small potentially #malicious #net gets a list of usernames determines which one of them are registered on #twitter and then checks whether any of the following emails are used b4 on #twitter (username@#hotmail #gmail #yahoo #outlook ). Using 2 #undocumented #twitter #api

- (a) username@#hotmail (Account Discovery (T1087)):
  - event.provider : "Microsoft-Windows-Sysmon/Operational"
  - AND winlog.event\_id : "3" AND message : (\*@hotmail\* OR hotmail.com\*)
- (b) username@#gmail (Account Discovery (T1087)):
  - event.provider : "Microsoft-Windows-Sysmon/Operational"
  - AND winlog.event\_id : "3" AND message : (\*@gmail\* OR \*gmail.com\*)
- (c) username@#yahoo (Account Discovery (T1087)):
  - event.provider : "Microsoft-Windows-Sysmon/Operational"
  - AND winlog.event\_id : "3" AND message : (\*@yahoo\* OR yahoo.com\*)
- (d) username@#outlook (Account Discovery (T1087)):
  - event.provider : "Microsoft-Windows-Sysmon/Operational"
  - AND winlog.event\_id : "3" AND message : (\*@outlook\* OR outlook.com\*)
- (e) Aggregate KQL Query:
  - ( event.provider : "Microsoft-Windows-Sysmon/Operational"
  - AND winlog.event\_id : "3" AND message : ( \*@outlook\* OR \*outlook.com\* OR \*@yahoo\* OR \*yahoo.com\* OR \*@gmail\*
  - OR \*gmail.com\* OR \*@hotmail\* OR \*hotmail.com\* ) )
- (f) The queries are designed to detect potential email enumeration and account discovery activities. They focus on network connections that contain email patterns from major providers (Gmail, Hotmail, Yahoo, Outlook) which could indicate attempts to validate email addresses against Twitter's API. The Sysmon event ID 3 is used to capture these network connections, with

specific patterns in the message field to match email-related strings. This approach allows us to detect both the reconnaissance activity of checking usernames and the subsequent correlation with email services, all through monitoring outbound network connections. The aggregate query combines these patterns to provide comprehensive visibility into potential user enumeration attempts across multiple email providers.

5. Interesting #malicious #JScript kills processes listed by a remote server and then dels them from filesystem. Next it dls the malware(s) specified by the remote server. Then it uses #regsrv32 to register a bunch of #dll including a #dll on a REMOTE server (unavailable)!

- (a) JScript (Process Discovery (T1057)):

event.provider : "Microsoft-Windows-Sysmon/Operational"

AND

winlog.event\_id : "1" AND message : (\*jscript\* OR

WSSCRIPT.EXE\* OR \*cscript.exe\* OR

WSSCRIPT.EXE\* OR \*CSCRIPT.EXE\*)

- (b) kills processes (Process Discovery (T1057)):

event.provider : "Microsoft-Windows-Sysmon/Operational"

AND

winlog.event\_id : ("1" OR "10") AND message : (\*taskkill\* OR \*Stop-Process\* OR \*kill\* OR \*terminate\*)

- (c) regsrv32 (Process Injection (T1055)):

event.provider : "Microsoft-Windows-Sysmon/Operational"

AND

winlog.event\_id : "1" AND message : (\*regsvr32\* OR \*REGSVR32\*)

- (d) register a bunch of dll (Process Injection (T1055)):

event.provider : "Microsoft-Windows-Sysmon/Operational"

AND ( (winlog.event\_id : ("7" OR "8") AND message : \*.dll\*) OR (winlog.event\_id : "1" AND message : (\*regsvr32\* OR \*REGSVR32\*)) )

- (e) dll on a REMOTE server (Dynamic-link Library Injection (T1055.001)):

event.provider : "Microsoft-Windows-Sysmon/Operational"

AND ( (winlog.event\_id : "3" AND message : (\*.dll\* AND (\*http\* OR \*https\* OR \*\\\\\\\*))) OR (winlog.event\_id : ("7" OR "8") AND message : (\*\\\\\\\* OR \*http\* OR \*https\*/)) )

- (f) Aggregate KQL Query:

event.provider : "Microsoft-Windows-Sysmon/Operational"

AND ( (winlog.event\_id : "1" AND message : ( \*jscript\* OR WSCRIPT.EXE\* OR \*cscript.exe\* OR WSCRIPT.EXE\* OR \*CSCRIPT.EXE\* OR \*taskkill\* OR \*Stop-Process\* OR \*kill\* OR \*terminate\* OR \*regsvr32\* OR REGSVR32\* ) ) OR (winlog.event\_id : ("7" OR "8") AND message : ( \*.dll\* OR (\*\\\\\\\* OR \*http\* OR \*https\*) ) ) OR (winlog.event\_id : "3" AND message : (\*.dll\* AND (\*http\* OR \*https\* OR \*\\\\\\\*)) ) OR (winlog.event\_id : "10" AND message : (\*taskkill\* OR \*Stop-Process\* OR \*kill\* OR \*terminate\*) ) )

- (g) The queries are structured to detect multiple malicious behaviors in this attack chain. First, they monitor for JScript execution through common script hosts (wscript/cscript). Process termination commands are tracked to detect the killing of processes. DLL-related activities are monitored through multiple angles: regsvr32 execution for DLL registration, DLL loading events, and network connections involving DLLs (particularly from remote sources). The aggregate query combines these detections to provide visibility across the entire attack flow, from initial script execution through

process termination and DLL operations, with special attention to remote DLL loading which is a particularly suspicious behavior.

6. RT @SevenLayerJedi: Another #diamondfox control panel zip opendir

meow://www.savntown[.]com/ControlPanel.zip CC

@hasherezade @James\_inthe\_box

(a) www.savntown[.]com (Exfiltration Over Web Service (T1567)):

event.provider : "Microsoft-Windows-Sysmon/Operational"

AND winlog.event\_id : "3" AND message : \*savntown\*

(b) meow://www.savntown[.]com/ControlPanel.zip (Exfiltration Over Web Service (T1567)):

event.provider : "Microsoft-Windows-Sysmon/Operational"

AND winlog.event\_id : "3" AND message : (\*meow\* AND \*savntown\* AND \*ControlPanel.zip\*)

(c) ControlPanel.zip (Data Encrypted for Impact (T1486)):

event.provider : "Microsoft-Windows-Sysmon/Operational"

AND ( (winlog.event\_id : ("11" OR "15") AND message : \*ControlPanel.zip\*) OR (winlog.event\_id : "1" AND message : (\*unzip\* OR \*expand\* OR \*7z\* OR \*winzip\* OR \*winrar\*)) )

(d) Aggregate KQL Query:

event.provider : "Microsoft-Windows-Sysmon/Operational"

AND ( (winlog.event\_id : "3" AND message : ( \*savntown\* OR (\*meow\* AND \*savntown\* AND \*ControlPanel.zip\*) ) ) OR (winlog.event\_id : ("11" OR "15") AND message : \*ControlPanel.zip\* ) OR (winlog.event\_id : "1" AND message : (\*unzip\* OR \*expand\* OR \*7z\* OR \*winzip\* OR \*winrar\*)) ) )

(e) The queries track both the network and file system activities related to this

malware panel. Network monitoring focuses on connections to the suspicious domain, particularly those using the non-standard "meow" protocol scheme. File operations are monitored for the specific ControlPanel.zip file, along with common archive handling processes that might be used to extract it. The aggregate query combines these elements to detect both the download attempt and any subsequent interaction with the control panel archive, providing coverage across the full attack path from initial download to potential execution.

## 7. #Malware

hxxp://lalecitinadesoja.com/imagenesdeunasdisenos.com/files/ #MalwareMust-Die

(a) hxxp://lalecitinadesoja.com/imagenesdeunasdisenos.com/files/

(Drive-by Compromise (T1189)):

event.provider : "Microsoft-Windows-Sysmon/Operational"

AND winlog.event\_id : "3" AND message : (\*lalecitinadesoja\* OR \*imagenesdeunasdisenos\*)

(b) Aggregate KQL Query:

event.provider : "Microsoft-Windows-Sysmon/Operational"

AND winlog.event\_id : "3" AND message : (\*lalecitinadesoja\* OR \*imagenesdeunasdisenos\*)

(c) The query focuses on network connections (Event ID 3) to detect any attempts to access the malicious domains. The search patterns look for both parts of the URL structure, as the domain contains a subdirectory that itself looks like a domain name. This approach helps detect the drive-by compromise attempt regardless of which part of the URL path is being accessed. The same query serves both as the individual and aggregate

search since we're monitoring a single, specific indicator.

8. Maybe it is better to say: part of the shellcode is similar to the metasploit one as it is not exactly the same.

- (a) shellcode (Command and Scripting Interpreter (T1059)):

```
event.provider : "Microsoft-Windows-Sysmon/Operational"
AND ((winlog.event_id : ("7" OR "8") AND message : (*VirtualAlloc* OR *VirtualProtect* OR *RtlMoveMemory* OR *WriteProcessMemory*)) OR (winlog.event_id : "10" AND message : (*OpenProcess* OR *CreateRemoteThread* OR *NtCreateThreadEx*)))
```

- (b) metasploit (Command and Scripting Interpreter (T1059)):

```
event.provider : "Microsoft-Windows-Sysmon/Operational"
AND ((
winlog.event_id : "1" AND message : (*meterpreter* OR *metasploit* OR *msf* OR *reverse_tcp* OR *reverse_https*)) OR (winlog.event_id : "3" AND message : (*4444* OR *4445* OR *5555* OR *5556*)))
```

- (c) Aggregate KQL Query:

```
event.provider : "Microsoft-Windows-Sysmon/Operational"
AND ((winlog.event_id : ("7" OR "8") AND message : (*VirtualAlloc* OR *VirtualProtect* OR *RtlMoveMemory* OR *WriteProcessMemory*)) OR (winlog.event_id : "10" AND message : (*OpenProcess* OR *CreateRemoteThread* OR *NtCreateThreadEx*)) OR (winlog.event_id : "1" AND message : (*meterpreter* OR *metasploit* OR *msf* OR *reverse_tcp* OR *reverse_https*)) OR (winlog.event_id : "3" AND message : (*4444* OR *4445* OR *5555* OR *5556*)))
```

- (d) The queries focus on two main aspects of shellcode and Metasploit detection. First, they monitor for common Windows API calls associated with

shellcode execution, such as memory allocation and process manipulation. Second, they look for specific Metasploit artifacts, including process names, common payload types, and default port numbers. The aggregate query combines these approaches to detect both generic shellcode behavior and specific Metasploit indicators, providing broader coverage for cases where the shellcode might be partially modified from its original Metasploit form.

9. Small `#powershell` code injects a `#shellcode` (which seems to be a `#backdoor`) in itself and executes the injected code. Part of the shellcode is the `#metasploit #bind_tcp #payload` listens on port 4444. The rest seems junk.

- (a) powershell (Process Injection (T1055)):

```
event.provider : "Microsoft-Windows-Sysmon/Operational"
AND ((winlog.event_id : "1" AND message : (*powershell* OR
POWERSHELL.EXE*)) OR (event.provider :
("Microsoft-Windows-PowerShell*" OR "Windows PowerShell") AND mes-
sage : (*Invoke-Expression* OR *IEX* OR *FromBase64* OR *-enc*)))
```

- (b) shellcode (Process Injection (T1055)):

```
event.provider : "Microsoft-Windows-Sysmon/Operational"
AND ((winlog.event_id : ("7" OR "8") AND message : (*VirtualAl-
loc* OR *VirtualProtect* OR *RtlMoveMemory* OR *WriteProcessMem-
ory*)) OR (winlog.event_id : "10" AND message : (*OpenProcess* OR
CreateRemoteThread OR *NtCreateThreadEx*)))
```

- (c) backdoor (Process Injection (T1055)):

```
event.provider : "Microsoft-Windows-Sysmon/Operational"
AND ((winlog.event_id : "1" AND message : (*cmd.exe* OR *CMD.EXE*
OR *powershell* OR *POWERSHELL.EXE*)) OR (winlog.event_id :
"3" AND message : (*4444* OR *bind* OR *reverse*)))
```



- (d) metasploit (Process Injection (T1055)):

event.provider : "Microsoft-Windows-Sysmon/Operational"

AND ( (winlog.event\_id : "1" AND message : (\*meterpreter\* OR \*metasploit\* OR \*msf\* OR \*bind\_tcp\*)) OR (winlog.event\_id : "3" AND message : \*4444\*) )

- (e) bind\_tcp (Process Injection (T1055)):

event.provider : "Microsoft-Windows-Sysmon/Operational"

AND winlog.event\_id : "3" AND message : (\*bind\* AND \*4444\*)

- (f) Aggregate KQL Query:

( event.provider : "Microsoft-Windows-Sysmon/Operational"

AND ( (winlog.event\_id : "1" AND message : (\*powershell\* OR \*POWERSHELL.EXE\* OR \*cmd.exe\* OR \*CMD.EXE\* OR \*meterpreter\* OR \*metasploit\* OR \*msf\* OR \*bind\_tcp\*)) OR (winlog.event\_id : ("7" OR "8") AND message : (\*VirtualAlloc\* OR \*VirtualProtect\* OR \*RtlMoveMemory\* OR \*WriteProcessMemory\*)) OR (winlog.event\_id : "10" AND message : (\*OpenProcess\* OR \*CreateRemoteThread\* OR \*NtCreateThreadEx\*)) OR (winlog.event\_id : "3" AND message : (\*4444\* OR \*bind\*)) ) ) OR (event.provider : ("Microsoft-Windows-PowerShell" OR "Windows PowerShell") AND message : (\*Invoke-Expression\* OR \*IEX\* OR \*FromBase64\* OR \*-enc\*))

- (g) The queries are designed to detect multiple aspects of this PowerShell-based attack chain. They monitor for PowerShell execution with common obfuscation techniques, shellcode injection through memory manipulation APIs, and specific Metasploit bind\_tcp payload indicators. Network connections are monitored for the typical port 4444 and bind shell patterns. The PowerShell-specific monitoring includes common obfuscation and encoding methods often used to hide malicious code. Memory operations

typically associated with shellcode injection are tracked through various Windows APIs. The aggregate query combines all these elements to provide comprehensive visibility across the entire attack chain, from initial PowerShell execution through shellcode injection and network callback attempts.

10. `hxxp://118.184.48.95:8000/in3.ps1`

`104.148.42[.]153:8000/in3.ps1`

`107.179.67[.]243:8000/in3.ps1`

`172.247.116[.]8:8000/in3.ps1`

`http://172.247.116.87:8000/re6.ps1` (windows servers)

(a) `118.184.48.95:8000/in3.ps1` (Web Service (T1102)):

`event.provider : "Microsoft-Windows-Sysmon/Operational"`

`AND winlog.event_id : "3" AND message : (*118.184.48.95* AND *8000* AND *in3.ps1*)`

(b) `104.148.42.153:8000/in3.ps1` (Web Service (T1102)):

`event.provider : "Microsoft-Windows-Sysmon/Operational"`

`AND winlog.event_id : "3" AND message : (*104.148.42.153* AND *8000* AND *in3.ps1*)`

(c) `107.179.67.243:8000/in3.ps1` (Web Service (T1102)):

`event.provider : "Microsoft-Windows-Sysmon/Operational"`

`AND winlog.event_id : "3" AND message : (*107.179.67.243* AND *8000* AND *in3.ps1*)`

(d) `172.247.116.8:8000/in3.ps1` (Web Service (T1102)):

`event.provider : "Microsoft-Windows-Sysmon/Operational"`

`AND winlog.event_id : "3" AND message : (*172.247.116.8* AND *8000* AND *in3.ps1*)`

(e) 172.247.116.87:8000/re6.ps1 (Web Service (T1102)):

event.provider : "Microsoft-Windows-Sysmon/Operational"

AND winlog.event\_id : "3" AND message : (\*172.247.116.87\* AND \*8000\* AND \*re6.ps1\*)

(f) Aggregate KQL Query:

event.provider : "Microsoft-Windows-Sysmon/Operational"

AND ( (winlog.event\_id : "3" AND message : ( (\*118.184.48.95\* AND \*8000\* AND \*in3.ps1\*) OR (\*104.148.42.153\* AND \*8000\* AND \*in3.ps1\*) OR (\*107.179.67.243\* AND \*8000\* AND \*in3.ps1\*) OR (\*172.247.116.8\* AND \*8000\* AND \*in3.ps1\*) OR (\*172.247.116.87\* AND \*8000\* AND \*re6.ps1\*) ) ) OR (winlog.event\_id : ("1" OR "11") AND message : (\*in3.ps1\* OR \*re6.ps1\*)) )

(g) The queries target network connections to multiple suspicious IP addresses serving PowerShell scripts. Each IP is monitored specifically for connections on port 8000 requesting either 'in3.ps1' or 're6.ps1'. The monitoring includes both the network connection attempts and potential file creation or execution of these scripts. The aggregate query combines all these indicators while also watching for any file operations involving these specific PowerShell script names, providing coverage for both the download and execution phases of the attack.

11. #malicious #vbscript hides its code between a large number of comments (44 code lines among 73k comment lines). seems the code is part of a another script [incomplete]. [Sometimes having a good tool helps a lot. Definitely notepad++ is one of them.]

(a) vbscript (Obfuscated Files or Information (T1027)):

event.provider : "Microsoft-Windows-Sysmon/Operational"

```
AND ((winlog.event_id : "1" AND message : (*wscript.exe* OR
WSCRIPT.EXE* OR *cscript.exe* OR *CSCRIPT.EXE*)) OR
(winlog.event_id : ("11" OR "15") AND message : *.vbs*))
```

- (b) comments (Obfuscated Files or Information (T1027)):

```
event.provider : "Microsoft-Windows-Sysmon/Operational"
AND winlog.event_id : ("11" OR "15") AND message : (*REM *
OR ** OR *.vbs*)
```

- (c) notepad++ (Obfuscated Files or Information (T1027)):

```
event.provider : "Microsoft-Windows-Sysmon/Operational"
AND winlog.event_id : "1" AND message :
(*notepad++* OR *notepad++.exe*
OR *NOTEPAD++.EXE*)
```

- (d) Aggregate KQL Query:

```
event.provider : "Microsoft-Windows-Sysmon/Operational"
AND ((winlog.event_id : "1" AND message : (*wscript.exe* OR
WSCRIPT.EXE* OR *cscript.exe* OR *CSCRIPT.EXE* OR
notepad++* OR *notepad++.exe* OR *NOTEPAD++.EXE*)) OR
(winlog.event_id : ("11" OR "15") AND message : (*.vbs* OR *REM
* OR **)))
```

- (e) The queries focus on detecting heavily obfuscated VBScript activity through multiple angles. First, they monitor for script host execution (wscript.exe or cscript.exe) and VBS file operations. The comment-based obfuscation is tracked by looking for common VBScript comment indicators (REM and single quote). Additionally, the queries monitor for Notepad++ usage, which might indicate manual analysis or modification of the script. The aggregate query combines these elements to detect both the execution attempt and any potential manipulation of the obfuscated script file.

12. Yesterday I found an interesting js file with the name of wtf.js: After seeing the code I realized two things: 1. first the name was appropriate 2. I need to develop a tool to assist me in deobfuscating such code mb by partially interpreting the code

(a) wtf.js (Deobfuscate/Decode Files or Information (T1140)):

```
event.provider : "Microsoft-Windows-Sysmon/Operational"
AND ((winlog.event_id : ("11" OR "15") AND message : *wtf.js*)
OR (winlog.event_id : "1" AND message : (*wscript.exe* OR
WSCRIPT.EXE* OR *cscript.exe* OR *CSCRIPT.EXE* OR *node.exe*
OR *NODE.EXE*) AND message : *wtf.js*))
```

(b) Aggregate KQL Query:

```
event.provider : "Microsoft-Windows-Sysmon/Operational"
AND ((winlog.event_id : ("11" OR "15") AND message : *wtf.js*)
OR (winlog.event_id : "1" AND message : (*wscript.exe* OR
WSCRIPT.EXE* OR *cscript.exe* OR *CSCRIPT.EXE* OR *node.exe*
OR *NODE.EXE*) AND message : *wtf.js*))
```

(c) The queries track both file operations and execution attempts of the specific JavaScript file "wtf.js". They monitor for the file being accessed or modified, as well as any attempts to execute it through common JavaScript engines (Windows Script Host or Node.js). The same query serves as both individual and aggregate since we're focusing on a single, specific file with known execution methods.

