BLACK CELL
Protecting critical infrastructures

Threat Hunting Methodology

Document history			
Version	**Date**	**Short description**	**Author**
V1.0	22 August 2023	Initial.	Balázs Polena

Table of Contents

1 Introduction

1.1 Purpose of this Document

Welcome to the comprehensive guide on Threat Hunting Methodology. The primary purpose of this document is to equip individuals with basic IT knowledge with the essential skills and knowledge required to become proficient threat hunters. In today's rapidly evolving cybersecurity landscape, traditional security measures are not enough to safeguard organizations against advanced threats. Threat hunting plays a critical role in proactively identifying and mitigating potential cyber threats before they escalate into full-blown attacks. This document aims to demystify the concept of threat hunting, providing clear and practical guidance on its principles, methodologies, and best practices. Whether you are an IT professional, security analyst, or an aspiring cybersecurity enthusiast, this guide will serve as your foundation to excel in threat hunting activities.

1.2 Prerequisites

Before diving into the realm of threat hunting, there are a few prerequisites that will enhance your learning experience:

- Basic IT Knowledge: Familiarity with networking concepts, computer systems, and security fundamentals will help you grasp the threat hunting concepts more effectively.
- Cybersecurity Mindset: An understanding of the importance of cybersecurity and the motivations behind cyber threats will be valuable throughout this documentation.
- Willingness to Learn: Threat hunting is a dynamic and ever-evolving field. A willingness to stay curious, explore new techniques, and learn from challenges is important for success.
- Access to Tools: While we will discuss various threat hunting tools, having access to a threat intelligence feed, a SIEM (Security Information and Event Management) solution, and an Endpoint Detection and Response (EDR) system will enhance your hands-on learning experience.

With these prerequisites in place, you are ready to embark on an exciting journey into the world of threat hunting. The knowledge and skills gained from this documentation will not only strengthen your organization's security posture but also enrich your overall understanding of cybersecurity.

2 Understanding Threat Hunting

2.1 What is Threat Hunting?

Threat hunting is a proactive and iterative cybersecurity approach aimed at identifying and mitigating advanced threats that may have bypassed traditional security defenses. Unlike traditional cybersecurity practices that primarily rely on reactive measures, threat hunting involves actively searching for potential indicators of compromise (IoCs) and advanced adversaries within an organization's network. Threat hunters leverage various data sources, security tools, and threat intelligence to identify subtle signs of malicious activities that may have gone undetected by automated security systems. By adopting the mindset of an attacker, threat hunters seek to uncover hidden threats and better understand an adversary's tactics, techniques, and procedures (TTPs).

2.2 Why is Threat Hunting Important?

Threat hunting is a key component of a comprehensive cybersecurity strategy for several reasons:

- Proactive Defense: Threat hunting allows organizations to take a proactive stance against cyber threats. Instead of waiting for alerts or incidents, threat hunters actively seek out potential threats, minimizing the dwell time of attackers within the network.
- Identifying Unknown Threats: Advanced threats and sophisticated attackers often use novel techniques that may not be covered by traditional signature-based security tools. Threat hunting helps in identifying previously unknown or zero-day threats.
- Deep Visibility: Threat hunting provides deeper visibility into the network, systems, and applications. This enhanced visibility enables organizations to detect and investigate threats that might otherwise remain hidden.
- Understanding Adversary Behavior: By studying an adversary's TTPs, threat hunters can gain valuable insights into their motives, intentions, and potential targets. This understanding helps in developing more robust defenses.
- Reducing Detection Time: The proactive nature of threat hunting allows organizations to detect and respond to threats more quickly, reducing the time an attacker has to carry out their malicious activities.
- Continuous Improvement: Threat hunting is an iterative process that involves learning from past incidents and applying that knowledge to improve security defenses continually.
- Complementing Existing Security Measures: Threat hunting complements existing security measures, such as firewalls, antivirus software, and intrusion detection systems (IDS), by providing an additional layer of defense.

Incorporating threat hunting into an organization's cybersecurity strategy enhances the overall resilience against cyber threats, making it an indispensable practice in today's threat landscape.

2.3 Key Concepts in Threat Hunting

Before diving into the practical aspects of threat hunting, it's necessary to understand some key concepts that form the foundation of this proactive approach:

Indicators of Compromise (IoCs)

Indicators of Compromise (IoCs) are traces or artifacts left behind by cyber threats during their malicious activities. These indicators can be IP addresses, domain names, file hashes, URLs, or patterns in network traffic that are associated with known malicious activities. Threat hunters actively search for IoCs to identify potentially compromised systems and mitigate threats.