13. #doc -> #vba macro deobfuscates a #document\_variable and then executes it -> #powershell dls #malware  
hxxp://shopthelighthouse.com/CHI/TTH.exe

(a) shopthelighthouse.com/CHI/TTH.exe (User Execution (T1204)):

```
event.provider : "Microsoft-Windows-Sysmon/Operational"
AND ((winlog.event_id : "3" AND message : (*shopthelighthouse* AND
TTH.exe)) OR (winlog.event_id : ("11" OR "15") AND message :
TTH.exe))
```

(b) Aggregate KQL Query:

```
event.provider : "Microsoft-Windows-Sysmon/Operational"
AND ((winlog.event_id : "3" AND message : (*shopthelighthouse* AND
TTH.exe)) OR (winlog.event_id : ("11" OR "15") AND message :
TTH.exe))
```

- (c) The queries focus on detecting both the network and file system activities related to the malicious executable. They monitor for network connections to the specific domain and path where the malware is hosted, while also watching for the creation or modification of the TTH.exe file on the local system. The same query serves as both individual and aggregate since we're tracking a specific malware download with known indicators.

14. another #amazing #rtf doc -> 1 #ole object -> dls and execs #malware:  
 #RTF #PE #NET  
 hxxp://meupload.site/1//f/7qONCSC Please investigate: @Malwageddon @decalage2 (next enhancement?)

(a) meupload.site/1//f/7qONCSC (Spearphishing Attachment (T1566.001)):

```
event.provider : "Microsoft-Windows-Sysmon/Operational"
AND ((winlog.event_id : ("11" OR "15") AND message : (*.rtf* OR
.ole)) OR (winlog.event_id : "3" AND message : *meupload.site*))
```

(b) meupload.site/1//f/7qONCSC (User Execution (T1204)):

```
event.provider : "Microsoft-Windows-Sysmon/Operational"
AND ((winlog.event_id : "1" AND message : (*winword.exe* OR *WIN-
```

```
WORD.EXE*)) OR (winlog.event_id : "3" AND
message : *meupload.site*))
```

(c) Aggregate KQL Query:

```
event.provider : "Microsoft-Windows-Sysmon/Operational"
AND ((winlog.event_id : ("11" OR "15") AND message : (*.rtf* OR
.ole)) OR (winlog.event_id : "1" AND message : (*winword.exe*
OR *WINWORD.EXE*)) OR (winlog.event_id : "3" AND message :
meupload.site))
```

(d) The queries monitor for multiple stages of this RTF-based attack chain. They track file operations related to RTF and OLE object files, Microsoft Word process execution for document handling, and network connections to the malicious domain. This multi-faceted approach allows detection of both the initial file-based attack vector and subsequent malware download attempts. The aggregate query combines these elements to provide visibility across the entire attack flow, from document opening through malware retrieval.

15. Interesting #bash script: extracts all hosts+usernames+(pwns/ssh keys) that the system has access or accessed cons and infects them with itself (#lateral\_movement). It also dls & execs a #cryptocoin #miner on the victims. #fileless #worm? @\_devonkerr\_ @ItsReallyNick @Dinosn

(a) hosts (Unsecured Credentials (T1552)):

```
event.provider : "Microsoft-Windows-Sysmon/Operational"
AND winlog.event_id : ("1" OR "11") AND message : (*hosts* OR
known_hosts OR *.ssh*)
```

(b) usernames (Unsecured Credentials (T1552)):

```
event.provider : "Microsoft-Windows-Sysmon/Operational"
```

AND winlog.event\_id : ("1" OR "11") AND message : (\*users\* OR \*username\* OR \*.ssh\* OR \*credentials\*)

- (c) pwds/ssh keys (Unsecured Credentials (T1552)):

event.provider : "Microsoft-Windows-Sysmon/Operational"

AND winlog.event\_id : ("1" OR "11") AND message : (\*id\_rsa\* OR \*id\_dsa\* OR \*known\_hosts\* OR \*.ssh\* OR \*.ppk\* OR \*putty\*)

- (d) cryptocoin miner (Resource Hijacking (T1496)):

event.provider : "Microsoft-Windows-Sysmon/Operational"

AND ( (winlog.event\_id : "1" AND message : (\*xmrig\* OR \*minerd\* OR \*minergate\* OR \*cpuminer\* OR \*cgminer\* OR \*bfgminer\* OR \*ethminer\*)) OR (winlog.event\_id : "3" AND message : (\*pool\* OR \*mining\* OR \*stratum\* OR \*xmr\* OR \*monero\* OR \*eth\* OR \*ethereum\*)) )

- (e) extracts all hosts (File and Directory Discovery (T1083)):

event.provider : "Microsoft-Windows-Sysmon/Operational"

AND winlog.event\_id : ("1" OR "11") AND message : (\*dir\* OR \*type\* OR \*findstr\* OR \*find\* OR \*hosts\* OR \*net\* OR \*ipconfig\*)

- (f) Aggregate KQL Query:

event.provider : "Microsoft-Windows-Sysmon/Operational"

AND ( (winlog.event\_id : ("1" OR "11") AND message : ( \*hosts\* OR \*known\_hosts\* OR \*.ssh\* OR \*users\* OR \*username\* OR \*credentials\* OR \*id\_rsa\* OR \*id\_dsa\* OR \*.ppk\* OR \*putty\* OR \*dir\* OR \*type\* OR \*findstr\* OR \*find\* OR \*net\* OR \*ipconfig\* ) ) OR (winlog.event\_id : "1" AND message : (\*xmrig\* OR \*minerd\* OR \*minergate\* OR \*cpuminer\* OR \*cgminer\* OR \*bfgminer\* OR \*ethminer\*)) OR (winlog.event\_id : "3" AND message : (\*pool\* OR \*mining\* OR \*stratum\* OR \*xmr\* OR \*monero\* OR \*eth\* OR \*ethereum\*)) )



(g) The queries target multiple aspects of this worm-like threat. They monitor for credential harvesting activities, including searches for SSH keys and stored credentials. System enumeration commands are tracked to detect host discovery attempts. Cryptocurrency mining activity is detected through both process execution and network connections to mining pools. The aggregate query combines these elements to provide comprehensive visibility across the entire attack chain, from initial reconnaissance through credential theft and crypto mining deployment.

16. #lnk -> #powershell -> dls #wsf file (#VBScript #Scriptlet) #RAT -> #Post data to server (unavailable) #wsf

hxxps://antwerptattoostudio.be/google-driver-attachment.pl/

katarzyna-dankow.wsf C2

hxxp://google-drive.myq-see.com:5399/ port 80 acts like #tarpit (keeps the con open)

(a) hxxps://antwerptattoostudio.be/google-driver-attachment.pl

/katarzyna-dankow.wsf (User Execution (T1204)):

event.provider : "Microsoft-Windows-Sysmon/Operational"

AND ( (winlog.event\_id : "3" AND message : (\*antwerptattoostudio\* OR \*katarzyna-dankow.wsf\*)) OR (winlog.event\_id : ("11" OR "15") AND message : \*katarzyna-dankow.wsf\*) )

(b) hxxps://antwerptattoostudio.be/google-driver-attachment.pl/

katarzyna-dankow.wsf (Data Obfuscation (T1001)):

event.provider : "Microsoft-Windows-Sysmon/Operational"

AND ( (winlog.event\_id : "3" AND message : (\*antwerptattoostudio\* OR \*katarzyna-dankow.wsf\*)) OR (winlog.event\_id : "1" AND message : (\*powershell\* OR \*wscript\* OR \*cscript\*)) )

- (c) `hxxp://google-drive.myq-see.com:5399` (Data Obfuscation (T1001)):

`event.provider : "Microsoft-Windows-Sysmon/Operational"`

`AND ( (winlog.event_id : "3" AND message : (*google-drive.myq-see* OR *5399*)) OR (winlog.event_id : "3" AND message : *80*) )`

- (d) Aggregate KQL Query:

`event.provider : "Microsoft-Windows-Sysmon/Operational"`

`AND ( (winlog.event_id : "3" AND message : (*antwerptattoostudio* OR *katarzyna-dankow.wsf* OR *google-drive.myq-see* OR *5399* OR *80* ) ) OR (winlog.event_id : ("11" OR "15") AND message : *katarzyna-dankow.wsf*) OR (winlog.event_id : "1" AND message : (*powershell* OR *POWERSHELL.EXE* OR *wscript* OR WSCRIPT.EXE* OR *cscript* OR *CSCRIPT.EXE*)) )`

- (e) The queries monitor multiple stages of this multi-component attack chain. They track network connections to both the initial WSF file download and subsequent C2 communications, including the specific ports used. Process execution monitoring covers PowerShell and script host processes that handle the WSF file. File operations are monitored for the WSF payload itself. The aggregate query combines these elements to detect activity across the entire attack flow, from initial script download through sustained C2 communication attempts.

17. The binary file is a `#NET` file. The actual `#malware` is `#encrypted` and `#stored` as a resource. The app `#decrypted` it and invoke it dynamically

- (a) `#encrypted` (Obfuscated Files or Information (T1027)):

`event.provider : "Microsoft-Windows-Sysmon/Operational"`

`AND ( (winlog.event_id : ("7" OR "8") AND message : (*System.Security.Cryptography* OR *mscorlib.dll* OR *mscorlib.dll* OR *clr.dll*)) OR`

```
(winlog.event_id : "1" AND message : (*InstallUtil* OR *RegAsm* OR
RegSvcs OR *.NET*)))
```

- (b) #decrypted (Obfuscated Files or Information (T1027)):

```
event.provider : "Microsoft-Windows-Sysmon/Operational"
AND ((winlog.event_id : ("7" OR "8") AND message :
(*System.Reflection* OR *Assembly.Load* OR *System.Resources*)) OR
(winlog.event_id : "10" AND message : *WriteProcessMemory*))
```

- (c) Aggregate KQL Query:

```
event.provider : "Microsoft-Windows-Sysmon/Operational"
AND ((winlog.event_id : ("7" OR "8") AND message : (*System.Security
.Cryptography* OR *System.Reflection* OR *Assembly.Load* OR *Sys-
tem.Resources* OR *mscorlib.dll* OR *mscorlib.dll* OR *clr.dll*)) OR
(winlog.event_id : "1" AND message : (*InstallUtil* OR *RegAsm* OR
RegSvcs OR *.NET*)) OR (winlog.event_id : "10" AND message :
WriteProcessMemory))
```

- (d) The queries focus on detecting .NET-based malware that uses encryption and dynamic loading techniques. They monitor for cryptographic operations through relevant .NET classes, as well as reflection and resource manipulation that might indicate dynamic code loading. Process memory modifications are also tracked to catch the decryption and execution phase. The aggregate query combines these elements to detect the full sequence of encryption, resource extraction, decryption, and dynamic execution typically seen in this type of .NET malware.

18. Sb is getting ready to launch his/her #malware dropper looking for a hosting (current one doesn't fly) #rtf -> 5 #ole object -> #xls document -> #vba exes #powershell stored in a cell #powershell dls 2 files 1 hosted on Seen b4:

- (a) *#malware dropper* (Ingress Tool Transfer (T1105)):
- ```
event.provider : "Microsoft-Windows-Sysmon/Operational"
AND ( (winlog.event_id : ("11" OR "15") AND message : (*.exe* OR
*.dll* OR *.msi*)) OR (winlog.event_id : "3" AND message : *down-
load*) )
```
- (b) *#rtf* (Ingress Tool Transfer (T1105)):
- ```
event.provider : "Microsoft-Windows-Sysmon/Operational"
AND winlog.event_id : ("11" OR "15") AND message : *.rtf*
```
- (c) *#ole object* (Ingress Tool Transfer (T1105)):
- ```
event.provider : "Microsoft-Windows-Sysmon/Operational"
AND winlog.event_id : ("11" OR "15") AND message : *.ole*
```
- (d) *#xls document* (Ingress Tool Transfer (T1105)):
- ```
event.provider : "Microsoft-Windows-Sysmon/Operational"
AND ((winlog.event_id : ("11" OR "15") AND message : (*.xls* OR
.xlsx)) OR (winlog.event_id : "1" AND message : (*excel.exe* OR
EXCEL.EXE)))
```
- (e) *#vba* (Ingress Tool Transfer (T1105)):
- ```
event.provider : "Microsoft-Windows-Sysmon/Operational"
AND ( (winlog.event_id : ("11" OR "15") AND message : *.vba*) OR
(winlog.event_id : "1" AND message : (*excel.exe* OR *EXCEL.EXE*
OR *winword.exe* OR *WINWORD.EXE*)) )
```
- (f) *#powershell* (Ingress Tool Transfer (T1105)):
- ```
event.provider : "Microsoft-Windows-Sysmon/Operational"
AND ((winlog.event_id : "1" AND message : (*powershell* OR *POW-
ERSHELL.EXE*)) OR (winlog.event_id : "3" AND message : *down-
load*))
```

(g) #powershell

(Command and Scripting Interpreter: PowerShell (T1059.001)):

```
(event.provider : ("Microsoft-Windows-PowerShell*" OR "Windows PowerShell") AND message : (*Invoke-Expression* OR *IEX* OR *FromBase64* OR *-enc* OR *hidden* OR *download*)) OR (event.provider : "Microsoft-Windows-Sysmon/Operational" AND winlog.event_id : "1" AND message : (*powershell* OR *POWERSHELL.EXE*))
```

(h) Aggregate KQL Query:

```
((event.provider : "Microsoft-Windows-Sysmon/Operational" AND ((winlog.event_id : ("11" OR "15") AND message : (*.exe* OR *.dll* OR *.msi* OR *.rtf* OR *.ole* OR *.xls* OR *.xlsx* OR *.vba*)) OR (winlog.event_id : "1" AND message : (*excel.exe* OR *EXCEL.EXE* OR *winword.exe* OR *WINWORD.EXE* OR *powershell* OR *POWERSHELL.EXE*)) OR (winlog.event_id : "3" AND message : *download*))) OR (event.provider : ("Microsoft-Windows-PowerShell*" OR "Windows PowerShell") AND message : (*Invoke-Expression* OR *IEX* OR *FromBase64* OR *-enc* OR *hidden* OR *download*)))
```

(i) The queries track multiple stages of this complex malware delivery chain. They monitor for file operations related to the various document formats (RTF, OLE, XLS) and their associated Office processes. PowerShell activity is monitored both through process execution and specific command patterns that might indicate malicious behavior. The aggregate query combines these elements to provide visibility across the entire attack chain, from initial document handling through PowerShell execution and file downloads. Special attention is paid to PowerShell's interaction with the Excel document and subsequent download attempts.

19. 0405.rar is a #coinminer

hxxp://74.222.14.61/0405.rar

(a) hxxp://74.222.14.61/0405.rar (Resource Hijacking (T1496)):

event.provider : "Microsoft-Windows-Sysmon/Operational"

AND ( (winlog.event\_id : "3" AND message :

(\*74.222.14.61\* OR \*0405.rar\*)) OR (winlog.event\_id : ("11" OR "15")

AND message : (\*0405.rar\* OR \*xmrig\* OR \*miner\* OR \*minergate\*

OR \*cpuminer\* OR \*cgminer\*)) OR (winlog.event\_id : "1" AND message

: (\*xmrig\* OR \*miner\* OR \*minergate\* OR \*cpuminer\* OR \*cgminer\*))

)

(b) Aggregate KQL Query:

event.provider : "Microsoft-Windows-Sysmon/Operational"

AND ( (winlog.event\_id : "3" AND message :

(\*74.222.14.61\* OR \*0405.rar\*)) OR (winlog.event\_id : ("11" OR "15")

AND message : (\*0405.rar\* OR \*xmrig\* OR \*miner\* OR \*minergate\*

OR \*cpuminer\* OR \*cgminer\*)) OR (winlog.event\_id : "1" AND message

: (\*xmrig\* OR \*miner\* OR \*minergate\* OR \*cpuminer\* OR \*cgminer\*))

)

(c) The queries monitor both the delivery and execution phases of this cryptocurrency mining malware. They track network connections to the malicious IP for the initial RAR download, file operations involving the archive, and subsequent execution of known cryptocurrency mining processes. The same query serves both individual and aggregate detection since we're monitoring a specific threat with well-defined indicators, from initial download through mining activity.

20. kill list:

hxxp://wmi.oo000oo.club:8888/kill.html mal list:

hxxp://wmi.oo000oo.club:8888/test.html remote dll:

hxxp://js.oo000oo.club:280/v.sct Note: the JScript seems to be a string ready to be exec by eval hence I unwrapped the code

(a) wmi.oo000oo.club:8888/kill.html

(Data from Information Repositories (T1213)):

event.provider : "Microsoft-Windows-Sysmon/Operational"

AND winlog.event\_id : "3" AND message : (\*wmi.oo000oo\* AND \*8888\* AND \*kill.html\*)

(b) wmi.oo000oo.club:8888/test.html

(Data from Information Repositories (T1213)):

event.provider : "Microsoft-Windows-Sysmon/Operational"

AND winlog.event\_id : "3" AND message : (\*wmi.oo000oo\* AND \*8888\* AND \*test.html\*)

(c) js.oo000oo.club:280/v.sct

(Data from Information Repositories (T1213)):

event.provider : "Microsoft-Windows-Sysmon/Operational"

AND ( (winlog.event\_id : "3" AND message : (\*js.oo000oo\* AND \*280\* AND \*v.sct\*)) OR (winlog.event\_id : ("11" OR "15") AND message : \*.sct\*) )

(d) Aggregate KQL Query:

event.provider : "Microsoft-Windows-Sysmon/Operational"

AND ( (winlog.event\_id : "3" AND message : ( (\*wmi.oo000oo\* AND (\*8888\* OR \*kill.html\* OR \*test.html\*)) OR (\*js.oo000oo\* AND \*280\* AND \*v.sct\*) ) ) OR (winlog.event\_id : ("11" OR "15") AND message : \*.sct\*) OR (winlog.event\_id : "1" AND message : (\*scrobj.dll\* OR

\*regsvr32\* OR \*REGSVR32.EXE\* OR \*wscript\* OR  
WSSCRIPT.EXE\*)) )

- (e) The queries monitor network connections to multiple suspicious endpoints serving different components of the attack. They track access to both the HTML-based lists and the SCT script file. Process execution is monitored for script hosts and COM registration that might handle the SCT file. The aggregate query combines these elements to detect activity across the full attack chain, including both the information gathering phase (kill and mal lists) and the execution phase (SCT loading). Special attention is paid to the eval-ready JScript by monitoring script host processes that could execute it.

21. Another active @Dropbox phishing kit with code exposed in #OpenDir Creds go to: \$sent ='dresultpage@gmail.com'; and another account hidden/encoded in the country sorter function.

- (a) dresultpage@gmail.com (Phishing (T1566)):

event.provider : "Microsoft-Windows-Sysmon/Operational"  
AND ( (winlog.event\_id : "3" AND message : (\*dresultpage\*  
OR \*gmail.com\*)) OR (winlog.event\_id : ("11" OR "15") AND message  
: (\*dresultpage\* OR \*gmail\*)) )

- (b) dresultpage@gmail.com (Exfiltration to Cloud Storage (T1567.002)):

event.provider : "Microsoft-Windows-Sysmon/Operational"  
AND ( (winlog.event\_id : "3" AND message : (  
(\*dropbox\* AND (\*dresultpage\*  
OR \*gmail.com\*)) OR (\*gmail.com\* AND \*POST\*)) OR  
(winlog.event\_id : "1" AND message : (\*curl\* OR \*wget\* OR \*certutil\*  
OR \*bitsadmin\*)) )



## (c) Aggregate KQL Query:

```
event.provider : "Microsoft-Windows-Sysmon/Operational"
AND ((winlog.event_id : "3" AND message : (*dresultpage* OR
gmail.com* OR (*dropbox* AND (*dresultpage* OR *gmail.com*)) OR
(*gmail.com* AND *POST*))) OR (winlog.event_id : ("11" OR "15")
AND message : (*dresultpage* OR *gmail*)) OR (winlog.event_id : "1"
AND message : (*curl* OR *wget* OR *certutil* OR *bitsadmin*)))
```

- (d) The queries focus on detecting both the phishing infrastructure and the exfiltration of stolen credentials. They monitor for network connections involving the Gmail address used for receiving credentials, particularly in conjunction with Dropbox services. POST requests to Gmail are tracked to catch credential exfiltration, while common download utilities are monitored to detect potential phishing kit deployment or updates. The aggregate query combines these elements to provide visibility into both the phishing operation setup and subsequent credential theft attempts.

22. RT @Techhelistcom: 86.105.53.140 hundreds of open dir domains. hundreds of phishing sites kits. a few pony panels.

## (a) 86.105.53.140 (Phishing (T1566)):

```
event.provider : "Microsoft-Windows-Sysmon/Operational"
AND ((winlog.event_id : "3" AND message : *86.105.53.140*) OR (win-
log.event_id : ("11" OR "15") AND message : (*.html* OR *.php* OR
.htm)))
```

## (b) 86.105.53.140 (Server Software Component (T1505)):

```
event.provider : "Microsoft-Windows-Sysmon/Operational"
AND ((winlog.event_id : "3" AND message : *86.105.53.140*) OR (win-
log.event_id : ("11" OR "15") AND message : (*panel* OR *pony* OR
```

```
.php)) OR (winlog.event_id : "1" AND message : (*w3wp.exe* OR
W3WP.EXE OR *httpd.exe* OR *HTTPD.EXE* OR *nginx.exe* OR
NGINX.EXE)))
```

(c) Aggregate KQL Query:

```
event.provider : "Microsoft-Windows-Sysmon/Operational"
AND ((winlog.event_id : "3" AND message : *86.105.53.140*) OR (win-
log.event_id : ("11" OR "15") AND message : (*.html* OR *.php* OR
.htm OR *panel* OR *pony*)) OR (winlog.event_id : "1" AND message
: (*chrome* OR *CHROME.EXE* OR *firefox* OR *FIREFOX.EXE*
OR *iexplore* OR *IEXPLORE.EXE* OR *msedge* OR
MSEDGE.EXE* OR *w3wp.exe* OR *W3WP.EXE* OR *httpd.exe* OR
HTTPD.EXE OR *nginx.exe* OR *NGINX.EXE*)))
```

(d) The queries target both client-side phishing activity and server-side control panel components. They monitor network connections to the known malicious IP address, while also watching for web server processes and file operations that might indicate phishing kit or control panel presence. Browser processes are monitored to detect potential phishing page visits. The aggregate query combines these elements to provide visibility into both phishing delivery and backend panel operations, particularly focusing on the Pony panel infrastructure.

23. Very low detection rate @Dropbox #Phishing bypasses Google Safe Browsing and most others.

hxxps://gnenco[.]global/images/stil/index.php Microsoft Live phishing same IP (111.90.147[.]134) also PayPal. ASN: Shinjiru – don't bother with abuse reports... or hide your identity

<https://t.co/hso5CTI5Gy>

- (a) gnenco.global/images/stil/index.php (Phishing (T1566)):

event.provider : "Microsoft-Windows-Sysmon/Operational"

AND ( (winlog.event\_id : "3" AND message : (\*gnenco.global\* OR \*gnenco\*)) OR (winlog.event\_id : ("11" OR "15") AND message : (\*.html\* OR \*.php\* OR \*.htm\*)) )

- (b) 111.90.147.134 (Phishing (T1566)):

event.provider : "Microsoft-Windows-Sysmon/Operational"

AND ( (winlog.event\_id : "3" AND message : \*111.90.147.134\*) OR (winlog.event\_id : ("11" OR "15") AND message : (\*.html\* OR \*.php\* OR \*.htm\*)) )

- (c) Aggregate KQL Query:

event.provider : "Microsoft-Windows-Sysmon/Operational"

AND ( (winlog.event\_id : "3" AND message : (\*gnenco.global\* OR \*gnenco\* OR \*111.90.147.134\*)) OR (winlog.event\_id : ("11" OR "15") AND message : (\*.html\* OR \*.php\* OR \*.htm\*)) OR (winlog.event\_id : "1" AND message : (\*chrome\* OR \*CHROME.EXE\* OR \*firefox\* OR \*FIREFOX.EXE\* OR \*iexplore\* OR \*IEXPLORE.EXE\* OR \*msedge\* OR \*MSEDGE.EXE\*)) )

- (d) The queries monitor for phishing activities targeting multiple services (Microsoft Live and PayPal) through both domain and IP-based indicators. They track network connections to both the domain name and the specific IP address known to host the phishing pages. Browser process execution is monitored to detect potential victim visits to these pages. The aggregate query combines these elements to provide comprehensive detection of phishing activity across both indicators, while also monitoring for browser-based interactions that might indicate successful phishing page delivery.

24. Active google docs phishing on compromised server at:

hxxps://emanuelandvalleriewedding.co[.]zw/snx/qoqdoc/ #Opendir and kit exposed at:

hxxps://emanuelandvalleriewedding.co[.]zw/snx/

hxxps://emanuelandvalleriewedding.co[.]zw/adm/ \$to =

"iamford918@gmail.com";

https://t.co/dR07yW1MNv

(a) emanuelandvalleriewedding.co.zw (Phishing (T1566)):

event.provider : "Microsoft-Windows-Sysmon/Operational"

AND winlog.event\_id : "3" AND message : \*emanuelandvalleriewedding\*

(b) hxxps://emanuelandvalleriewedding.co[.]zw/snx/qoqdoc/

(Phishing (T1566)):

event.provider : "Microsoft-Windows-Sysmon/Operational"

AND winlog.event\_id : "3" AND message : (\*emanuelandvalleriewedding\* AND \*snx\* AND \*qoqdoc\*)

(c) hxxps://emanuelandvalleriewedding.co[.]zw/snx/ (Phishing (T1566)):

event.provider : "Microsoft-Windows-Sysmon/Operational"

AND winlog.event\_id : "3" AND message : (\*emanuelandvalleriewedding\* AND \*snx\*)

(d) hxxps://emanuelandvalleriewedding.co[.]zw/adm/

(Phishing (T1566)):

event.provider : "Microsoft-Windows-Sysmon/Operational"

AND winlog.event\_id : "3" AND message : (\*emanuelandvalleriewedding\* AND \*adm\*)

(e) iamford918@gmail.com (Phishing (T1566)):

event.provider : "Microsoft-Windows-Sysmon/Operational"

AND ( (winlog.event\_id : "3" AND message : (\*iamford918\*  
OR \*gmail.com\*)) OR (winlog.event\_id : ("11" OR "15") AND message  
: (\*iamford918\* OR \*gmail\*)) )

- (f) emanuelandvalleriewedding.co.zw (Drive-by Compromise (T1189)):

event.provider : "Microsoft-Windows-Sysmon/Operational"

AND winlog.event\_id : "3" AND message : \*emanuelandvalleriewedding\*

- (g) hxxps://emanuelandvalleriewedding.co[.]zw/snx/qoqdoc/ (Drive-by Com-  
promise (T1189)):

event.provider : "Microsoft-Windows-Sysmon/Operational"

AND winlog.event\_id : "3" AND message : (\*emanuelandvalleriewed-  
ding\* AND \*snx\* AND \*qoqdoc\*)

- (h) Aggregate KQL Query:

event.provider : "Microsoft-Windows-Sysmon/Operational"

AND ( (winlog.event\_id : "3" AND message : ( \*emanuelandvalleriewed-  
ding\* OR (\*emanuelandvalleriewedding\* AND \*snx\* AND \*qoqdoc\*) OR  
(\*emanuelandvalleriewedding\* AND \*snx\*) OR (\*emanuelandvalleriewed-  
ding\* AND \*adm\*) OR \*iamford918\* OR \*gmail.com\* ) ) OR (win-  
log.event\_id : ("11" OR "15") AND message : (\*iamford918\* OR \*gmail\*))  
OR (winlog.event\_id : "1" AND message : (\*chrome\* OR  
CHROME.EXE\* OR \*firefox\* OR \*FIREFOX.EXE\* OR \*iexplore\* OR  
IEXPLORE.EXE\* OR \*msedge\* OR  
MSEDGE.EXE\*)) )

- (i) The queries monitor both phishing delivery and backend infrastructure components. Network connections are tracked to various paths on the compromised wedding website domain, including the phishing page, exposed directories, and admin panel. The Gmail address used for credential collection is monitored in both network traffic and file operations. Browser

processes are tracked to detect potential victim interactions with the phishing pages. The aggregate query combines these elements to provide visibility across the full attack chain, from initial phishing page delivery through potential credential exfiltration.

25. Active @googledocs phishing page hosted :

hxxp://tantal-service[.]pl/wp-admin/user/g-doc/ kit is exposed here:

hxxp://tantal-service[.]pl/wp-admin/user/g-doc.zip source code shows creds being delivered to: \$to = "rderothschild69@gmail.com";

https://t.co/o36cij5ZPo

(a) tantal-service.pl/wp-admin/user/g-doc/ (Phishing (T1566)):

event.provider : "Microsoft-Windows-Sysmon/Operational"

AND ( (winlog.event\_id : "3" AND message : (\*tantal-service\* AND \*wp-admin\* AND \*g-doc\*)) OR (winlog.event\_id : ("11" OR "15") AND message : (\*.html\* OR \*.php\* OR \*.htm\*)) )

(b) tantal-service.pl/wp-admin/user/g-doc.zip (Phishing (T1566)):

event.provider : "Microsoft-Windows-Sysmon/Operational"

AND ( (winlog.event\_id : "3" AND message : (\*tantal-service\* AND \*wp-admin\* AND \*g-doc.zip\*)) OR (winlog.event\_id : ("11" OR "15") AND message : \*.zip\*) )

(c) rderothschild69@gmail.com (Phishing (T1566)):

event.provider : "Microsoft-Windows-Sysmon/Operational"

AND ( (winlog.event\_id : "3" AND message : (\*rderothschild69\* OR \*gmail.com\*)) OR (winlog.event\_id : ("11" OR "15") AND message : (\*rderothschild69\* OR \*gmail\*)) )

(d) Aggregate KQL Query:

event.provider : "Microsoft-Windows-Sysmon/Operational"

```

AND ((winlog.event_id : "3" AND message : ((*tantal-service* AND
(*wp-admin* OR *g-doc* OR *g-doc.zip*)) OR *rderothschild69* OR
gmail.com)) OR (winlog.event_id : ("11" OR "15") AND message
: (*.html* OR *.php* OR *.htm* OR *.zip* OR *rderothschild69* OR
gmail)) OR (winlog.event_id : "1" AND message : (*chrome* OR
CHROME.EXE OR *firefox* OR *FIREFOX.EXE* OR *iexplore* OR
IEXPLORE.EXE OR *msedge* OR
MSEDGE.EXE*)))

```

- (e) The queries monitor both the phishing infrastructure and credential collection components. They track network connections to the WordPress-based phishing page and the exposed phishing kit ZIP file. The Gmail address used for credential collection is monitored in both network traffic and file operations. Browser processes are tracked to detect potential victim interactions with the phishing page. The aggregate query combines these elements to provide comprehensive visibility across the attack chain, from kit deployment through credential theft attempts.

26. Active @Dropbox #Phishing with #OpenDir and exposed kit here:

```

hxxp://safelinkonlineverify[.]com/perthchamber2/ $to
= "lee.ryan9713@gmail.com"; sneaky (?) little back door in the country sorter
function adds an obscured email address to the sender list.
(resultbox14@gmail.com)
https://t.co/uaV1v6crei

```

- (a) safelinkonlineverify[.]com/perthchamber2/ (Phishing (T1566)):
- ```

event.provider : "Microsoft-Windows-Sysmon/Operational"
AND ( (winlog.event_id : "3" AND message : (*safelinkonlineverify* AND
*perthchamber2*)) OR (winlog.event_id : ("11" OR "15") AND message

```

: (*.html* OR *.php* OR *.htm*)))

(b) lee.ryan9713@gmail.com (Phishing (T1566)):

event.provider : "Microsoft-Windows-Sysmon/Operational"

AND ((winlog.event_id : "3" AND message : (*lee.ryan9713*
OR *gmail.com*)) OR (winlog.event_id : ("11" OR "15") AND message
: (*lee.ryan9713* OR *gmail*)))

(c) resultbox14@gmail.com (Phishing (T1566)):

event.provider : "Microsoft-Windows-Sysmon/Operational"

AND ((winlog.event_id : "3" AND message : (*resultbox14*
OR *gmail.com*)) OR (winlog.event_id : ("11" OR "15") AND message
: (*resultbox14* OR *gmail*)))

(d) safelinkonlineverify[.]com/perthchamber2/ (Exfiltration Over Web Service (T1567)):

event.provider : "Microsoft-Windows-Sysmon/Operational"

AND ((winlog.event_id : "3" AND message : ((*safelinkonlineverify*
AND *perthchamber2* AND *POST*) OR (*dropbox* AND *safelinkon-
lineverify*))))

(e) lee.ryan9713@gmail.com (Exfiltration Over Web Service (T1567)):

event.provider : "Microsoft-Windows-Sysmon/Operational"

AND ((winlog.event_id : "3" AND message : ((*lee.ryan9713* AND
POST) OR (*gmail.com* AND *POST*))))

(f) resultbox14@gmail.com (Exfiltration Over Web Service (T1567)):

event.provider : "Microsoft-Windows-Sysmon/Operational"

AND ((winlog.event_id : "3" AND message : ((*resultbox14* AND
POST) OR (*gmail.com* AND *POST*))))

(g) Aggregate KQL Query:


```

event.provider : "Microsoft-Windows-Sysmon/Operational"
AND ( (winlog.event_id : "3" AND message : ( *safelinkonlineverify* OR
(*safelinkonlineverify* AND *perthchamber2*) OR (*safelinkonlineverify*
AND *perthchamber2* AND *POST*) OR (*dropbox* AND *safelinkonlineverify*)
OR *lee.ryan9713* OR *resultbox14* OR *gmail.com* OR
(*gmail.com* AND *POST*) ) ) OR (winlog.event_id : ("11" OR "15")
AND message : (*.html* OR *.php* OR *.htm* OR *lee.ryan9713* OR
*resultbox14* OR *gmail*)) OR (winlog.event_id : "1" AND message :
(*chrome* OR *CHROME.EXE* OR *firefox* OR *FIREFOX.EXE* OR
*iexplore* OR *IEXPLORE.EXE* OR *msedge* OR
MSEDGE.EXE*)) )

```

- (h) The queries monitor both the phishing delivery mechanism and multiple exfiltration channels. They track network connections to the phishing domain and Dropbox infrastructure, while monitoring for POST requests that might indicate credential theft. Special attention is paid to both the primary and hidden Gmail addresses that receive stolen credentials. Browser processes are monitored to detect potential victim interactions. The aggregate query combines these elements to provide visibility across the full attack chain, including the backup exfiltration channel through the hidden email address in the country sorter function.

27. Compromised server hosing @Dropbox phishing kit with open dir here:

```

hxxps://fj-construction[.]com/nuti/ $send = "gloryfirm ltd@gmail.com";
https://t.co/nreps6VgeQ

```

- (a) `hxxps://fj-construction[.]com/nuti/ (Phishing (T1566)):`

```

event.provider : "Microsoft-Windows-Sysmon/Operational"
AND ( (winlog.event_id : "3" AND message : (*fj-construction* AND

```

```
*nuti*)) OR (winlog.event_id : ("11" OR "15") AND message : (*.html*
OR *.php* OR *.htm*)) )
```

- (b) gloryfirltd@gmail.com (Phishing (T1566)):

```
event.provider : "Microsoft-Windows-Sysmon/Operational"
AND ( (winlog.event_id : "3" AND message : (*gloryfirltd*
OR *gmail.com*)) OR (winlog.event_id : ("11" OR "15") AND message
: (*gloryfirltd* OR *gmail*)) )
```

- (c) Aggregate KQL Query:

```
event.provider : "Microsoft-Windows-Sysmon/Operational"
AND ( (winlog.event_id : "3" AND message : ( (*fj-construction* AND
*nuti*) OR *gloryfirltd* OR *gmail.com* OR (*dropbox* AND *fj-
construction*)) ) OR (winlog.event_id : ("11" OR "15") AND message :
(*.html* OR *.php* OR *.htm* OR *gloryfirltd* OR *gmail*)) OR (win-
log.event_id : "1" AND message : (*chrome* OR *CHROME.EXE* OR
*firefox* OR *FIREFOX.EXE* OR *iexplore* OR *IEXPLORE.EXE*
OR *msedge* OR
MSEDGE.EXE*)) )
```

- (d) The queries target both the phishing infrastructure and credential collection mechanisms. They monitor network connections to the compromised construction website domain and track interactions with the exposed directory containing the phishing kit. The Gmail address used for credential collection is monitored in both network traffic and file operations. Browser processes are tracked to detect potential victim interactions with the phishing page. The aggregate query combines these elements to provide complete visibility of the attack chain, from initial phishing page delivery through credential theft attempts.

28. Malicious PDF (md5: 11c6baa421c01e3f376fa9eaa80793e3f0bc41c7) linking to @Microsoft One Drive phishing kit on a compromised server. #Open dir and exposed kit here:

hxxp://dewsdueshil[.]in[.]net/inc/ actor email: wirebox1oz@gmail.com URL extracted via:

<https://t.co/KoANOx8axm>

<https://t.co/SOcw57DE1q>

- (a) 11c6baa421c01e3f376fa9eaa80793e3f0bc41c7 (Phishing (T1566)):
- ```
event.provider : "Microsoft-Windows-Sysmon/Operational"
AND ((winlog.event_id : ("11" OR "15") AND message : *.pdf*) OR
(winlog.event_id : "1" AND message : (*AcroRd32.exe* OR
ACRORD32.EXE* OR *Acrobat.exe* OR
ACROBAT.EXE*)))
```
- (b) hxxp://dewsdueshil[.]in[.]net/inc/ (Phishing (T1566)):
- ```
event.provider : "Microsoft-Windows-Sysmon/Operational"
AND ( (winlog.event_id : "3" AND message : (*dewsdueshil* AND
*inc*)) OR (winlog.event_id : ("11" OR "15") AND message : (*.html*
OR *.php* OR *.htm*)) )
```
- (c) wirebox1oz@gmail.com (Phishing (T1566)):
- ```
event.provider : "Microsoft-Windows-Sysmon/Operational"
AND ((winlog.event_id : "3" AND message : (*wirebox1oz*
OR *gmail.com*)) OR (winlog.event_id : ("11" OR "15") AND message
: (*wirebox1oz* OR *gmail*)))
```
- (d) Aggregate KQL Query:
- ```
event.provider : "Microsoft-Windows-Sysmon/Operational"
AND ( (winlog.event_id : "3" AND message : (*dewsdueshil* OR (*dews-
```

```

dueshil* AND *inc*) OR *wirebox1oz* OR *gmail.com* OR (*onedrive*
AND *dewsdueshil*) ) ) OR (winlog.event_id : ("11" OR "15") AND
message : (*.pdf* OR *.html* OR *.php* OR *.htm* OR *wirebox1oz*
OR *gmail*)) OR (winlog.event_id : "1" AND message : ( *chrome* OR
*CHROME.EXE* OR *firefox* OR *FIREFOX.EXE* OR *iexplore* OR
*IEXPLORE.EXE* OR *msedge* OR
MSEDGE.EXE* OR *AcroRd32.exe* OR *ACRORD32.EXE* OR *Acro-
bat.exe* OR *ACROBAT.EXE* ) ) )

```

- (e) The queries monitor multiple stages of this PDF-based phishing attack. They track PDF file operations and Adobe Reader process execution for the initial malicious document, network connections to both the compromised server and OneDrive infrastructure, and the Gmail address used for credential collection. Browser and PDF reader processes are monitored to detect victim interactions with both the malicious PDF and subsequent phishing pages. The aggregate query combines these elements to provide visibility across the full attack chain, from initial PDF delivery through potential credential theft.