Tactics, Techniques, and Procedures (TTPs)

Tactics, Techniques, and Procedures (TTPs) refer to the methods and tools used by attackers to achieve their objectives. Understanding an adversary's TTPs is crucial for threat hunters as it enables them to predict the next moves of attackers and improve their defenses accordingly.

Adversary Behavior Profiling

Adversary Behavior Profiling involves studying the behavior patterns of known threat actors and understanding how they operate. By profiling adversaries, threat hunters can anticipate their actions and develop specific hunting strategies to detect their presence more effectively.

In the subsequent sections, we will explore the methodologies, data analysis techniques, and practical aspects of conducting threat hunting campaigns.

3 Preparing for Threat Hunting

3.1 Defining Objectives and Scope

The objectives should align with the organization's overall cybersecurity strategy and may include:

- Identifying Advanced Threats: The primary objective of threat hunting is to proactively identify advanced threats that have evaded traditional security measures.
- Enhancing Security Posture: By gaining deeper insights into an organization's security gaps and potential weaknesses, threat hunting can lead to more robust security measures.
- Understanding Adversary Behavior: Profiling adversary behavior helps in understanding their motives, tactics, and potential targets, thereby enabling more targeted defense strategies.

The scope of the threat hunting campaign should consider the following factors:

- Timeframe: Determine the duration of the hunting campaign, whether it's a continuous process or a time-bound exercise.
- Network Segments: Decide which network segments or systems will be included in the hunting campaign.
- Data Sources: Identify the data sources (e.g., logs, traffic data) that will be used for analysis.
- Threat Intelligence: Determine if threat intelligence feeds will be used to enrich the hunting process.
- Adversary Types: Specify the types of adversaries or threat actors that will be focused on during the campaign.

3.2 Acquiring Necessary Tools

For an effective threat hunting campaign, having the right tools and technologies is vital. Some essential tools include:

SIEM (Security Information and Event Management)

A SIEM system aggregates and analyzes log data from various sources across the network. It provides a centralized platform for threat hunters to monitor, correlate, and investigate security events.

Endpoint Detection and Response (EDR) Tools

EDR tools are deployed on endpoints (e.g., workstations, servers) to monitor and record activities at the endpoint level. They provide detailed visibility into endpoint behavior, aiding threat hunters in detecting malicious activities. Extending the endpoint dataset or enriching the telemetry is very welcomed approach with tools like Sysmon or Auditd with a proper configuration file.

Threat Intelligence Feeds

Threat intelligence feeds deliver real-time information about known threat actors, their TTPs, and emerging threats. Integrating threat intelligence into the hunting process can help identify indicators of known threats.

Internal Network Traffic

Internal mirrored traffic is a crucial asset for threat hunting, and the role of Zeek in extracting metadata from this traffic cannot be overstated. Zeek, formerly known as Bro, plays a pivotal role by dissecting network packets and extracting rich metadata, including protocol details, connection information, and file transfers. This metadata provides threat hunters with invaluable insights, helping them profile normal network behavior and quickly spot deviations that could signify malicious activity.

3.3 Assembling a Threat Hunting Team

A threat hunting team should consist of skilled and knowledgeable professionals who can work collaboratively to identify and respond to threats effectively. The team may include:

- Threat Hunters: Experienced analysts with in-depth knowledge of cybersecurity, incident response, and threat hunting methodologies.
- Security Analysts: Experts in analyzing security events and conducting incident response activities.
- Threat Intelligence Analysts: Individuals who specialize in gathering and analyzing threat intelligence to aid the hunting process.
- Data Engineers: Professionals with expertise in analyzing large datasets to identify patterns and anomalies.

In the next sections, we will delve into data collection and analysis techniques, methodologies for conducting threat hunting campaigns, and leveraging threat intelligence in the hunting process.

4 Data Collection and Analysis

4.1 Data Sources for Threat Hunting

To conduct successful threat hunting, you need access to relevant and comprehensive data sources. These sources provide valuable insights into network activity, system behavior, and potential security incidents. Some key data sources include:

Network Traffic Logs

Network traffic logs capture information about communication between devices within the network. These logs include data such as source and destination IP addresses, protocols used, port numbers, and packet size. Analyzing network traffic logs can reveal unusual or suspicious patterns indicative of malicious activities.

Endpoint Logs

Endpoint logs record events and activities on individual devices, such as workstations, servers, and mobile devices. These logs contain valuable information about file access, process execution, user logins, and system changes. Endpoint logs play a key role in detecting and investigating threats on specific devices.

Firewall and Proxy Logs

Firewall and proxy logs provide details about traffic entering and leaving the network, as well as web browsing activities. Analyzing these logs helps identify unauthorized access attempts, potential data exfiltration, and suspicious connections to known malicious domains.

4.2 Data Analysis Techniques

Effectively analyzing the collected data is a critical aspect of threat hunting. Various data analysis techniques can aid in identifying potential threats and anomalies:

Signature-based Analysis

Signature-based analysis involves comparing collected data against known patterns or signatures of known threats. Security analysts use signature databases or threat intelligence feeds to identify indicators of known malware, malicious IP addresses, or file hashes associated with known malware families.

Anomaly-based Analysis

Anomaly-based analysis focuses on identifying activities or behaviors that deviate from normal patterns. By establishing a baseline of normal behavior, any deviations or anomalies can be flagged as potential security incidents. Anomaly detection can help identify new or unknown threats that do not match any predefined signatures.

Behavioral Analysis

Behavioral analysis involves monitoring and profiling the behavior of users, systems, and applications over time. This analysis aims to identify suspicious or malicious actions based on typical adversary behavior. Behavioral analytics can help detect insider threats, lateral movement, and other advanced attack techniques.

4.3 Leveraging Data Visualization

Visualizing data allows threat hunters to identify patterns, outliers, and potential correlations that might not be evident in raw logs or datasets. Tools such as graphs, charts, and heatmaps can provide intuitive representations of complex data, facilitating the identification of unusual activities or trends. Here are some ways data visualization can be leveraged in threat hunting:

- **Identifying Patterns and Trends**: Visualization tools can display data in graphs, charts, or heatmaps, helping analysts identify patterns, trends, and recurring behaviors that might indicate malicious activities. For example, visualizing login activity can reveal unusual login times or multiple failed login attempts, indicating potential brute-force attacks.
- **Behavioral Analysis**: Visualization can be used to understand normal user behavior and detect deviations from the norm. Anomalies in user activity, network traffic, or data access can be quickly identified through visual representations, raising red flags for further investigation.
- **Network Traffic Analysis**: Visualizing network traffic can help detect unusual data flows or communication patterns that may indicate the presence of malware, command-and-control (C2) servers, or data exfiltration attempts.
- **Geospatial Visualization**: Mapping IP addresses or network traffic geographically can help identify unauthorized access from unusual locations or regions, indicating possible intrusion attempts.
- **Timeline Analysis**: By representing events along a timeline, analysts can trace the sequence of events leading up to a potential security incident. This chronological view can be invaluable in reconstructing an attack's progression.

Source:

- Visual Link Analysis with Splunk

5 Types of Threat Hunting

Structured hunting and unstructured hunting are two distinct approaches used in threat hunting within cybersecurity.

5.1 Structured Hunting

Structured hunting is a proactive cybersecurity practice that follows a systematic and hypothesis-driven approach. In this method, threat hunters start by formulating hypotheses or educated guesses about potential security threats or malicious activities based on available information and threat intelligence. These hypotheses are like investigative theories that guide the hunting process. The hypotheses could be centered around specific threat indicators, patterns, or behaviors that attackers might exhibit. Once the hypotheses are formulated, the hunting activity is scoped, meaning that the focus is narrowed down to specific areas or systems within the organization's network. Then, the threat hunters conduct in-depth investigations to validate or refute these hypotheses. They analyze network logs, system data, and other relevant information to find evidence of suspicious behavior or indicators of compromise.

5.2 Unstructured Hunting

Unstructured hunting, on the other hand, takes a more flexible and data-driven approach. In this method, threat hunters explore available data, logs, and other information without starting with specific hypotheses. Instead of following a predefined path, they allow the data to guide their investigation. The goal of unstructured hunting is to identify anomalies or patterns that may indicate potential threats or malicious activities. Threat hunters may come across suspicious behavior or indicators they weren't initially looking for, leading them to pivot their investigation in response to what they find. Unstructured hunting is particularly useful for discovering previously unknown threats or detecting uncommon attack patterns. It enables threat hunters to adapt to evolving attack techniques and behaviors. However, it may require more time and resources compared to structured hunting because it involves a more exploratory and less directed approach.