29. RT @BleepinComputer: Drupalgeddon 2 Vulnerability Used to Infect Servers With Backdoors and Coinminers - by @campuscodi

<https://t.co/etMNw62d>

- (a) Drupalgeddon 2 (Exploit Public-Facing Application (T1190)):
- ```

event.provider : "Microsoft-Windows-Sysmon/Operational"
AND ((winlog.event_id : "3" AND message : (*drupal* OR
/user/register* OR */admin* OR */?q=*)) OR (winlog.event_id : ("11"
OR "15")
AND message : (*.php* OR *index.php*)))

```

- (b) Backdoors (Exploit Public-Facing Application (T1190)):

```
event.provider : "Microsoft-Windows-Sysmon/Operational"
AND ((winlog.event_id : "1" AND message : (*cmd.exe* OR *CMD.EXE*
OR *powershell* OR *POWERSHELL.EXE* OR *nc.exe* OR *ncat* OR
netcat)) OR (winlog.event_id : "3" AND message : (*reverse* OR
bind OR *4444* OR *5555* OR *443*)))
```

- (c) Coinminers (Exploit Public-Facing Application (T1190)):

```
event.provider : "Microsoft-Windows-Sysmon/Operational"
AND ((winlog.event_id : "1" AND message : (*xmrig* OR *minerd*
OR *minergate* OR *cpuminer* OR *cgminer* OR *bfgminer* OR *eth-
miner*)) OR (winlog.event_id : "3" AND message : (*pool* OR *mining*
OR *stratum* OR *xmr* OR *monero* OR *eth* OR *ethereum*)))
```

- (d) Aggregate KQL Query:

```
event.provider : "Microsoft-Windows-Sysmon/Operational"
AND ((winlog.event_id : "3" AND message : ((*drupal* OR
/user/register* OR */admin* OR */?q=*) OR (*reverse* OR *bind* OR
4444 OR *5555* OR *443*) OR (*pool* OR *mining* OR *stratum* OR
xmr OR *monero* OR *eth* OR *ethereum*)))) OR (winlog.event_id
: ("11" OR "15") AND message : (*.php* OR *index.php*)) OR (win-
log.event_id : "1" AND message : (*cmd.exe* OR *CMD.EXE* OR
powershell OR *POWERSHELL.EXE* OR *nc.exe* OR *ncat* OR
netcat OR *xmrig* OR *minerd* OR *minergate* OR *cpuminer* OR
cgminer OR *bfgminer* OR *ethminer*)))
```

- (e) The queries target three main aspects of this attack chain: the initial Drupal vulnerability exploitation, backdoor deployment, and cryptocurrency mining activity. They monitor for suspicious access to Drupal-specific paths and files that might indicate exploitation attempts. Backdoor de-

tection focuses on common command shells and network connections indicating reverse or bind shells. Cryptocurrency mining is detected through both process execution and network connections to mining pools. The aggregate query combines these elements to provide comprehensive visibility across the full attack chain, from initial compromise through both types of malicious payloads.

30. 2018-04-19 - #Hancitor (aka #Chanitor or #Tordal) #malspam - followup malware #ZeusPandaBanker (again) I got Send Safe Enterprise (SSE) #spambot #malware as well - 54 email examples #pcap of infection traffic and associated malware samples available at:

<https://t.co/XiUzo50zI8>

<https://t.co/oXIx0FXH2K>

- (a) <https://t.co/XiUzo50zI8> (Spearphishing Attachment (T1566.001)):

event.provider : "Microsoft-Windows-Sysmon/Operational"

AND ( (winlog.event\_id : "3" AND message : \*XiUzo50zI8\*) OR (winlog.event\_id : ("11" OR "15") AND message : (\*.doc\* OR \*.xls\* OR \*.pdf\* OR \*.zip\* OR \*.htm\*)) )

- (b) <https://t.co/oXIx0FXH2K> (Spearphishing Attachment (T1566.001)):

event.provider : "Microsoft-Windows-Sysmon/Operational"

AND ( (winlog.event\_id : "3" AND message : \*oXIx0FXH2K\*) OR (winlog.event\_id : ("11" OR "15") AND message : (\*.doc\* OR \*.xls\* OR \*.pdf\* OR \*.zip\* OR \*.htm\*)) )

- (c) Aggregate KQL Query:

event.provider : "Microsoft-Windows-Sysmon/Operational"

AND ( (winlog.event\_id : "3" AND message : (\*XiUzo50zI8\* OR oXIx0FXH2K\*)) OR (winlog.event\_id : ("11" OR "15") AND message :

```
(*doc* OR *.xls* OR *.pdf* OR *.zip* OR *.htm*)) OR (winlog.event_id : "1" AND message : (*svchost* OR *SVCHOST.EXE*)))
```

- (d) The queries focus on detecting access to the URLs containing malware samples and associated traffic captures. They monitor both network connections to these URLs and potential file downloads of common malicious attachment types. The aggregate query combines these elements while also monitoring for suspicious process execution that might indicate successful infection. This provides visibility into both the research/analysis phase (accessing the samples) and potential execution of similar malware in the environment.

31. RT @dvk01uk: Fake â Your Sage subscription invoice is Dueâ delivers smoke loader which downloads Trickbot

<https://t.co/z9LrmgqLu0>

- (a) Fake â Your Sage subscription invoice is Due (Spearphishing Attachment (T1566.001)):

```
event.provider : "Microsoft-Windows-Sysmon/Operational"
```

```
AND ((winlog.event_id : ("11" OR "15") AND message : (*sage* OR *invoice* OR *subscription*)) OR (winlog.event_id : "1" AND message : (*outlook.exe* OR *OUTLOOK.EXE* OR *thunderbird.exe* OR *THUNDERBIRD.EXE*)))
```

- (b) smoke loader (Spearphishing Attachment (T1566.001)):

```
event.provider : "Microsoft-Windows-Sysmon/Operational"
```

```
AND ((winlog.event_id : "1" AND message : (*rundll32* OR RUNDLL32.EXE* OR *regsvr32* OR *REGSVR32.EXE*)) OR (winlog.event_id : ("7" OR "8") AND message : (*user32.dll* OR *kernel32.dll* OR *advapi32.dll*)))
```

- (c) Trickbot (Spearphishing Attachment (T1566.001)):

```
event.provider : "Microsoft-Windows-Sysmon/Operational"
AND ((winlog.event_id : "1" AND message : (*svchost* OR
SVCHOST.EXE* OR *explorer.exe* OR *EXPLORER.EXE*)) OR (win-
log.event_id : "3" AND message : (*port 447* OR *port 449* OR *gtag*
OR *group tag*)) OR (winlog.event_id : ("11" OR "15") AND message
: (*%APPDATA%* OR *%TEMP%* OR *.tmp*)))
```

- (d) Aggregate KQL Query:

```
event.provider : "Microsoft-Windows-Sysmon/Operational"
AND ((winlog.event_id : ("11" OR "15") AND message : (*sage* OR
invoice OR *subscription* OR *%APPDATA%* OR *%TEMP%* OR
.tmp)) OR (winlog.event_id : "1" AND message : (*outlook.exe* OR
OUTLOOK.EXE OR *thunderbird.exe* OR *THUNDERBIRD.EXE*
OR *rundll32* OR
RUNDLL32.EXE* OR *regsvr32* OR *REGSVR32.EXE* OR *svchost*
OR *SVCHOST.EXE* OR *explorer.exe* OR *EXPLORER.EXE*)) OR
(winlog.event_id : ("7" OR "8") AND message : (*user32.dll* OR *ker-
nel32.dll* OR *advapi32.dll*)) OR (winlog.event_id : "3" AND message
: (*port 447* OR *port 449* OR *gtag* OR *group tag*)))
```

- (e) The queries monitor multiple stages of this malware infection chain. Initial detection focuses on the phishing email about Sage subscriptions. The smoke loader stage is detected through DLL loading patterns and common injection techniques. Trickbot's presence is identified through its characteristic process creation, network connections to specific ports, and file operations in user directories. The aggregate query combines these elements to provide visibility across the entire attack chain, from initial phishing through both stages of malware execution.



32. CFCCChamber an abandoned website cavecreekbicyclefestival[dot]com is compromised and hosting fake login pages used by phishing emails. The Town of Cave Creek AZ is listed as the POC for it. Technical details in my previous tweet at:

<https://t.co/cc8hEz9RZa>

- (a) cavecreekbicyclefestival.com (Drive-by Compromise (T1189)):

```
event.provider : "Microsoft-Windows-Sysmon/Operational"
AND ((winlog.event_id : "3" AND message : (*cavecreekbicyclefestival*
OR *cave* AND *creek* AND *bicycle*)) OR (winlog.event_id : ("11"
OR "15") AND message : (*.html* OR *.php* OR *.htm* OR *.js*)))
```

- (b) Aggregate KQL Query:

```
event.provider : "Microsoft-Windows-Sysmon/Operational"
AND ((winlog.event_id : "3" AND message : (*cavecreekbicyclefestival*
OR *cave* AND *creek* AND *bicycle*)) OR (winlog.event_id : ("11"
OR "15") AND message : (*.html* OR *.php* OR *.htm* OR *.js*)) OR
(winlog.event_id : "1" AND message : (*chrome* OR *CHROME.EXE*
OR *firefox* OR *FIREFOX.EXE* OR *iexplore* OR *IEXPLORE.EXE*
OR *msedge* OR
MSEDGE.EXE*)))
```

- (c) The queries focus on detecting interactions with the compromised bicycle festival website. They monitor network connections to the domain as well as common web file operations that might indicate phishing page delivery. Browser process execution is tracked to detect potential victim visits. The same basic detection approach serves both individual and aggregate queries, since we're primarily concerned with identifying attempts to access or deliver phishing content from this specific compromised domain.

### 33. 2018-04-18 - #malspam campaign pushing #Hancitor

(aka: #Chanitor or #Tordal) - Today's theme was an IRS notification - Mostly the same with #Pony #EvilPony (in memory) and #ZeusPandaBanker (on disk) saw the Panda do DNS but no full TCP connections -

<https://t.co/yFSfNFyibO>

<https://t.co/HeaQUiENWL>

#### (a) Hancitor (Spearphishing Attachment (T1566.001)):

event.provider : "Microsoft-Windows-Sysmon/Operational"

AND ( (winlog.event\_id : "3" AND message : (\*gate.php\* OR \*index.php\* OR \*hancitor\*)) OR (winlog.event\_id : "1" AND message : (\*svchost\* OR \*SVCHOST.EXE\*)) )

#### (b) Chanitor (Spearphishing Attachment (T1566.001)):

event.provider : "Microsoft-Windows-Sysmon/Operational"

AND ( (winlog.event\_id : "3" AND message : (\*gate.php\* OR \*index.php\* OR \*chanitor\*)) OR (winlog.event\_id : "1" AND message : (\*svchost\* OR \*SVCHOST.EXE\*)) )

#### (c) Tordal (Spearphishing Attachment (T1566.001)):

event.provider : "Microsoft-Windows-Sysmon/Operational"

AND ( (winlog.event\_id : "3" AND message : (\*gate.php\* OR \*index.php\* OR \*tordal\*)) OR (winlog.event\_id : "1" AND message : (\*svchost\* OR \*SVCHOST.EXE\*)) )

#### (d) IRS notification (Spearphishing Attachment (T1566.001)):

event.provider : "Microsoft-Windows-Sysmon/Operational"

AND ( (winlog.event\_id : ("11" OR "15") AND message : (\*irs\* OR \*tax\* OR \*notification\*)) OR (winlog.event\_id : "1" AND message : (\*outlook.exe\* OR \*OUTLOOK.EXE\* OR \*thunderbird.exe\* OR \*THUNDER-

BIRD.EXE\*)) )

- (e) Pony (Spearphishing Attachment (T1566.001)):

event.provider : "Microsoft-Windows-Sysmon/Operational"  
 AND ( (winlog.event\_id : "1" AND message : (\*rundll32\* OR  
 RUNDLL32.EXE\*)) OR (winlog.event\_id : ("7" OR "8") AND message  
 : (\*pony\* OR \*panel\*)) )

- (f) EvilPony (Spearphishing Attachment (T1566.001)):

event.provider : "Microsoft-Windows-Sysmon/Operational"  
 AND ( (winlog.event\_id : "1" AND message : (\*rundll32\* OR  
 RUNDLL32.EXE\*)) OR (winlog.event\_id : ("7" OR "8") AND message  
 : (\*evilpony\* OR \*panel\*)) )

- (g) ZeusPandaBanker (Spearphishing Attachment (T1566.001)):

event.provider : "Microsoft-Windows-Sysmon/Operational"  
 AND ( (winlog.event\_id : "3" AND message : (\*panda\* OR \*zeus\* OR  
 \*gate.php\*)) OR (winlog.event\_id : ("11" OR "15") AND message : \*)

- (h) Aggregate KQL Query:

event.provider : "Microsoft-Windows-Sysmon/Operational"  
 AND ( (winlog.event\_id : "3" AND message : ( \*gate.php\* OR \*in-  
 dex.php\* OR \*hancitor\* OR \*chanitor\* OR \*tordal\* OR \*panda\* OR  
 \*zeus\* OR \*panel\* ) ) OR (winlog.event\_id : ("11" OR "15") AND  
 message : (\*irs\* OR \*tax\* OR \*notification\* OR \*%APPDATA%\*)) OR  
 (winlog.event\_id : "1" AND message : ( \*svchost\* OR \*SVCHOST.EXE\*  
 OR \*outlook.exe\* OR \*OUTLOOK.EXE\* OR \*thunderbird.exe\* OR  
 THUNDERBIRD.EXE\* OR \*rundll32\* OR  
 RUNDLL32.EXE\* ) ) OR (winlog.event\_id : ("7" OR "8") AND message  
 : (\*pony\* OR \*evilpony\* OR \*panel\*)) )

- (i) The queries target multiple stages of this sophisticated malware campaign. Initial phishing detection focuses on IRS-themed email content and common email client processes. The various malware components (Hancitor/Chanitor/Tordal, Pony/EvilPony, and ZeusPandaBanker) are detected through their characteristic process creation, DLL loading, and file operations patterns. DNS activity is specifically monitored for the banking malware component. The aggregate query combines these elements to provide visibility across the entire attack chain, from initial phishing through the deployment of multiple malware payloads.

34. RT @thlnk3r: Compromised websites with injected js exploiting CVE-2018-4878. Final payload is XMR Miner. Powershell script:

<https://t.co/pâ>

- (a) CVE-2018-4878 (Drive-by Compromise (T1189)):

```
event.provider : "Microsoft-Windows-Sysmon/Operational"
AND ((winlog.event_id : "1" AND message : (*flashplayer* OR *flash*
OR *FLASHPLAYER.EXE* OR *FlashUtil* OR *FLASH.OCX*)) OR
(winlog.event_id : ("7" OR "8") AND message : (*flash*
OR *FLASH.OCX* OR *FlashUtil*)) OR (winlog.event_id : "3" AND
message : *.swf*))
```

- (b) Powershell script

```
(Command and Scripting Interpreter: PowerShell (T1059.001)):
(event.provider : ("Microsoft-Windows-PowerShell*" OR "Windows Pow-
erShell") AND message : (*IEX* OR *Invoke-Expression* OR *From-
Base64* OR *-enc* OR *hidden* OR *downloadstring* OR *miner* OR
xmr)) OR (event.provider : "Microsoft-Windows-Sysmon/Operational"
AND winlog.event_id : "1" AND message : (*powershell* OR *POWER-
```

SHELL.EXE\*))

(c) Aggregate KQL Query:

```
((event.provider : "Microsoft-Windows-Sysmon/Operational"
AND ((winlog.event_id : "1" AND message : (*flashplayer* OR *flash*
OR *FLASHPLAYER.EXE* OR *FlashUtil* OR *FLASH.OCX*
OR *powershell* OR *POWERSHELL.EXE*)) OR (winlog.event_id :
("7" OR "8") AND
message : (*flash* OR *FLASH.OCX* OR *FlashUtil*)) OR
(winlog.event_id : "3" AND message : (*.swf* OR *pool* OR *mining*
OR *xmr* OR *monero*)))) OR (event.provider : ("Microsoft-Windows-
PowerShell*" OR "Windows PowerShell") AND message : (*IEX* OR
Invoke-Expression OR *FromBase64* OR *-enc* OR *hidden* OR *down-
loadstring* OR *miner* OR *xmr*)))
```

(d) The queries monitor this attack chain from initial Flash exploitation through cryptocurrency mining activity. They track Flash-related processes and DLL loading that might indicate CVE-2018-4878 exploitation, along with PowerShell execution patterns common in malicious scripts. Special attention is paid to PowerShell commands related to cryptocurrency mining and network connections to mining pools. The aggregate query combines these elements to provide visibility across the full attack flow, from initial compromise through mining payload execution.

35. 2018-04-16 - A #phishing email with a link to a #malware EXE -and- a link to a fake Alibaba login page? Genius! - Sanitized email posted to Pastebin at:  
<https://t.co/4tD4IO7XKf> - Address.exe submitted to VT:  
<https://t.co/Wy98NhU9S5>  
<https://t.co/Ch4n7kmNAt>

- (a) Address.exe (User Execution (T1204)):

```
event.provider : "Microsoft-Windows-Sysmon/Operational"
AND ((winlog.event_id : "1" AND message : *Address.exe*) OR (win-
log.event_id : ("11" OR "15") AND message : *Address.exe*) OR (win-
log.event_id : "3" AND message : (*alibaba* OR *Address.exe*)))
```

- (b) Aggregate KQL Query:

```
event.provider : "Microsoft-Windows-Sysmon/Operational"
AND ((winlog.event_id : "1" AND message : *Address.exe*) OR (win-
log.event_id : ("11" OR "15") AND message : *Address.exe*) OR (win-
log.event_id : "3" AND message : (*alibaba* OR *Address.exe*)))
```

- (c) The queries focus on detecting both the malicious executable and potential interaction with the phishing page. They monitor for execution and file operations involving the specific "Address.exe" malware, while also tracking network connections that might indicate access to the fake Alibaba login page. The same query serves as both individual and aggregate detection since we're monitoring a specific set of indicators with known malware naming and targeting patterns.

36. RT @dvk01uk: More Gootkit banking trojan via fake Invoice Overdue Notice spoofing Metro Finance using Mailgun SMTP relay service

<https://t.â>

- (a) Gootkit (Spearphishing Attachment (T1566.001)):

```
event.provider : "Microsoft-Windows-Sysmon/Operational"
AND ((winlog.event_id : "1" AND message : (*rundll32* OR
RUNDLL32.EXE* OR *regsvr32* OR *REGSVR32.EXE*)) OR (win-
log.event_id : "3" AND message : (*gootkit* OR *.dat* OR *.bin*))
)
```

- (b) Invoice Overdue Notice (Spearphishing Attachment (T1566.001)):
- ```
event.provider : "Microsoft-Windows-Sysmon/Operational"
AND ( (winlog.event_id : ("11" OR "15") AND message : (*invoice*
OR *overdue* OR *notice*)) OR (winlog.event_id : "1" AND message :
(*outlook.exe* OR *OUTLOOK.EXE* OR *thunderbird.exe* OR *THUN-
DERBIRD.EXE*)) )
```
- (c) Metro Finance (Spearphishing Attachment (T1566.001)):
- ```
event.provider : "Microsoft-Windows-Sysmon/Operational"
AND ((winlog.event_id : ("11" OR "15") AND message : (*metro* OR
finance)) OR (winlog.event_id : "1" AND message : (*outlook.exe* OR
OUTLOOK.EXE OR *thunderbird.exe* OR *THUNDERBIRD.EXE*))
)
```
- (d) Mailgun SMTP relay service (Spearphishing Attachment (T1566.001)):
- ```
event.provider : "Microsoft-Windows-Sysmon/Operational"
AND ( (winlog.event_id : "3" AND message : (*mailgun* OR *smtp*))
OR (winlog.event_id : "1" AND message : (*outlook.exe* OR *OUT-
LOOK.EXE* OR *thunderbird.exe* OR *THUNDERBIRD.EXE*)) )
```
- (e) Aggregate KQL Query:
- ```
event.provider : "Microsoft-Windows-Sysmon/Operational"
AND ((winlog.event_id : "1" AND message : (*rundll32* OR
RUNDLL32.EXE* OR *regsvr32* OR *REGSVR32.EXE* OR
outlook.exe* OR *OUTLOOK.EXE* OR *thunderbird.exe* OR
THUNDERBIRD.EXE*)) OR (winlog.event_id : ("11" OR "15") AND
message : (*invoice* OR *overdue* OR *notice* OR *metro* OR *fi-
nance*)) OR (winlog.event_id : "3" AND message : (*gootkit* OR
.dat OR *.bin* OR *mailgun* OR *smtp*)))
```
- (f) The queries monitor multiple aspects of this banking trojan campaign.

They track both the email delivery infrastructure (Mailgun SMTP) and the phishing lure content (Metro Finance invoice notices). Gootkit-specific detections focus on its characteristic process creation and network patterns. The aggregate query combines these elements to provide visibility across the full attack chain, from initial phishing email delivery through banking trojan execution.