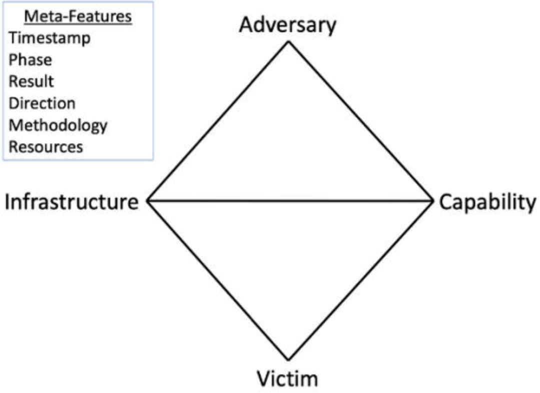
In practice, organizations can benefit from both structured and unstructured hunting. Structured hunting allows for targeted investigations and validates specific hypotheses, while unstructured hunting provides opportunities for discovering novel threats and staying ahead of sophisticated attackers. Combining these two approaches can create a comprehensive threat hunting strategy that enhances an organization's overall cybersecurity posture.

6 Threat Hunting Frameworks and Models

To effectively identify and mitigate potential security breaches, security professionals rely on frameworks and models that provide structured approaches to understanding adversary behaviors and attack patterns. In this context, various threat hunting frameworks and models play a pivotal role. These frameworks, such as the Diamond Model of Intrusion Analysis, the Cyber Kill Chain Model, and the MITRE ATT&CK Framework, offer systematic methodologies to dissect and analyze cyber threats. This introduction delves into the significance of these frameworks and models within the realm of threat hunting, highlighting their roles in enhancing an organization's ability to detect, respond to, and defend against emerging security risks.

6.1 The Diamond Model of Intrusion Analysis

The Diamond Model of Intrusion Analysis is a framework that helps threat hunters organize and analyze cyber threats. It provides a structured approach for dissecting and visualizing the key components of a cyber-attack, helping security analysts make sense of complex incidents. The model consists of four core elements represented as a diamond: Victim, Adversary, Infrastructure, and Capability.



Diamond Model of Intrusion Analysis (source)

The "**Adversary**" component represents the threat actor or group responsible for the intrusion. Understanding the motives, capabilities, and TTPs of the adversary is crucial in predicting their behavior and implementing effective defenses.

The "**Victim**" component identifies the target of the attack, typically an organization or individual. Analyzing the victim's environment, assets, and vulnerabilities provides insights into why the adversary chose the target and what they aim to achieve.

The "**Infrastructure**" component encompasses the tools, servers, and network infrastructure used by the adversary to conduct the attack. Analyzing the infrastructure helps in identifying patterns and commonalities across multiple attacks, enabling more proactive defense measures.

The "**Capability**" component represents the technical capabilities of the adversary, such as their expertise in malware development, exploitation techniques, and evasion tactics. Understanding the adversary's capabilities aids in predicting their potential future activities and crafting effective countermeasures.

6.2 Cyber Kill Chain Model

The Cyber Kill Chain Model, developed by Lockheed Martin, outlines the stages of a typical cyber-attack. Understanding each stage of the kill chain helps threat hunters identify and disrupt the attack before it reaches the final objective. The stages of the Cyber Kill Chain are as follows:

1. **Reconnaissance**: In this initial stage, the adversary gathers information about the target, such as scanning for open ports and identifying potential vulnerabilities.
2. **Weaponization**: The adversary creates or acquires a weapon (e.g., malware) designed to exploit the identified vulnerabilities.
3. **Delivery**: The weaponized payload is delivered to the victim's system, often through methods like phishing emails or malicious websites.
4. **Exploitation**: The weapon is used to exploit the identified vulnerabilities in the victim's system, gaining access and control.
5. **Installation**: Once inside the system, the adversary installs persistent access points and establishes a foothold.
6. **Command and Control (C2)**: The adversary sets up communication channels to maintain control over the compromised systems.
7. **Actions on Objectives**: Finally, the adversary carries out their intended actions, which could include data exfiltration, financial fraud, or other malicious activities.

6.3 MITRE ATT&CK Framework

The MITRE ATT&CK (Adversarial Tactics, Techniques, and Common Knowledge) Framework is a comprehensive knowledge base that documents the actions and techniques used by adversaries across different stages of an attack. It categorizes adversary behaviors into tactics and techniques, providing a standardized way to describe cyber threats. The ATT&CK Framework consists of several matrices, each focusing on a specific platform (e.g., Windows, Linux) or environment (e.g., cloud). Threat hunters use this framework to map observed behaviors to specific tactics and techniques, enabling them to gain a deeper understanding of an adversary's actions and anticipate their next moves.