37. RT @securityaffairs: Exclusive - #APT group exploited still unpatched #0day in #IE dubbed 'double play' #securityaffairs #hacking  
[https://â](https://twitter.com/securityaffairs/status/1119011111111111111)

(a) #0day (Exploit Public-Facing Application (T1190)):

```
event.provider : "Microsoft-Windows-Sysmon/Operational"
AND ((winlog.event_id : ("7" OR "8") AND message : (*urlmon.dll* OR
wininet.dll OR *mshtml.dll*)) OR (winlog.event_id : ("11" OR "15")
AND message : (*.html* OR *.js* OR *.htm*)))
```

(b) #IE (Exploit Public-Facing Application (T1190)):

```
event.provider : "Microsoft-Windows-Sysmon/Operational"
AND ((winlog.event_id : "1" AND message : (*iexplore.exe* OR *IEXPLORE.EXE*)) OR (winlog.event_id : ("7" OR "8") AND message :
(*mshtml.dll* OR *jscript.dll* OR *vbscript.dll*)) OR (winlog.event_id :
"3" AND message : (*iexplore.exe* OR *IEXPLORE.EXE*)))
```

(c) Aggregate KQL Query:

```
event.provider : "Microsoft-Windows-Sysmon/Operational"
AND ((winlog.event_id : "1" AND message : (*iexplore.exe* OR *IEXPLORE.EXE*)) OR (winlog.event_id : ("7" OR "8") AND message :
(*urlmon.dll* OR *wininet.dll* OR *mshtml.dll* OR *mshtml.dll* OR
jscript.dll OR *vbscript.dll*)) OR (winlog.event_id : ("11" OR "15")
```



AND message : (\*.html\* OR \*.js\* OR \*.htm\*)) OR (winlog.event\_id : "3" AND message : (\*iexplore.exe\* OR \*IEXPLORE.EXE\*)) )

- (d) The queries target exploitation of an unpatched Internet Explorer vulnerability. They monitor IE process execution and the loading of key browser-related DLLs that might be leveraged in exploitation. File operations related to potentially malicious web content are tracked. The aggregate query combines these elements to provide visibility into potential exploitation attempts through IE's various components and subsystems.

38. RT @360CoreSec: We uncovered an IE 0day vulnerability has been embedded in malicious MS Office document targeting limited users by a knownâ

- (a) IE 0day (T1203 - Exploitation for Client Execution):

event.provider : "Microsoft-Windows-Sysmon/Operational"

AND ( (winlog.event\_id : "1" AND message : (\*iexplore.exe\* OR \*IEXPLORE.EXE\*)) OR (winlog.event\_id : ("7" OR "8") AND message : (\*mshtml.dll\* OR \*jscript.dll\* OR \*vbscript.dll\* OR \*urlmon.dll\*)) )

- (b) malicious MS Office document (T1203 - Exploitation for Client Execution):

event.provider : "Microsoft-Windows-Sysmon/Operational"

AND ( (winlog.event\_id : "1" AND message : (\*winword.exe\* OR \*WINWORD.EXE\* OR \*excel.exe\* OR \*EXCEL.EXE\* OR \*powerpnt.exe\* OR \*POWERPNT.EXE\*)) OR (winlog.event\_id : ("11" OR "15") AND message : (\*.doc\* OR \*.docx\* OR \*.xls\* OR \*.xlsx\* OR \*.ppt\* OR \*.pptx\*)) OR (winlog.event\_id : ("7" OR "8") AND message : (\*vbe7.dll\* OR \*vbe7intl.dll\* OR \*vbscript.dll\*)) )

- (c) Aggregate KQL Query:

event.provider : "Microsoft-Windows-Sysmon/Operational"

AND ( (winlog.event\_id : "1" AND message : (\*iexplore.exe\* OR \*IEXPLORE.EXE\*)) )

```

PLORE.EXE* OR *winword.exe* OR *WINWORD.EXE* OR *excel.exe*
OR *EXCEL.EXE* OR *powerpnt.exe* OR *POWERPNT.EXE*)) OR
(winlog.event_id : ("7" OR "8") AND message : (*mshtml.dll* OR
jscript.dll OR *vbscript.dll* OR *urlmon.dll* OR
vbe7.dll* OR *vbe7intl.dll*)) OR (winlog.event_id : ("11" OR "15")
AND message : (*.doc* OR *.docx* OR *.xls* OR *.xlsx* OR *.ppt* OR
.pptx)))

```

- (d) The queries monitor both Office document handling and Internet Explorer components that might be leveraged in this exploit chain. They track Office process execution and file operations, along with the loading of key DLLs used by both Office and IE that could be involved in exploitation. The aggregate query combines these elements to detect the full attack chain, from initial document opening through potential IE vulnerability exploitation.

### 39. RT @OsandaMalith: Stealing #NetNTLM #hashes with

C:\Windows\System32\regini.exe Randomly found this while vacationing in somewhere :D httpâ

- (a) C:\Windows\System32\regini.exe (Credential Dumping (T1003)):
- ```

event.provider : "Microsoft-Windows-Sysmon/Operational"
AND ( (winlog.event_id : "1" AND message : (*regini.exe* OR
REGINI.EXE*)) OR (winlog.event_id : "3" AND message : (*IPC$*
OR *ADMIN$* OR *C$*)) OR (winlog.event_id : "7" AND message :
(*samlib.dll* OR *security.dll*)) )

```

- (b) Aggregate KQL Query:

```

event.provider : "Microsoft-Windows-Sysmon/Operational"

```

```
AND ( (winlog.event_id : "1" AND message : (*regini.exe* OR
REGINI.EXE*)) OR (winlog.event_id : "3" AND message : (*IPC$*
OR *ADMIN$* OR *C$*)) OR (winlog.event_id : "7" AND message :
(*samlib.dll* OR *security.dll*)) )
```

- (c) The queries monitor for potential abuse of regini.exe for credential access. They track execution of the regini.exe binary, network connections to common Windows administrative shares that might be used in NTLM hash capture, and loading of security-related DLLs. The same detection approach serves both individual and aggregate needs since we're focusing on a specific binary and its common abuse patterns for credential theft.

40. RT @malware_traffic: 2018-04-19 - abandoned website cavecreekbicyclefestival[.]com compromised and hosting fake UBS banking login page - Usa

- (a) cavecreekbicyclefestival[.]com (Drive-by Compromise (T1189)):
- ```
event.provider : "Microsoft-Windows-Sysmon/Operational"
AND ((winlog.event_id : "3" AND message : (*cavecreekbicyclefestival*
OR *cave* AND *creek* AND *bicycle*)) OR (winlog.event_id : ("11"
OR "15") AND message : (*.html* OR *.php* OR *.htm* OR *.js*)) OR
(winlog.event_id : "1" AND message : (*chrome* OR *CHROME.EXE*
OR *firefox* OR *FIREFOX.EXE* OR *iexplore* OR *IEXPLORE.EXE*
OR *msedge* OR
MSEDGE.EXE*)))
```

- (b) Aggregate KQL Query:

```
event.provider : "Microsoft-Windows-Sysmon/Operational"
AND ((winlog.event_id : "3" AND message : (*cavecreekbicyclefestival*
OR *cave* AND *creek* AND *bicycle*)) OR (winlog.event_id : ("11"
OR "15") AND message : (*.html* OR *.php* OR *.htm* OR *.js*)) OR
```

```
(winlog.event_id : "1" AND message : (*chrome* OR *CHROME.EXE*
OR *firefox* OR *FIREFOX.EXE* OR *iexplore* OR *IEXPLORE.EXE*
OR *msedge* OR
MSEDGE.EXE*)))
```

- (c) The queries monitor for interactions with the compromised bicycle festival website hosting a fake UBS banking page. They track network connections to the domain, web-related file operations that might indicate phishing page delivery, and browser process execution that could signal victim visits. The same comprehensive detection approach serves both individual and aggregate queries, as we're focusing on identifying any potential interaction with this specific compromised domain and its phishing content.

41. RT @James\_inthe\_box: #agenttesla via highjacked subject #malspam

<https://t.co/FQqzTN5QYM> active panel at 31.220.40[.]22/~whoizzup/bon/Webâ

- (a) 31.220.40.22 (Command and Control: Web Protocols (T1071.001)):
- ```
event.provider : "Microsoft-Windows-Sysmon/Operational"
AND ( (winlog.event_id : "3" AND message : *31.220.40.22*) OR (win-
log.event_id : "1" AND message : (*powershell* OR
POWERSHELL.EXE* OR *cmd.exe* OR *CMD.EXE*)) )
```
- (b) <http://31.220.40.22/> whoizzup/bon/Web (Command and Control: Web Protocols (T1071.001)):
- ```
event.provider : "Microsoft-Windows-Sysmon/Operational"
AND ((winlog.event_id : "3" AND message : *31.220.40.22*) OR (win-
log.event_id : "1" AND message : (*powershell* OR
POWERSHELL.EXE* OR *cmd.exe* OR *CMD.EXE*)))
```
- (c) Aggregate KQL Query:
- ```
event.provider : "Microsoft-Windows-Sysmon/Operational"
```

```

AND ( (winlog.event_id : "3" AND message : ( *31.220.40.22* OR
(*31.220.40.22* AND (*whoizzup* OR *bon* OR *Web*)) )) OR (win-
log.event_id : ("11" OR "15") AND message : (*whoizzup* OR
*bon* OR *Web*)) OR (winlog.event_id : "1" AND message : (*pow-
ershell* OR *POWERSHELL.EXE* OR *cmd.exe* OR *CMD.EXE*))
)

```

- (d) The queries focus on detecting AgentTesla command and control activity. They monitor network connections to both the specific IP address and the full C2 panel path, while also watching for common process execution patterns associated with this malware family. The aggregate query combines these elements to provide visibility into both direct C2 communication and potential malware execution activities.

42. RT @infosecn1nja: If powershell.exe & dlls cmd.exe certutil.exe bitsadmin.exe ftp.exe x/copy.exe and print.exe is already to blocked?â

- (a) powershell.exe (Command and Scripting Interpreter (T1059)):
 event.provider : "Microsoft-Windows-Sysmon/Operational"
 AND winlog.event_id : "1" AND message : (*powershell.exe* OR *POWERSHELL.EXE*)
- (b) cmd.exe (Command and Scripting Interpreter (T1059)):
 event.provider : "Microsoft-Windows-Sysmon/Operational"
 AND winlog.event_id : "1" AND message : (*cmd.exe* OR *CMD.EXE*)
- (c) certutil.exe (Command and Scripting Interpreter (T1059)):
 event.provider : "Microsoft-Windows-Sysmon/Operational"
 AND winlog.event_id : "1" AND message : (*certutil.exe* OR *CERTUTIL.EXE*)
- (d) bitsadmin.exe (Command and Scripting Interpreter (T1059)):

event.provider : "Microsoft-Windows-Sysmon/Operational"

AND winlog.event_id : "1" AND message : (*bitsadmin.exe* OR *BITSADMIN.EXE*)

- (e) ftp.exe (Command and Scripting Interpreter (T1059)):

event.provider : "Microsoft-Windows-Sysmon/Operational"

AND winlog.event_id : "1" AND message :

(*ftp.exe* OR *FTP.EXE*)

- (f) copy.exe (Command and Scripting Interpreter (T1059)):

event.provider : "Microsoft-Windows-Sysmon/Operational"

AND winlog.event_id : "1" AND message :

(*copy.exe* OR *COPY.EXE*)

- (g) print.exe (Command and Scripting Interpreter (T1059)):

event.provider : "Microsoft-Windows-Sysmon/Operational"

AND winlog.event_id : "1" AND message :

(*print.exe* OR *PRINT.EXE*)

- (h) Aggregate KQL Query:

event.provider : "Microsoft-Windows-Sysmon/Operational"

AND winlog.event_id : "1" AND message : (*powershell.exe* OR *POWERSHELL.EXE* OR *cmd.exe* OR *CMD.EXE* OR *certutil.exe* OR *CERTUTIL.EXE* OR *bitsadmin.exe* OR *BITSADMIN.EXE* OR *ftp.exe* OR *FTP.EXE* OR *copy.exe* OR *COPY.EXE* OR *print.exe* OR *PRINT.EXE*)

- (i) The queries monitor execution of commonly abused Windows utilities. Each binary is tracked individually through process creation events, allowing for granular detection and potential blocking of specific tools. The aggregate query combines monitoring of all these binaries to provide com-

prehensive visibility into potential abuse of these utilities, which are frequently leveraged in various attack chains.

43. RT @sysopfb: Stealing NTLM hashes with

C:\windows\system32\nltest.exe on Windows 10

<https://t.co/JBd2YtCASj>

(a) C:\windows\system32\nltest.exe

(Exploitation for Credential Access (T1212)):

event.provider : "Microsoft-Windows-Sysmon/Operational"

AND ((winlog.event_id : "1" AND message : (*nltest.exe* OR *NL-TEST.EXE*)) OR (winlog.event_id : "7" AND message : (*netapi32.dll* OR *secur32.dll* OR *ntdsapi.dll*)) OR (winlog.event_id : "3" AND message : (*IPC\$* OR *ADMIN\$* OR *C\$*)))

(b) Aggregate KQL Query:

event.provider : "Microsoft-Windows-Sysmon/Operational"

AND ((winlog.event_id : "1" AND message : (*nltest.exe* OR *NL-TEST.EXE*)) OR (winlog.event_id : "7" AND message : (*netapi32.dll* OR *secur32.dll* OR *ntdsapi.dll*)) OR (winlog.event_id : "3" AND message : (*IPC\$* OR *ADMIN\$* OR *C\$*)))

(c) The queries monitor for potential abuse of nltest.exe for credential access. They track execution of the nltest.exe binary, loading of network and security-related DLLs commonly used in domain operations, and network connections to administrative shares that might indicate credential capture attempts. The same detection approach serves both individual and aggregate queries since we're focusing on a specific binary and its associated artifacts used in NTLM hash theft.

44. Remcos RAT delivered via fake CCICM international debt recovery service

<https://t.co/gZmcXaudn2>

<https://t.co/Zst5TKPeBA>

- (a) <https://t.co/gZmcXaudn2> (Spearphishing Link (T1566.002)):

```
event.provider : "Microsoft-Windows-Sysmon/Operational"
AND ( (winlog.event_id : "3" AND message : (*remcos* OR *CCICM*
OR *debt* OR *recovery*)) OR (winlog.event_id : ("11" OR "15") AND
message : (*.html* OR *.php* OR *.htm*)) OR (winlog.event_id : "1"
AND message : (*chrome* OR *CHROME.EXE* OR *firefox* OR *FIRE-
FOX.EXE* OR *iexplore* OR *IEXPLORE.EXE* OR *msedge* OR
MSEDGE.EXE*)) )
```

- (b) <https://t.co/Zst5TKPeBA> (Spearphishing Link (T1566.002)):

```
event.provider : "Microsoft-Windows-Sysmon/Operational"
AND ( (winlog.event_id : "3" AND message : (*Zst5TKPeBA* OR *rem-
cos* OR *CCICM* OR *debt* OR *recovery*)) OR (winlog.event_id :
("11" OR "15") AND message : (*.html* OR *.php* OR *.htm*)) )
```

- (c) Aggregate KQL Query:

```
event.provider : "Microsoft-Windows-Sysmon/Operational"
AND ( (winlog.event_id : "3" AND message : (*remcos* OR *CCICM*
OR *debt* OR *recovery*)) OR (winlog.event_id : ("11" OR "15") AND
message : (*.html* OR *.php* OR *.htm*)) OR (winlog.event_id : "1"
AND message : (*chrome* OR *CHROME.EXE* OR *firefox* OR *FIRE-
FOX.EXE* OR *iexplore* OR *IEXPLORE.EXE* OR *msedge* OR
MSEDGE.EXE*)) )
```

- (d) The queries monitor for both phishing delivery and Remcos RAT activity. They track network connections and file operations related to the fake debt recovery service, while also monitoring browser processes that might

indicate victim interaction with the phishing content. Additional focus is placed on potential Remcos RAT-specific indicators. The aggregate query combines these elements to provide visibility across both the social engineering and malware deployment phases of the attack.

45. ps66uk that is smokeloader delivering trickbot

<https://t.co/z9LrmgqLu0> quite different to the lyreco fake which was gozi

<https://t.co/3h0IDSfWKu>

(a) <https://t.co/z9LrmgqLu0> (Application Layer Protocol (T1071)):

event.provider : "Microsoft-Windows-Sysmon/Operational"

AND ((winlog.event_id : "3" AND message : (*smoke* OR *trickbot*
OR *.onion* OR *gate.php* OR *post.php*)) OR (winlog.event_id : "1"
AND message : (*rundll32* OR

RUNDLL32.EXE* OR *svchost* OR *SVCHOST.EXE*)))

(b) <https://t.co/3h0IDSfWKu> (Application Layer Protocol (T1071)):

event.provider : "Microsoft-Windows-Sysmon/Operational"

AND ((winlog.event_id : "3" AND message : (*gozi* OR *ursnif* OR
.onion OR *gate.php* OR *post.php*)) OR (winlog.event_id : "1" AND
message : (*rundll32* OR

RUNDLL32.EXE* OR *svchost* OR *SVCHOST.EXE*)))

(c) Aggregate KQL Query:

event.provider : "Microsoft-Windows-Sysmon/Operational"

AND ((winlog.event_id : "3" AND message : (*smoke* OR *trickbot*
OR *gozi* OR *ursnif* OR *.onion* OR *gate.php* OR *post.php*)) OR
(winlog.event_id : "1" AND message : (*rundll32* OR

RUNDLL32.EXE* OR *svchost* OR *SVCHOST.EXE*)) OR (

winlog.event_id : ("7" OR "8") AND message : (*user32.dll* OR *ker-

nel32.dll* OR *advapi32.dll*)))

- (d) The queries target the command and control activity of multiple malware families. They monitor network connections for indicators of Smokeloader, Trickbot, and Gozi/Ursnif C2 traffic, while also tracking process creation and DLL loading patterns common to these malware families. The aggregate query combines these elements to provide comprehensive visibility across the various malware families' communication patterns and execution behaviors.

46. @ps66uk @malwrhunterteam they are rar not 7z. rename to rar & it extracts #Lokibot
<https://t.co/eARahkO5ML>
<https://t.co/uDTum544vb>

- (a) rar (Masquerade File Type (T1036.002)):

event.provider : "Microsoft-Windows-Sysmon/Operational"

AND ((winlog.event_id : ("11" OR "15") AND message : (*.rar* OR *winrar*)) OR (winlog.event_id : "1" AND message : (*winrar* OR *WINRAR.EXE*)))

- (b) 7z (Masquerade File Type (T1036.002)):

event.provider : "Microsoft-Windows-Sysmon/Operational"

AND ((winlog.event_id : ("11" OR "15") AND message : (*.7z* OR *7zip*)) OR (winlog.event_id : "1" AND message : (*7z* OR *7ZG.EXE* OR *7zFM.exe*)))

- (c) rename to rar (Deobfuscate/Decode Files or Information (T1140)):

event.provider : "Microsoft-Windows-Sysmon/Operational"

AND winlog.event_id : ("11" OR "15") AND message : (*rename* OR *ren* OR *.rar*)

- (d) extracts (Deobfuscate/Decode Files or Information (T1140)):

```
event.provider : "Microsoft-Windows-Sysmon/Operational"
AND ( (winlog.event_id : "1" AND message : (*winrar* OR *WIN-
RAR.EXE* OR *7z* OR *7ZG.EXE*)) OR (winlog.event_id : ("11" OR
"15") AND message : (*extract* OR *expand*)) )
```

- (e) <https://t.co/eARahkO5ML> (Ingress Tool Transfer (T1105)):

```
event.provider : "Microsoft-Windows-Sysmon/Operational"
AND ( (winlog.event_id : "3" AND message : (*lokibot* OR *loki* OR
*.php*)) OR (winlog.event_id : ("11" OR "15") AND message : (*.exe*
OR *.dll* OR *.rar* OR *.7z*)) )
```

- (f) <https://t.co/uDTum544vb> (Ingress Tool Transfer (T1105)):

```
event.provider : "Microsoft-Windows-Sysmon/Operational"
AND ( (winlog.event_id : "3" AND message : (*lokibot* OR *loki* OR
*.php*)) OR (winlog.event_id : ("11" OR "15") AND message : (*.exe*
OR *.dll* OR *.rar* OR *.7z*)) )
```

- (g) Aggregate KQL Query:

```
event.provider : "Microsoft-Windows-Sysmon/Operational"
AND ( (winlog.event_id : ("11" OR "15") AND message : ( *.rar* OR
*winrar* OR *.7z* OR *7zip* OR *rename* OR *ren* OR *extract* OR
*expand* OR *.exe* OR *.dll* OR *lokibot* OR *loki* ) ) OR (win-
log.event_id : "1" AND message : ( *winrar* OR *WINRAR.EXE* OR
*7z* OR *7ZG.EXE* OR *7zFM.exe* ) ) OR (winlog.event_id : "3" AND
message : (*lokibot* OR *loki* OR *.php*)) )
```

- (h) The queries track multiple stages of this obfuscated Lokibot delivery. They monitor for both 7z and RAR-related file operations and process execution, with specific focus on file renaming activities that might indicate the obfuscation technique. Archive extraction is tracked through both pro-

cess execution and file operations. Network and file indicators specific to Lokibot are monitored to detect the final payload. The aggregate query combines these elements to provide visibility across the full attack chain, from initial archive handling through malware execution.

47. hawkeye keylogger fake shipping invoice

<https://app.any.run/tasks/2bff3a14-63b1-40a5-b78c-96e924c39ab3>

<https://t.co/Qfk75SPIo2>

(a) hawkeye keylogger (Input Capture (T1056.001)):

```
event.provider : "Microsoft-Windows-Sysmon/Operational"
AND ( (winlog.event_id : "1" AND message : (*hawkeye* OR *keylogger*
OR *rundll32* OR
RUNDLL32.EXE*)) OR (winlog.event_id : ("7" OR "8") AND message :
(*user32.dll* OR *kernel32.dll* OR *advapi32.dll*)) OR (winlog.event_id
: "13" AND message :
(*HKEY_LOCAL_MACHINE
\\SOFTWARE\\Microsoft\\Windows\\
CurrentVersion\\Run*
OR *HKEY_CURRENT_USER
\\Software\\Microsoft\\Windows\\
CurrentVersion\\Run*)) )
```

(b) <https://app.any.run/tasks/2bff3a14-63b1-40a5-b78c-96e924c39ab3> (Input Capture (T1056.001)):

```
event.provider : "Microsoft-Windows-Sysmon/Operational"
AND ( (winlog.event_id : "1" AND message : (*rundll32* OR
RUNDLL32.EXE* OR *regsvr32* OR *REGSVR32.EXE*)) OR (win-
log.event_id : "3" AND message : *invoice*) OR (winlog.event_id : ("11"
```

OR "15") AND message : (*invoice* OR *.exe* OR *.dll*)))

(c) Aggregate KQL Query:

event.provider : "Microsoft-Windows-Sysmon/Operational"

AND ((winlog.event_id : "1" AND message : (*hawkeye* OR *keylogger* OR *rundll32* OR

RUNDLL32.EXE* OR *regsvr32* OR *REGSVR32.EXE*)) OR (winlog.event_id : ("7" OR "8") AND message : (*user32.dll* OR *kernel32.dll* OR *advapi32.dll*)) OR (winlog.event_id : "13" AND message

:

(*HKEY_LOCAL_MACHINE

\\SOFTWARE\\Microsoft\\Windows\\

CurrentVersion\\Run* OR

HKEY_CURRENT_USER

\\Microsoft\\Windows\\

CurrentVersion\\Run*)) OR (winlog.event_id : "3" AND message : *invoice*) OR (winlog.event_id : ("11" OR "15") AND message : (*invoice* OR *.exe* OR *.dll*)))

(d) The queries monitor both the phishing delivery mechanism and the keylogger's activity. They track shipping invoice-themed phishing indicators while monitoring process creation, DLL loading, and registry modifications characteristic of HawkEye keylogger. The aggregate query combines these elements to provide visibility across both the social engineering and malware execution phases of the attack.

48. Fake HSBC Your HSBC application documents delivers Trickbot via Microsoft

Equation Editor Exploits

<https://t.co/FcoeXg9imH>

<https://t.co/JfoDIIfwej3>

- (a) <https://t.co/FcoeXg9imH> (Exploitation for Client Execution (T1203)):

```
event.provider : "Microsoft-Windows-Sysmon/Operational"
AND ( (winlog.event_id : "1" AND message : (*eqnedt32.exe* OR
EQNEDT32.EXE* OR *winword.exe* OR *WINWORD.EXE*)) OR (win-
log.event_id : ("7" OR "8") AND message : (*equation.3* OR *equation*
OR
EQNEDT32.EXE*)) OR (winlog.event_id : ("11" OR "15") AND mes-
sage : (*.doc* OR *.docx* OR *hsbc*)) )
```

- (b) <https://t.co/JfoDIIfwej3> (Exploitation for Client Execution (T1203)):

```
event.provider : "Microsoft-Windows-Sysmon/Operational"
AND ( (winlog.event_id : "3" AND message : (*trickbot* OR *gate.php*
OR *post.php* OR *hsbc*)) OR (winlog.event_id : "1" AND message
: (*svchost* OR *SVCHOST.EXE*)) OR (winlog.event_id : "13" AND
message :
(*HKEY_CURRENT_USER
\\Software\\Microsoft\\Windows\\
CurrentVersion\\Run* OR *CurrentVersion\\Run*)) )
```

- (c) Aggregate KQL Query:

```
event.provider : "Microsoft-Windows-Sysmon/Operational"
AND ( (winlog.event_id : "1" AND message : ( *eqnedt32.exe* OR
EQNEDT32.EXE* OR *winword.exe* OR *WINWORD.EXE* OR *sv-
chost* OR *SVCHOST.EXE* ) ) OR (winlog.event_id : ("7" OR "8")
AND message : (*equation.3* OR *equation* OR
EQNEDT32.EXE*)) OR (winlog.event_id : ("11" OR "15") AND mes-
sage : (*.doc* OR *.docx* OR *hsbc*)) OR (winlog.event_id : "3" AND
message : (*trickbot* OR *gate.php* OR *post.php* OR *hsbc*)) OR
(winlog.event_id : "13" AND message : ( *HKEY_CURRENT_USER
```

```
\\Software\\Microsoft\\Windows\\
CurrentVersion\\Run* OR *CurrentVersion
Run* )) )
```

- (d) The queries monitor this multi-stage attack chain from initial exploitation through Trickbot execution. They track Equation Editor process execution and related DLL loading that might indicate CVE exploitation, along with Word processes handling the malicious document. Trickbot activity is detected through characteristic process creation, network connections, and persistence mechanisms. The aggregate query combines these elements to provide visibility across the full attack flow, from initial document opening through banking malware execution.

49. Fake Payment recovery email spoofing CCICM international debt recovery service delivers Remcos rat via Microsoft Equation Editor Exploits

<https://t.co/a4tIIgqJDd>

<https://t.co/Tm7iqhoWzD>

- (a) CCICM international debt recovery

(Spearphishing Attachment (T1566.001)):

event.provider : "Microsoft-Windows-Sysmon/Operational"

AND ((winlog.event_id : ("11" OR "15") AND message : (*CCICM* OR *debt* OR *recovery* OR *.doc* OR *.docx*)) OR (winlog.event_id : "1" AND message : (*outlook.exe* OR *OUTLOOK.EXE* OR *thunderbird.exe* OR *THUNDERBIRD.EXE*)))

- (b) Remcos rat (Spearphishing Attachment (T1566.001)):

event.provider : "Microsoft-Windows-Sysmon/Operational"

AND ((winlog.event_id : "3" AND message : (*remcos* OR *rat* OR *.exe*)) OR (winlog.event_id : "1" AND message : (*svchost* OR *SV-

CHOST.EXE*)) OR (winlog.event_id : ("11" OR "15") AND message : *.exe*)

(c) Microsoft Equation Editor Exploits

(Spearphishing Attachment (T1566.001)):

event.provider : "Microsoft-Windows-Sysmon/Operational"

AND ((winlog.event_id : "1" AND message : (*eqnedt32.exe* OR EQNEDT32.EXE* OR *winword.exe* OR *WINWORD.EXE*)) OR (winlog.event_id : ("7" OR "8") AND message : (*equation.3* OR *equation* OR EQNEDT32.EXE*)))

(d) Microsoft Equation Editor Exploits (Exploitation for Client Execution (T1203)):

event.provider : "Microsoft-Windows-Sysmon/Operational"

AND ((winlog.event_id : "1" AND message : (*eqnedt32.exe* OR EQNEDT32.EXE*)) OR (winlog.event_id : ("7" OR "8") AND message : (*equation.3* OR *equation* OR EQNEDT32.EXE*)) OR (winlog.event_id : "13" AND message : *SOFTWARE\\Microsoft\\Office*))

(e) Aggregate KQL Query:

event.provider : "Microsoft-Windows-Sysmon/Operational"

AND ((winlog.event_id : "1" AND message : (*outlook.exe* OR *OUTLOOK.EXE* OR *thunderbird.exe* OR *THUNDERBIRD.EXE* OR *svchost* OR *SVCHOST.EXE* OR *eqnedt32.exe* OR EQNEDT32.EXE* OR *winword.exe* OR *WINWORD.EXE*)) OR (winlog.event_id : ("7" OR "8") AND message : (*equation.3* OR *equation* OR EQNEDT32.EXE*)) OR (winlog.event_id : ("11" OR "15") AND mes-

sage : (*CCICM* OR *debt* OR *recovery* OR *.doc* OR *.docx* OR *.exe*)) OR (winlog.event_id : "3" AND message : (*remcos* OR *rat* OR *.exe*)) OR (winlog.event_id : "13" AND message : *SOFTWARE\\Microsoft\\Office*)

- (f) The queries track multiple stages of this sophisticated attack chain. Initial phishing detection focuses on debt recovery-themed email content and document handling. The Equation Editor exploitation is monitored through process creation and DLL loading patterns. Remcos RAT activity is detected through process creation, network connections, and file operations. The aggregate query combines these elements to provide visibility across the full attack flow, from initial phishing through exploitation and RAT deployment.

50. RT @Techhelplistcom: 86.105.53.140 hundreds of open dir domains. hundreds of phishing sites kits. a few pony panels.

<https://t.co/jGqeqTâ>

- (a) 86.105.53.140 (Phishing (T1566)):

event.provider : "Microsoft-Windows-Sysmon/Operational"
 AND ((winlog.event_id : "3" AND message : *86.105.53.140*) OR (winlog.event_id : ("11" OR "15") AND message : (*.html* OR *.php* OR *.htm*)) OR (winlog.event_id : "1" AND message : (*chrome* OR *CHROME.EXE* OR *firefox* OR *FIREFOX.EXE* OR *iexplore* OR *IEXPLORE.EXE* OR *msedge* OR *MSEDGE.EXE*)))

- (b) 86.105.53.140 (Server Software Component (T1505)):

event.provider : "Microsoft-Windows-Sysmon/Operational"
 AND ((winlog.event_id : "3" AND message : *86.105.53.140*) OR (win-

```
log.event_id : ("11" OR "15") AND message : (*panel* OR *pony* OR
*.php*)) OR (winlog.event_id : "1" AND message : (*w3wp.exe* OR
*W3WP.EXE* OR *httpd.exe* OR *HTTPD.EXE* OR *nginx.exe* OR
*NGINX.EXE*)) )
```

(c) Aggregate KQL Query:

```
event.provider : "Microsoft-Windows-Sysmon/Operational"
AND ( (winlog.event_id : "3" AND message : *86.105.53.140*) OR (win-
log.event_id : ("11" OR "15") AND message : (*.html* OR *.php* OR
*.htm* OR *panel* OR *pony*)) OR (winlog.event_id : "1" AND message
: ( *chrome* OR *CHROME.EXE* OR *firefox* OR *FIREFOX.EXE*
OR *iexplore* OR *IEXPLORE.EXE* OR *msedge* OR
MSEDGE.EXE* OR *w3wp.exe* OR *W3WP.EXE* OR *httpd.exe* OR
*HTTPD.EXE* OR *nginx.exe* OR *NGINX.EXE* )) )
```

(d) The queries monitor both the client-side phishing activity and server-side panel components. They track network connections to the known malicious IP, web-related file operations that might indicate phishing kit presence, and both browser and web server process execution. The aggregate query combines these elements to detect both phishing delivery and backend panel operations, with particular focus on the Pony panels' infrastructure.

51. Project email from Georgia (85.118.97.87) word doc attachment

<https://t.co/>

EVYbwHLoPs with DDE update links prompts to update from

[hxxp://test1.ru/](https://test1.ru/)

newbuild/tiii.php?stats=send&thread=0 couldn't access site but HA shows .bat files & possibly this payload

<https://t.co/cWy4TR3S4o>

<https://t.co/jPOVnCWg9f>

- (a) `hxxp://test1.ru/newbuild/tiii.php?stats=send&thread=0` (Drive-by Compromise (T1189)):

`event.provider : "Microsoft-Windows-Sysmon/Operational"`

`AND ((winlog.event_id : "3" AND message : (*test1.ru* OR *new-build* OR *tiii.php*)) OR (winlog.event_id : "1" AND message : (*winword.exe* OR *WINWORD.EXE*)) OR (winlog.event_id : ("11" OR "15") AND message : (*.doc* OR *.docx*)))`

- (b) .bat files (Command and Scripting Interpreter (T1059)):

`event.provider : "Microsoft-Windows-Sysmon/Operational"`

`AND ((winlog.event_id : ("11" OR "15") AND message : *.bat*) OR (winlog.event_id : "1" AND message : (*cmd.exe* OR *CMD.EXE*)) OR (winlog.event_id : "7" AND message : *cmd.exe*))`

- (c) Aggregate KQL Query:

`event.provider : "Microsoft-Windows-Sysmon/Operational"`

`AND ((winlog.event_id : "3" AND message : (*test1.ru* OR *new-build* OR *tiii.php*)) OR (winlog.event_id : ("11" OR "15") AND message : (*.doc* OR *.docx* OR *.bat*)) OR (winlog.event_id : "1" AND message : (*winword.exe* OR *WINWORD.EXE* OR *cmd.exe* OR *CMD.EXE*)) OR (winlog.event_id : "7" AND message : *cmd.exe*))`

- (d) The queries monitor both the initial Word document exploitation and subsequent batch file execution. They track network connections to the malicious domain, Word process execution and document handling, and potential batch file creation and execution through `cmd.exe`. The aggregate query combines these elements to detect activity across the full attack chain, from initial document opening through potential batch file execu-

tion.

52. Invoice email with rtf attachment

<https://t.co/Q0dcKuv0W2> uses bitsadmin to download #lokibot exe payload

<https://t.co/tpreiastephenville> com/jazz.exe

<https://t.co/dI1sLXbLRD> more details pasted here

<https://t.co/IKn6bugQi1> cc @CyrusOne @enom #phishing #malware #takeit-down

<https://t.co/HDwNtscpvq>

(a) tpreiastephenville[.]com/jazz.exe (Ingress Tool Transfer (T1105)):

event.provider : "Microsoft-Windows-Sysmon/Operational"

AND ((winlog.event_id : "3" AND message : (*tpreiastephenville* OR *jazz.exe*)) OR (winlog.event_id : ("11" OR "15")

AND message : *jazz.exe*))

(b) rtf attachment (User Execution (T1204)):

event.provider : "Microsoft-Windows-Sysmon/Operational"

AND ((winlog.event_id : ("11" OR "15") AND message : (*.rtf* OR *invoice*)) OR (winlog.event_id : "1" AND message : (*winword.exe* OR *WINWORD.EXE*)))

(c) bitsadmin (Command and Scripting Interpreter (T1059)):

event.provider : "Microsoft-Windows-Sysmon/Operational"

AND ((winlog.event_id : "1" AND message : (*bitsadmin* OR *BITSADMIN.EXE*)) OR (winlog.event_id : "13" AND message : *BITS*) OR (winlog.event_id : "3" AND message : *bits*))

(d) phishing (Phishing (T1566)):

event.provider : "Microsoft-Windows-Sysmon/Operational"

AND ((winlog.event_id : ("11" OR "15") AND message : (*invoice* OR

```
*.rtf*)) OR (winlog.event_id : "1" AND message : (*outlook.exe* OR
*OUTLOOK.EXE* OR *thunderbird.exe* OR *THUNDERBIRD.EXE*))
)
```

(e) Aggregate KQL Query:

```
event.provider : "Microsoft-Windows-Sysmon/Operational"
AND ( (winlog.event_id : "3" AND message : ( *tpreiastephenville* OR
*jazz.exe* OR *bits* OR *lokibot* )) OR (winlog.event_id : ("11" OR
"15") AND message : ( *.rtf* OR *invoice* OR *jazz.exe* )) OR (win-
log.event_id : "1" AND message : ( *winword.exe* OR *WINWORD.EXE*
OR *bitsadmin* OR *BITSADMIN.EXE* OR *outlook.exe* OR *OUT-
LOOK.EXE* OR *thunderbird.exe* OR *THUNDERBIRD.EXE* )) OR
(winlog.event_id : "13" AND message : *BITS*) )
```

(f) The queries monitor multiple stages of this Lokibot delivery chain. They track initial phishing email and RTF document handling, followed by BITS abuse for malware download. Network connections and file operations related to the Lokibot executable are monitored. The aggregate query combines these elements to provide visibility across the full attack chain, from initial phishing through BITS-based malware retrieval.

53. RT @SevenLayerJedi: #Adwind #RAT #Malware Executable JAR hosted on dropbox from #Phishing Email
<https://t.co/jxFwSt2gVr> #hybridAnalysis #ANâ

(a) dropbox (Phishing (T1566)):

```
event.provider : "Microsoft-Windows-Sysmon/Operational"
AND ( (winlog.event_id : "3" AND message : (*dropbox* OR *.drop-
box.com*)) OR (winlog.event_id : ("11" OR "15") AND message : (*.html*
OR *.htm* OR *.eml*)) OR (winlog.event_id : "1" AND message : (*out-
```

```
look.exe* OR *OUTLOOK.EXE* OR *thunderbird.exe* OR *THUNDER-
BIRD.EXE*)) )
```

- (b) Executable JAR (Software Packing (T1027)):

```
event.provider : "Microsoft-Windows-Sysmon/Operational"
AND ( (winlog.event_id : ("11" OR "15") AND message : (*.jar* OR
*adwind*)) OR (winlog.event_id : "1" AND message : (*java* OR
JAVA.EXE* OR *javaw.exe* OR *JAVAW.EXE*)) OR (winlog.event_id
: "7" AND message : (*java* OR *jvm.dll* OR *javaw*)) )
```

- (c) Aggregate KQL Query:

```
event.provider : "Microsoft-Windows-Sysmon/Operational"
AND ( (winlog.event_id : "3" AND message : (*dropbox* OR *.drop-
box.com*)) OR (winlog.event_id : ("11" OR "15") AND message : (*.jar*
OR *adwind* OR *.html* OR *.htm* OR *.eml*)) OR (winlog.event_id
: "1" AND message : ( *java* OR *JAVA.EXE* OR *javaw.exe* OR
*JAVAW.EXE* OR *outlook.exe* OR *OUTLOOK.EXE* OR *thunder-
bird.exe* OR *THUNDERBIRD.EXE* )) OR (winlog.event_id : "7" AND
message : (*java* OR *jvm.dll* OR *javaw*)) )
```

- (d) The queries target both the phishing delivery mechanism and Java-based malware execution. They monitor Dropbox-related network connections and phishing email artifacts, while tracking JAR file operations and Java process execution patterns associated with Adwind RAT. The aggregate query combines these elements to provide visibility across both the phishing delivery and malware execution phases of the attack.

54. RT @K_N1kolenko: #malware #powershell #backdoor backdoor reverse shell tunnel over DNS

hxxp://dnslab.cf/module.ps1 VT 0/59

<https://t.co/Jâ>

(a) dnslab.cf (PowerShell (T1059.001)):

```
event.provider : "Microsoft-Windows-Sysmon/Operational"
AND ( (winlog.event_id : "3" AND message : *dnslab*) OR
(event.provider : ("Microsoft-Windows-PowerShell*" OR "Windows Pow-
erShell") AND message : (*dns* OR *tunnel* OR *reverse*)) )
```

(b) hxxp://dnslab.cf/module.ps1 (PowerShell (T1059.001)):

```
event.provider : "Microsoft-Windows-Sysmon/Operational"
AND ( (winlog.event_id : "3" AND message : (*dnslab* AND *mod-
ule.ps1*)) OR (winlog.event_id : ("11" OR "15") AND message : *.ps1*)
)
```

(c) Command and Scripting Interpreter (T1059):

```
event.provider : "Microsoft-Windows-Sysmon/Operational"
AND ( (winlog.event_id : "1" AND message : (*powershell* OR *POW-
ERSHELL.EXE*)) OR (winlog.event_id : "3" AND message : *dnslab*)
)
```

(d) Application Layer Protocol (T1071):

```
event.provider : "Microsoft-Windows-Sysmon/Operational"
AND ( (winlog.event_id : "3" AND message : (*dnslab* OR *DNS*))
OR (winlog.event_id : "22" AND message : *DNS*) )
```

(e) Aggregate KQL Query:

```
( (event.provider : "Microsoft-Windows-Sysmon/Operational"
AND ( (winlog.event_id : "1" AND message : (*powershell* OR *POW-
ERSHELL.EXE*)) OR (winlog.event_id : "3" AND message : (*dnslab*
OR *module.ps1* OR *DNS*)) OR (winlog.event_id : ("11" OR "15")
AND message : *.ps1*) OR (winlog.event_id : "22" AND message :
```

```
*DNS*) ) ) OR (event.provider : ("Microsoft-Windows-PowerShell*" OR
"Windows PowerShell") AND message : (*dns* OR *tunnel* OR *reverse*
OR *downloadstring* OR *IEX* OR *Invoke-Expression*)) )
```

- (f) The queries monitor this PowerShell-based DNS tunneling backdoor through multiple detection approaches. They track network connections to the malicious domain, PowerShell script download and execution patterns, and DNS protocol abuse indicators. The aggregate query combines these elements to provide visibility across the full attack chain, from initial script retrieval through command and control via DNS tunneling.

55. Malicious `#linux #Coinminer sha256:`

`9f1b067998c6fe243e5f8084b2b1b`

`c929dacab11247c07a857374ba798071a22` downloaded all the files from the drop-per script (logo4 file) download all the files here:

`https://t.co/acHsUzJOZo #Malware #CyberSecurity`

(a) `9f1b067998c6fe243e5f8084b2b1bc929dacab11247c07a857374ba798071a22`

(Resource Hijacking (T1496)):

```
event.provider : "Microsoft-Windows-Sysmon/Operational"
```

```
AND ( (winlog.event_id : "1" AND message : (*xmrig* OR *minerd*
OR *minergate* OR *cpuminer* OR *cgminer* OR *bfgminer* OR *eth-
miner*)) OR (winlog.event_id : ("11" OR "15") AND message : (*logo4*
OR *.sh* OR *miner*)) )
```

(b) `https://t.co/acHsUzJOZo` (Resource Hijacking (T1496)):

```
event.provider : "Microsoft-Windows-Sysmon/Operational"
```

```
AND ( (winlog.event_id : "3" AND message : (*pool* OR *mining*
OR *stratum* OR *xmr* OR *monero* OR *eth* OR *ethereum*)) OR
(winlog.event_id : "13" AND message : *CurrentVersion\\Run*) )
```


(c) Aggregate KQL Query:

```
event.provider : "Microsoft-Windows-Sysmon/Operational"
AND ( (winlog.event_id : "1" AND message : ( *xmrig* OR *minerd*
OR *minergate* OR *cpuminer* OR *cgminer* OR *bfgminer* OR *eth-
miner* )) OR (winlog.event_id : ("11" OR "15") AND message : (*logo4*
OR *.sh* OR *miner*)) OR (winlog.event_id : "3" AND message : (
*pool* OR *mining* OR *stratum* OR *xmr* OR *monero* OR *eth*
OR *ethereum* )) OR (winlog.event_id : "13" AND message : *Cur-
rentVersion\\Run* ) )
```

(d) The queries focus on detecting cryptocurrency mining activity and its supporting components. They monitor for execution of common mining software, shell script operations related to the dropper, and network connections to mining pools. Registry modifications for persistence are also tracked. The aggregate query combines these elements to provide visibility into both the initial deployment and ongoing operation of the mining malware.

56. #Hawkeye #Keylogger #Malware sha256:

1ae71a599df5bee528a93

7df40103713be27eb77e83b22cecffae883de4d82ec

<https://t.co/8ryG2thr4T> uses gmail to send logs

(a) 1ae71a599df5bee528a937df40103713be27eb77e83b22cecffae883de4d82ec

(Exfiltration Over Web Service (T1567)):

```
event.provider : "Microsoft-Windows-Sysmon/Operational"
AND ( (winlog.event_id : "1" AND message : (*rundll32* OR
RUNDLL32.EXE* OR *hawkeye* OR *keylogger*)) OR (winlog.event_id
: ("7" OR "8") AND message : (*user32.dll* OR *kernel32.dll* OR *ad-
```

vapi32.dll*)) OR (winlog.event_id : "13" AND message :
(*HKEY_LOCAL_MACHINE

\\SOFTWARE\\Microsoft\\Windows\\CurrentVersion\\Run* OR
HKEY_CURRENT_USER
\\Software\\Microsoft\\Windows\\CurrentVersion\\Run*)))

(b) gmail (Exfiltration Over Web Service (T1567)):

event.provider : "Microsoft-Windows-Sysmon/Operational"
AND ((winlog.event_id : "3" AND message : ((*gmail.com* AND
POST) OR *smtp.gmail.com* OR *imap.gmail.com*)) OR
(winlog.event_id : "1" AND message : (*outlook.exe* OR *OUTLOOK.EXE*
OR *thunderbird.exe* OR *THUNDERBIRD.EXE*)))

(c) Aggregate KQL Query:

event.provider : "Microsoft-Windows-Sysmon/Operational"
AND ((winlog.event_id : "1" AND message : (*rundll32* OR
RUNDLL32.EXE* OR *hawkeye* OR *keylogger* OR *outlook.exe* OR
OUTLOOK.EXE OR *thunderbird.exe* OR *THUNDERBIRD.EXE*
)) OR (winlog.event_id : ("7" OR "8") AND message : (*user32.dll* OR
kernel32.dll OR *advapi32.dll*)) OR (winlog.event_id : "13" AND mes-
sage : (*HKEY_LOCAL_MACHINE
\\SOFTWARE\\Microsoft\\Windows\\
CurrentVersion\\Run* OR *HKEY_CURRENT_USER
\\Software\\Microsoft\\Windows\\
CurrentVersion\\Run*)) OR (winlog.event_id : "3" AND message : ((*gmail.com* AND *POST*) OR *smtp.gmail.com* OR *imap.gmail.com*
)))

(d) The queries monitor both the keylogger's operation and its exfiltration

mechanism. They track process creation and DLL loading patterns characteristic of HawkEye, along with its persistence mechanisms through registry modifications. Email-based exfiltration is detected through monitoring of Gmail-related network connections, particularly SMTP and IMAP protocols. The aggregate query combines these elements to provide visibility across both the keylogging activity and subsequent data exfiltration.

57. interesting #trojan xml exploiting #DDE to run powershell script.

<https://t.co/hw8T32o4ts> connects to tiamos[.]co paste w/ info including IOC's and whois records on all contacted IP's @sudosev

@James_inthe_box @JAMESWT_MHT you know what this is? #Malware #Trojan #Exploit

<https://t.co/JslVroAEM0>

(a) tiamos[.]co (Exploitation for Client Execution (T1203)):

event.provider : "Microsoft-Windows-Sysmon/Operational"

AND ((winlog.event_id : "3" AND message : *tiamos*) OR

(winlog.event_id : ("11" OR "15") AND message : (*.xml* OR *.ps1*))

OR

(winlog.event_id : "1" AND message : (*powershell* OR *POWER-SHELL.EXE* OR *winword.exe* OR *WINWORD.EXE* OR *excel.exe* OR *EXCEL.EXE*)))

(b) Aggregate KQL Query:

((event.provider : "Microsoft-Windows-Sysmon/Operational"

AND ((winlog.event_id : "3" AND message : *tiamos*) OR

(winlog.event_id : ("11" OR "15") AND message : (*.xml* OR *.ps1*))

OR

(winlog.event_id : "1" AND message : (*powershell* OR *POWER-

SHELL.EXE* OR *winword.exe* OR *WINWORD.EXE* OR *excel.exe* OR *EXCEL.EXE*)))) OR (event.provider : ("Microsoft-Windows-PowerShell*" OR "Windows PowerShell") AND message : (*IEX* OR *Invoke-Expression* OR *FromBase64* OR *-enc* OR *hidden* OR *downloadstring*)))

- (c) The queries track multiple stages of this DDE-based attack chain. They monitor for Office process execution that might involve DDE abuse, XML and PowerShell script operations, and network connections to the malicious domain. Additional PowerShell-specific monitoring covers common obfuscation and execution patterns. The aggregate query combines these elements to provide visibility across the full attack flow, from initial DDE exploitation through PowerShell script execution.

58. RT @DissectMalware: Small #powershell code injects a #shellcode (which seems to be a #backdoor) in itself and executes the injected code. Pâ

- (a) powershell (Process Injection (T1055)):

event.provider : "Microsoft-Windows-Sysmon/Operational"
 AND ((winlog.event_id : "1" AND message : (*powershell* OR *POWERSHELL.EXE*)) OR (event.provider : ("Microsoft-Windows-PowerShell*" OR "Windows PowerShell") AND message : (*IEX* OR Invoke-Expression* OR *FromBase64* OR *-enc* OR *VirtualAlloc*)))

- (b) shellcode (Process Injection (T1055)):

event.provider :
 "Microsoft-Windows-Sysmon/Operational"
 AND ((winlog.event_id : ("7" OR "8") AND message : (*VirtualAlloc* OR *VirtualProtect* OR *WriteProcessMemory* OR *RtlMoveMem-

ory*)) OR (winlog.event_id : "10" AND message : (*OpenProcess* OR *CreateRemoteThread* OR *NtCreateThreadEx*)))

(c) backdoor (Process Injection (T1055)):

event.provider :

"Microsoft-Windows-Sysmon/Operational"

AND ((winlog.event_id : "3" AND message : (*connect* OR *bind* OR *reverse*)) OR (winlog.event_id : "1" AND message : (*cmd.exe* OR *CMD.EXE* OR *powershell* OR *POWERSHELL.EXE*)))

(d) Aggregate KQL Query:

((event.provider : "Microsoft-Windows-Sysmon/Operational"
AND ((winlog.event_id : "1" AND message : (*powershell* OR *POWERSHELL.EXE* OR *cmd.exe* OR *CMD.EXE*)) OR (winlog.event_id : ("7" OR "8") AND message : (*VirtualAlloc* OR *VirtualProtect* OR *WriteProcessMemory* OR *RtlMoveMemory*)) OR (winlog.event_id : "10" AND message : (*OpenProcess* OR *CreateRemoteThread* OR *NtCreateThreadEx*)) OR (winlog.event_id : "3" AND message : (*connect* OR *bind* OR *reverse*)))) OR (event.provider : ("Microsoft-Windows-PowerShell" OR "Windows PowerShell") AND message : (*IEX* OR *Invoke-Expression* OR *FromBase64* OR *-enc* OR *VirtualAlloc*)))

(e) The queries monitor this PowerShell-based process injection attack through multiple detection approaches. They track PowerShell execution with common obfuscation patterns, memory allocation and manipulation APIs commonly used in shellcode injection, and network activity that might indicate backdoor communication. The aggregate query combines these elements to provide visibility across the full attack chain, from initial PowerShell execution through shellcode injection and potential backdoor activity.

59. #Remcos #RAT #backdoor 11/65 detection on VT sha256:

fa685175e79c9bd2b514c7c61549a53170472028a148ed9de6317fb0c137998a try to
connect to: remrem.onmypc[.]net

<https://t.co/wIzouqIisH> #Malware #Infosec #CyberSecurity

<https://t.co/XGRar15ApU>

(a) fa685175e79c9bd2b514c7c61549a53170472028a148ed9de6317fb0c137998a

(Remote Access Tools (T1219)):

event.provider : "Microsoft-Windows-Sysmon/Operational"

AND ((winlog.event_id : "1" AND message : (*remcos* OR *rat* OR
svchost OR *SVCHOST.EXE*)) OR (winlog.event_id : ("7" OR "8")

AND message : (*user32.dll* OR *kernel32.dll* OR *advapi32.dll*)) OR

(winlog.event_id : "13" AND message :

(*HKEY_LOCAL_MACHINE

\\SOFTWARE\\Microsoft\\Windows\\

CurrentVersion\\Run*

OR *HKEY_CURRENT_USER

\\Software\\Microsoft\\Windows\\

CurrentVersion\\Run*)))

(b) remrem.onmypc[.]net (Remote Access Tools (T1219)):

event.provider : "Microsoft-Windows-Sysmon/Operational"

AND ((winlog.event_id : "3" AND message : (*remrem* OR
onmypc.net* OR *remcos*)) OR (winlog.event_id : "22" AND message

: (*remrem* OR *onmypc.net*)) OR (winlog.event_id : ("11" OR "15")

AND message : (*.exe* OR *.dll*)))

(c) Aggregate KQL Query:

event.provider : "Microsoft-Windows-Sysmon/Operational"

AND ((winlog.event_id : "1" AND message : (*remcos* OR *rat* OR

```
*svchost* OR *SVCHOST.EXE*)) OR (winlog.event_id : ("7" OR "8")
AND message : (*user32.dll* OR *kernel32.dll* OR *advapi32.dll*)) OR
(winlog.event_id : "13" AND message : ( *HKEY_LOCAL_MACHINE
\\SOFTWARE\\Microsoft\\Windows\\
CurrentVersion\\Run* OR *HKEY_CURRENT_USER
\\Software\\Microsoft\\Windows\\
CurrentVersion\\Run* )) OR (winlog.event_id : "3" AND message : (*rem-
rem* OR *onmypc.net* OR *remcos*)) OR (winlog.event_id : "22" AND
message : (*remrem* OR *onmypc.net*)) OR (winlog.event_id : ("11"
OR "15") AND message : (*.exe* OR *.dll*)) )
```

- (d) The queries monitor both the execution and network activity of the Remcos RAT. They track process creation, DLL loading, and persistence mechanisms through registry modifications characteristic of this malware. Network connections to the C2 domain are monitored through both direct connections and DNS queries. The aggregate query combines these elements to provide comprehensive visibility into the RAT's operation, from initial execution through command and control activity.

60. #Python #Backdoor / #Rootkit sample filename: office.exe sha256:
971a2f86020acb593f2eeea1226b9497edc295483b2e894e89538241b00aa071 connec-
tions and DNS request to: office16.homedns[.]org
<https://t.co/Yz24bmKOtG> IP: 104[.]131[.]38[.]231 hosted by @digitalocean

- (a) office.exe (System Binary Proxy Execution (T1218)):

```
event.provider : "Microsoft-Windows-Sysmon/Operational"
AND ( (winlog.event_id : "1" AND message : (*office.exe* OR *python*
OR *PYTHON.EXE* OR *py.exe* OR *PY.EXE*)) OR (winlog.event_id
: ("11" OR "15") AND message : *office.exe*) OR (winlog.event_id : "3"
```

AND message : (*office16.homedns.org* OR *104.131.38.231*)))

(b) 971a2f86020acb593f2eeea1226b9497edc295483b2e894e89538241b00aa071

(System Binary Proxy Execution (T1218)):

event.provider : "Microsoft-Windows-Sysmon/Operational"

AND ((winlog.event_id : "1" AND message : (*office.exe* OR *python*
OR *PYTHON.EXE* OR *rundll32* OR

RUNDLL32.EXE*)) OR (winlog.event_id : ("7" OR "8") AND message
: (*python* OR *python3*.dll* OR *python*.dll*)) OR (winlog.event_id
: "13" AND message :

(*HKEY_LOCAL_MACHINE

\\SOFTWARE\\Microsoft\\Windows\\

CurrentVersion\\Run*

OR *HKEY_CURRENT_USER

\\Software\\Microsoft\\Windows\\

CurrentVersion\\Run*)))

(c) Aggregate KQL Query:

event.provider : "Microsoft-Windows-Sysmon/Operational"

AND ((winlog.event_id : "1" AND message : (*office.exe* OR *python*
OR *PYTHON.EXE* OR *py.exe* OR *PY.EXE* OR *rundll32* OR

RUNDLL32.EXE*)) OR (winlog.event_id : ("11" OR "15") AND mes-
sage : *office.exe*) OR (winlog.event_id : ("7" OR "8") AND message :
(*python* OR *python3*.dll* OR *python*.dll*)) OR (winlog.event_id :

"13" AND message : (*HKEY_LOCAL_MACHINE

\\SOFTWARE\\Microsoft\\Windows\\

CurrentVersion\\Run* OR *HKEY_CURRENT_USER

\\Software\\Microsoft\\Windows\\

CurrentVersion\\Run*)) OR (winlog.event_id : "3" AND message : (*of-

office16.homedns.org* OR *104.131.38.231*)) OR (winlog.event_id : "22"
AND message : *office16.homedns.org*))

- (d) The queries monitor this Python-based backdoor through multiple detection approaches. They track the malicious executable's creation and execution, Python-related process and DLL activity, persistence attempts through registry modifications, and network communications to both the domain and IP address. DNS queries are specifically monitored for the C2 domain. The aggregate query combines these elements to provide visibility across the full attack chain, from initial execution through command and control activity.

61. RT @tmmalanalyst: Apr-19 2018(JST). Japanese MalSpam included HTML Link -> ZIP -> js -> Exe -> Infects #Ursnif #Malware. Leads js file VT:â

- (a) HTML Link (Spearphishing Attachment (T1566.001)):

event.provider : "Microsoft-Windows-Sysmon/Operational"
AND ((winlog.event_id : ("11" OR "15") AND message : (*.html* OR *.htm*)) OR (winlog.event_id : "1" AND message : (*outlook.exe* OR *OUTLOOK.EXE* OR *thunderbird.exe* OR THUNDERBIRD.EXE*)))

- (b) ZIP (Spearphishing Attachment (T1566.001)):

event.provider : "Microsoft-Windows-Sysmon/Operational"
AND ((winlog.event_id : ("11" OR "15") AND message : *.zip*) OR (winlog.event_id : "1" AND message : (*winrar* OR *WINRAR.EXE* OR *7z* OR *7Z.EXE*)))

- (c) js (Spearphishing Attachment (T1566.001)):

event.provider : "Microsoft-Windows-Sysmon/Operational"
AND ((winlog.event_id : ("11" OR "15") AND message : *.js*) OR

```
(winlog.event_id : "1" AND message : (*wscript.exe* OR
WSSCRIPT.EXE* OR *cscript.exe* OR *CSCRIPT.EXE*)) )
```

- (d) Exe (Spearphishing Attachment (T1566.001)):

```
event.provider : "Microsoft-Windows-Sysmon/Operational"
AND ( (winlog.event_id : ("11" OR "15") AND message : *.exe*) OR
(winlog.event_id : "1" AND message : (*rundll32* OR
RUNDLL32.EXE*)) )
```

- (e) Ursnif (Spearphishing Attachment (T1566.001)):

```
event.provider : "Microsoft-Windows-Sysmon/Operational"
AND ( (winlog.event_id : "3" AND message : (*ursnif* OR *gozi*)) OR
(winlog.event_id : "13" AND message : *CurrentVersion\\Run*) )
```

- (f) Aggregate KQL Query:

```
event.provider : "Microsoft-Windows-Sysmon/Operational"
AND ( (winlog.event_id : ("11" OR "15") AND message : ( *.html* OR
*.htm* OR *.zip* OR *.js* OR *.exe* )) OR (winlog.event_id : "1" AND
message : ( *outlook.exe* OR *OUTLOOK.EXE* OR *thunderbird.exe*
OR *THUNDERBIRD.EXE* OR *winrar* OR *WINRAR.EXE* OR *7z*
OR *7Z.EXE* OR *wscript.exe* OR
WSSCRIPT.EXE* OR *cscript.exe* OR *CSCRIPT.EXE* OR *rundll32*
OR
RUNDLL32.EXE* )) OR (winlog.event_id : "3" AND message : (*ursnif*
OR *gozi*)) OR (winlog.event_id : "13" AND message : *CurrentVer-
sion\\Run*) )
```

- (g) The queries monitor the complete infection chain of this Ursnif malware campaign. They track each stage from initial HTML link through archive extraction, script execution, and final payload deployment. File operations and process execution are monitored for each stage of the attack. Network

and registry indicators specific to Ursnif are tracked to detect successful infection. The aggregate query combines these elements to provide comprehensive visibility across all stages of the attack.

62. RT @avman1995: #Emotet

<https://t.co/evF6vHymRO> c2: 45.56.91.17:443/ drops:

<https://t.co/xZtp9uPhXb> dumped binaries:

<https://t.co/FUDcfmâ>

(a) 45.56.91.17:443

(Exfiltration Over Command and Control Channel (T1041)):

event.provider : "Microsoft-Windows-Sysmon/Operational"

AND ((winlog.event_id : "3" AND message : (*45.56.91.17* OR *443*))

OR (winlog.event_id : "1" AND message : (*outlook.exe* OR *OUT-

LOOK.EXE* OR *svchost* OR *SVCHOST.EXE*)) OR (winlog.event_id

: ("11" OR "15") AND message : (*.exe* OR *.dll* OR *emotet*)))

(b) Aggregate KQL Query:

event.provider : "Microsoft-Windows-Sysmon/Operational"

AND ((winlog.event_id : "3" AND message : (*45.56.91.17* OR *443*))

OR (winlog.event_id : "1" AND message : (*outlook.exe* OR *OUT-

LOOK.EXE* OR *svchost* OR *SVCHOST.EXE*)) OR (winlog.event_id

: ("11" OR "15") AND message : (*.exe* OR *.dll* OR *emotet*)) OR

(winlog.event_id : "13" AND message : (*HKEY_LOCAL_MACHINE

\\SOFTWARE\\Microsoft\\Windows\\

CurrentVersion\\Run* OR *HKEY_CURRENT_USER

\\Software\\Microsoft\\Windows\\

CurrentVersion\\Run*)))

(c) The queries monitor Emotet command and control activity and payload

operations. They track network connections to the C2 server, process execution patterns common to Emotet, and file operations related to dropped payloads. Registry modifications for persistence are also monitored. The aggregate query combines these elements to provide visibility across both C2 communication and local malware activity.

63. @FewAtoms notcleanedstubs:

<https://t.co/TVPVStLoqk>

<https://t.co/s7f11LXBi0>

<https://t.co/62eOzOYKcI> PDB stub:

<https://t.co/a84gqCBaha> obfuscated .net binary:

<https://t.co/mox5AwlJLV> deobfuscated final binary:

<https://t.co/n2thGrAinR>

(a) <https://t.co/TVPVStLoqk> (Obfuscated Files or Information (T1027)):

event.provider : "Microsoft-Windows-Sysmon/Operational"

AND ((winlog.event_id : "1" AND message : (*InstallUtil* OR *RegAsm* OR *RegSvcs* OR *.NET* OR *dnx* OR *dotnet*)) OR (winlog.event_id : ("7" OR "8") AND message : (*mscorlib.dll* OR *mscorlib.dll* OR *System.dll*)) OR (winlog.event_id : ("11" OR "15") AND message : (*.dll* OR *.exe* OR *.pdb*)))

(b) <https://t.co/s7f11LXBi0> (Obfuscated Files or Information (T1027)):

event.provider : "Microsoft-Windows-Sysmon/Operational"

AND ((winlog.event_id : "1" AND message : (*InstallUtil* OR *RegAsm* OR *RegSvcs* OR *.NET* OR *dnx* OR *dotnet*)) OR (winlog.event_id : ("7" OR "8") AND message : (*mscorlib.dll* OR *mscorlib.dll* OR *System.dll*)) OR (winlog.event_id : ("11" OR "15") AND message : (*.dll* OR *.exe* OR *.pdb*)))

(c) <https://t.co/62eOzOYKcI> (Obfuscated Files or Information (T1027)):

event.provider : "Microsoft-Windows-Sysmon/Operational"

AND ((winlog.event_id : "1" AND message : (*InstallUtil* OR *RegAsm* OR *RegSvcs* OR *.NET* OR *dnx* OR *dotnet*)) OR (winlog.event_id : ("7" OR "8") AND message : (*mscorlib.dll* OR *mscorlib.dll* OR *System.dll*)) OR (winlog.event_id : ("11" OR "15") AND message : (*.dll* OR *.exe* OR *.pdb*)))

(d) <https://t.co/a84gqCBaha> (Obfuscated Files or Information (T1027)):

event.provider : "Microsoft-Windows-Sysmon/Operational"

AND ((winlog.event_id : "1" AND message : (*InstallUtil* OR *RegAsm* OR *RegSvcs* OR *.NET* OR *dnx* OR *dotnet*)) OR (winlog.event_id : ("7" OR "8") AND message : (*mscorlib.dll* OR *mscorlib.dll* OR *System.dll*)) OR (winlog.event_id : ("11" OR "15") AND message : (*.dll* OR *.exe* OR *.pdb*)))

(e) <https://t.co/mox5AwlJLV> (Obfuscated Files or Information (T1027)):

event.provider : "Microsoft-Windows-Sysmon/Operational"

AND ((winlog.event_id : "1" AND message : (*InstallUtil* OR *RegAsm* OR *RegSvcs* OR *.NET* OR *dnx* OR *dotnet*)) OR (winlog.event_id : ("7" OR "8") AND message : (*mscorlib.dll* OR *mscorlib.dll* OR *System.dll*)) OR (winlog.event_id : ("11" OR "15") AND message : (*.dll* OR *.exe* OR *.pdb*)))

(f) Aggregate KQL Query:

event.provider : "Microsoft-Windows-Sysmon/Operational"

AND ((winlog.event_id : "1" AND message : (*InstallUtil* OR *RegAsm* OR *RegSvcs* OR *.NET* OR *dnx* OR *dotnet* OR *ngen* OR *MSBuild*)) OR (winlog.event_id : ("7" OR "8") AND message : (*mscorlib.dll* OR *mscorlib.dll* OR *System.dll* OR *clr.dll* OR *Sys-

```
tem.Reflection* )) OR (winlog.event_id : ("11" OR "15") AND message
: (*.dll* OR *.exe* OR *.pdb* OR *.resources*)) OR (winlog.event_id :
"13" AND message : *SOFTWARE\\Microsoft\\NETFramework*) )
```

- (g) The queries target various aspects of .NET binary manipulation and execution. They monitor for common .NET-related processes and utilities, DLL loading patterns specific to the .NET runtime, and file operations involving executables and debugging information. Registry modifications related to .NET Framework configuration are also tracked. The aggregate query combines these elements to provide visibility into both the obfuscated and deobfuscated stages of .NET binary manipulation.

64. RT @PO3T1985: #Reteefe dropper still loads the payload the same way only obfuscation changed from base64 to XOR hence the function was repâ

- (a) base64 (Deobfuscate/Decode Files or Information (T1140)):

```
event.provider : "Microsoft-Windows-Sysmon/Operational"
AND ( (event.provider : ("Microsoft-Windows-PowerShell*" OR "Win-
dows PowerShell") AND message : (*FromBase64* OR *base64* OR *Con-
vert* OR *encode*)) OR (winlog.event_id : ("11" OR "15") AND mes-
sage : (*base64* OR *.b64*)) OR (winlog.event_id : "1" AND message
: (*certutil* OR *CERTUTIL.EXE* OR *powershell* OR *POWER-
SHELL.EXE*)) )
```

- (b) XOR (Deobfuscate/Decode Files or Information (T1140)):

```
event.provider : "Microsoft-Windows-Sysmon/Operational"
AND ( (winlog.event_id : "1" AND message : (*powershell* OR *POW-
ERSHELL.EXE* OR *python* OR *PYTHON.EXE*)) OR
(winlog.event_id : ("7" OR "8") AND message : (*mscorlib.dll* OR *ker-
nel32.dll* OR *System.Security.Cryptography*)) OR (winlog.event_id :
```

("11" OR "15") AND message : (*.dat* OR *.bin* OR *.enc*)))

(c) Aggregate KQL Query:

```
( (event.provider : ("Microsoft-Windows-PowerShell*" OR "Windows PowerShell") AND message : (*FromBase64* OR *base64* OR *Convert* OR *encode* OR *xor*)) OR (event.provider : "Microsoft-Windows-Sysmon/Operational" AND ( (winlog.event_id : "1" AND message : ( *certutil* OR *CERTUTIL.EXE* OR *powershell* OR *POWERSHELL.EXE* OR *python* OR *PYTHON.EXE* )) OR (winlog.event_id : ("7" OR "8") AND message : ( *mscoree.dll* OR *kernel32.dll* OR System.Security.Cryptography* )) OR (winlog.event_id : ("11" OR "15") AND message : (*.dat* OR *.bin* OR *.enc* OR *.b64*)) ) ) )
```

(d) The queries monitor changes in Retefe's obfuscation techniques. They track both base64 encoding operations and potential XOR-based encryption through various indicators. PowerShell and Python processes that might handle these operations are monitored, along with relevant DLL loading and file operations. The aggregate query combines these elements to detect both the original base64 encoding and the new XOR-based obfuscation methods.

65. RT @CertPa: On Shodan there are 600+ #Lantronix Ethernet Adapters registered (in Italy) with free access via port 9999 thx to: @dalmoz_

(a) port 9999 (Non-Standard Port (T1571)):

```
event.provider:"Microsoft-Windows-Sysmon/Operational"
AND ( (winlog.event_id:"3" AND message:(*9999*)) OR
(winlog.event_id:"1" AND message:(*netsh* OR *NETSH.EXE* OR *netcat* OR *nc.exe* OR *NC.EXE*)) )
```

(b) Aggregate KQL Query:

event.provider:

"Microsoft-Windows-Sysmon/Operational"

AND ((winlog.event_id:"3" AND message:(*9999*)) OR

(winlog.event_id:"1" AND message:(*netsh* OR *NETSH.EXE* OR *netcat* OR *nc.exe* OR *NC.EXE*)))

- (c) The queries monitor for network activity on the non-standard port 9999 commonly used by Lantronix Ethernet Adapters. They track both direct connections to this port and the execution of network utilities that might be used to interact with these devices. The same approach serves both individual and aggregate detection since we're focusing on specific port-based activity.

66. PO doc email from Belize (94.177.123.114)

<https://t.co/1oKqrFbwz3> many StrReverse & 1 Shell

Shell (StrReverse(StrReverse(nVBj))) 0 powershell gets payload but url is down & no VT license for me :-)

[hxxp://coastmotorsupply\[.\]com/MicrosoftWordUpdate.exe](http://coastmotorsupply[.]com/MicrosoftWordUpdate.exe)

<https://t.co/115cA4ONpl>

<https://t.co/SWsDnPeBJq>

- (a) [hxxp://coastmotorsupply\[.\]com/MicrosoftWordUpdate.exe](http://coastmotorsupply[.]com/MicrosoftWordUpdate.exe)

(User Execution (T1204)):

event.provider : "Microsoft-Windows-Sysmon/Operational"

AND ((winlog.event_id : "3" AND message : (*coastmotorsupply* OR *MicrosoftWordUpdate.exe*)) OR (winlog.event_id : "1" AND message : (*winword.exe* OR *WINWORD.EXE* OR

WSSCRIPT.EXE* OR

WSSCRIPT.EXE*)) OR (winlog.event_id : ("11" OR "15") AND message

: (*MicrosoftWordUpdate.exe* OR *StrReverse*)))

(b) Aggregate KQL Query:

event.provider : "Microsoft-Windows-Sysmon/Operational"

AND ((winlog.event_id : "3" AND message : (*coastmotorsupply* OR *MicrosoftWordUpdate.exe*)) OR (winlog.event_id : "1" AND message :

(*winword.exe* OR *WINWORD.EXE* OR

WSCRIPT.EXE* OR

WSCRIPT.EXE*)) OR (winlog.event_id : ("11" OR "15") AND message

: (*MicrosoftWordUpdate.exe* OR *StrReverse*)))

(c) The queries monitor this malicious document's execution chain. They track network connections to the malware hosting domain, Word and script host process execution, and file operations involving both the malicious executable and VBScript functionality like StrReverse. The same comprehensive detection approach serves both individual and aggregate needs since we're tracking specific indicators across process execution, network, and file activity.

67. The next steps use the excellent PHP Sandbox at

<https://t.co/19KLsD1Y5V>. The script is heavily obfuscated redundant loops and lots of string manipulation. The strings look like base64 though and one of the first functions translates to 'base64_decode' so where is it used?

<https://t.co/Wj5SuLIUW4>

(a) base64_decode (Obfuscated Files or Information (T1027)):

event.provider:"Microsoft-Windows-Sysmon/Operational"

AND ((event.provider:("Microsoft-Windows-PowerShell*" OR "Windows PowerShell") AND message:(*FromBase64* OR *base64* OR *Convert*

OR *encode*)) OR (winlog.event_id:"1" AND message:(*certutil* OR

```
*CERTUTIL.EXE* OR *powershell* OR *POWERSHELL.EXE*)) OR
(winlog.event_id:("11" OR "15") AND message:(*base64* OR *.b64*)) )
```

(b) Aggregate KQL Query:

```
event.provider:"Microsoft-Windows-Sysmon/Operational"
AND ( (event.provider:("Microsoft-Windows-PowerShell*" OR "Windows
PowerShell") AND message:(*FromBase64* OR *base64* OR *Convert*
OR *encode*)) OR (winlog.event_id:"1" AND message:(*certutil* OR
*CERTUTIL.EXE* OR *powershell* OR *POWERSHELL.EXE*)) OR
(winlog.event_id:("11" OR "15") AND message:(*base64* OR *.b64*)) )
```

(c) The queries focus on detecting base64 decoding operations that might indicate deobfuscation of malicious code. They monitor for PowerShell commands and utilities commonly used for base64 decoding, as well as file operations involving base64-encoded content. The same detection approach serves both individual and aggregate needs since we're focusing specifically on base64 decoding activities.

68. RT @hackerfantastic: RSA leaked all the attendees personal details via unsecured public facing API's using hard coded credentials in a mobia

(a) hard coded credentials (Unsecured Credentials (T1552)):

```
event.provider:"Microsoft-Windows-Sysmon/Operational"
AND ( (winlog.event_id:("11" OR "15") AND message:(*.config* OR
*.ini* OR *.conf* OR *.xml* OR *.json* OR *.properties*)) OR (win-
log.event_id:"13" AND message:(*SOFTWARE\\Microsoft\\Windows\\
CurrentVersion\\Run* OR *SOFTWARE\\Microsoft\\Windows NT\\
CurrentVersion\\Winlogon*)) OR (winlog.event_id:"1" AND
message:(*type* OR *findstr* OR *find* OR *strings* OR *grep.exe* OR
*GREP.EXE*)) )
```

(b) Aggregate KQL Query:

```
event.provider:"Microsoft-Windows-Sysmon/Operational"
AND ( ( winlog.event_id:("11" OR "15") AND message:(*.config* OR
*.ini* OR *.conf* OR *.xml* OR *.json* OR *.properties*)) OR (win-
log.event_id:"13" AND message:(*SOFTWARE\\Microsoft\\Windows\\
CurrentVersion\\Run* OR *SOFTWARE\\Microsoft\\Windows NT\\
CurrentVersion\\Winlogon*)) OR (winlog.event_id:"1" AND
message:(*type* OR *findstr* OR *find* OR *strings* OR *grep.exe* OR
*GREP.EXE*)) )
```

(c) The queries monitor for access to files and registry locations that commonly contain credentials. They track operations on configuration files, execution of text searching utilities that might be used to discover credentials, and registry locations known to store authentication information. The same detection approach serves both individual and aggregate needs since we're focusing specifically on potential credential exposure through common storage locations.

69. RT @TheHackersNews: Hackers are exploiting #Drupal RCE vulnerability (CVE-2018-7600) in the wild to backdoor and infect websites with #crypâ

(a) CVE-2018-7600 (Exploit Public-Facing Application (T1190)):

```
event.provider:"Microsoft-Windows-Sysmon/Operational"
AND ( ( winlog.event_id:"3" AND message:(*drupal* OR */user/register*
OR */admin* OR */?q=*)) OR (winlog.event_id:"1" AND
message:(*php.exe* OR *PHP.EXE* OR *w3wp.exe* OR *W3WP.EXE*
OR *httpd.exe* OR *HTTPD.EXE*)) OR (winlog.event_id:("11" OR
"15") AND message:(*.php* OR *index.php* OR *settings.php*)) )
```

(b) Aggregate KQL Query:

```
event.provider:"Microsoft-Windows-Sysmon/Operational"
AND ( (winlog.event_id:"3" AND message:(*drupal* OR */user/register*
OR */admin* OR */?q=*)) OR (winlog.event_id:"1" AND
message:(*php.exe* OR *PHP.EXE* OR *w3wp.exe* OR *W3WP.EXE*
OR *httpd.exe* OR *HTTPD.EXE*)) OR (winlog.event_id:("11" OR
"15") AND message:(*.php* OR *index.php* OR *settings.php*)) )
```

- (c) The queries focus on detecting exploitation attempts of the Drupal vulnerability CVE-2018-7600. They monitor network connections to common Drupal paths that might be targeted, web server process execution, and PHP file operations that could indicate exploitation or backdoor deployment. The same comprehensive detection approach serves both individual and aggregate needs since we're tracking specific indicators of Drupal exploitation attempts.

70. The dns.txt is a certificate file which than us decoded with certutil

- (a) dns.txt (Deobfuscate/Decode Files or Information (T1140)):

```
event.provider:"Microsoft-Windows-Sysmon/Operational"
AND ( (winlog.event_id:("11" OR "15") AND message:(*dns.txt* OR
*.cer* OR *.crt*)) OR
(winlog.event_id:"1" AND message:(*type* OR *more* OR *findstr* OR
*find*)) )
```

- (b) certutil (Deobfuscate/Decode Files or Information (T1140)):

```
event.provider:"Microsoft-Windows-Sysmon/Operational"
AND ( (winlog.event_id:"1" AND
message:(*certutil* OR *CERTUTIL.EXE*)) OR (winlog.event_id:"7"
AND message:(*crypt32.dll* OR *cryptnet.dll*)) OR
(winlog.event_id:"3" AND message:*decode*) )
```

- (c) Aggregate KQL Query:

```
event.provider:"Microsoft-Windows-Sysmon/Operational"
AND ( (winlog.event_id:"1" AND
message>(*certutil* OR *CERTUTIL.EXE* OR *type* OR *more* OR
*findstr* OR *find*)) OR (winlog.event_id:("11" OR "15") AND mes-
sage>(*dns.txt* OR *.cer* OR *.crt*)) OR
(winlog.event_id:"7" AND message(*crypt32.dll* OR *cryptnet.dll*)) OR
(winlog.event_id:"3" AND message:*decode*) )
```

- (d) The queries monitor abuse of certutil for decoding malicious content. They track operations involving certificate files and text files that might contain encoded data, along with certutil execution and related DLL loading. Network connections with "decode" in the command are also monitored. The aggregate query combines these elements to detect both the file handling and decoding phases of this technique.

71. Word.Exploit.CVE-2017-11882 and payload from /efficientmarketing.co

- (a) Word.Exploit.CVE-2017-11882

(Exploitation for Client Execution (T1203)):

```
event.provider:"Microsoft-Windows-Sysmon/Operational"
AND ( (winlog.event_id:"1" AND message(*eqnedt32.exe* OR
EQNEDT32.EXE* OR *winword.exe* OR *WINWORD.EXE*)) OR (win-
log.event_id:("7" OR "8") AND message(*equation.3* OR
EQNEDT32.EXE*)) OR (winlog.event_id:("11" OR "15") AND
message(*.doc* OR *.docx* OR *.rtf*)) )
```

- (b) efficientmarketing.co (Exploitation for Client Execution (T1203)):

```
event.provider:"Microsoft-Windows-Sysmon/Operational"
AND ( (winlog.event_id:"3" AND message:*efficientmarketing.co*) OR
```

```
(winlog.event_id:"1"
AND message>(*winword.exe* OR *WINWORD.EXE*)) OR
(winlog.event_id:("11" OR "15") AND message(*.doc* OR *.docx* OR
*.rtf*)) )
```

(c) Aggregate KQL Query:

```
event.provider:"Microsoft-Windows-Sysmon/Operational"
AND ( (winlog.event_id:"1" AND message(*eqnedt32.exe* OR
EQNEDT32.EXE* OR *winword.exe* OR *WINWORD.EXE*)) OR (win-
log.event_id:("7" OR "8") AND message(*equation.3* OR
EQNEDT32.EXE*)) OR (winlog.event_id:("11" OR "15") AND
message(*.doc* OR *.docx* OR *.rtf*)) OR (winlog.event_id:"3" AND
message:*efficientmarketing.co*) )
```

- (d) The queries monitor both the CVE-2017-11882 exploitation attempt and subsequent payload delivery. They track Equation Editor and Word process execution, related DLL loading patterns, document file operations, and connections to the malicious domain. The aggregate query combines these elements to provide visibility across both the initial exploitation and payload retrieval phases of the attack.