By employing these frameworks and models into threat hunting methodologies, organizations can streamline their hunting efforts and gain deeper insights into potential threats. In the following section, we will explore how to identify indicators of compromise (IoCs) and understand adversary tactics, techniques, and procedures (TTPs) to enhance the effectiveness of threat hunting campaigns.

7 Identifying Indicators of Compromise (IoCs)

Indicators of Compromise (IoCs) are valuable artifacts left behind by cyber threats during their activities. Threat hunters actively search for IoCs in network and endpoint data to identify potential security breaches and compromised systems. In this section, we will explore common types of IoCs and techniques to extract and analyze them:

7.1 Common Types of IoCs

- **IP Addresses**: Malicious IP addresses are known sources or destinations of cyber threats. They can be used by attackers as command and control (C2) servers, hosting malware, or as part of a phishing campaign. Monitoring and analyzing network traffic to detect communications with suspicious IP addresses is indispensable for threat hunting.
- **Domain Names**: Malicious domain names are used in various attack vectors, such as phishing, malware distribution, and C2 communications. Threat hunters can analyze DNS logs and web proxy data to uncover potentially malicious domains, like DGAs.
- **File Hashes**: File hashes, such as MD5, SHA-1, or SHA-256, represent a unique fingerprint of a file's content. Threat hunters can use threat intelligence feeds to compare file hashes against known malicious files and identify potentially harmful executables or documents.
- **URLs and URIs**: Malicious URLs and Uniform Resource Identifiers (URIs) often lead to phishing websites or distribute malware. Analyzing web proxy logs and email data can help identify suspicious URLs.
- **Registry Keys and File Paths**: Certain registry keys and file paths are commonly targeted by malware for persistence and execution. Identifying unusual or unauthorized modifications to registry keys and file paths can signal potential compromise.

7.2 Extracting IoCs from Threat Intelligence Feeds

Threat intelligence feeds provide valuable information about known threats, including IoCs associated with various adversaries and malware families. Threat hunters can integrate threat intelligence feeds into their SIEM or other analysis tools to enrich their data with up-to-date IoCs. Automated integration allows for real-time correlation of collected data against threat intelligence feeds, enabling the rapid detection of potential threats as soon as they are identified in the wild.

Source:

- https://github.com/hslatman/awesome-threat-intelligence

7.3 Analyzing IoCs to Identify Potential Threats

Once IoCs are collected and extracted from various sources, threat hunters must analyze them to identify potential threats within the organization's network. This process involves:

- **Correlation**: Threat hunters correlate collected IoCs with logs from different data sources to identify related events or activities. For example, a network connection to a known malicious IP address combined with attempts to modify critical system files might indicate a compromised system.

- **Contextualization**: Contextualizing IoCs involves understanding the significance of each IOC in the context of the organization's environment. For example, an IOC that appears benign in one organization might be highly suspicious in another.
- **Investigation**: Investigating potential threats involves in-depth analysis and research. Threat hunters must determine the extent of the potential compromise, the adversary's TTPs, and the impact on the organization's security posture.

By diligently identifying and analyzing IoCs, threat hunters can swiftly respond to potential threats, limit the impact of security incidents, and continuously improve their understanding of adversary behavior. In the next section, we will explore common adversary tactics, techniques, and procedures (TTPs) and how they can aid in successful threat hunting campaigns.

8 Understanding Tactics, Techniques, and Procedures (TTPs)

Tactics, Techniques, and Procedures (TTPs) represent the methods and procedures used by adversaries to achieve their objectives during cyber-attacks. Understanding common adversary TTPs is essential for threat hunters as it helps them anticipate and detect potential threats more effectively.

8.1 Common TTPs Used by Attackers

Phishing Attacks

Phishing is a prevalent tactic used by attackers to deceive users into revealing sensitive information or downloading malware. Threat hunters must be vigilant in monitoring email logs and user behavior to detect phishing attempts.

Credential Theft

Attackers often attempt to steal user credentials to gain unauthorized access to systems and applications. Threat hunters should closely monitor login activity and analyze authentication logs for any suspicious or unusual behavior.

Malware Delivery

Malware delivery involves distributing malicious software to target systems. Threat hunters should look for indicators of malware delivery, such as suspicious file attachments, malicious URLs, or compromised websites.

Lateral Movement

After gaining initial access, attackers move laterally through the network to explore and compromise additional systems. Threat hunters should be vigilant in monitoring lateral movement activities, such as unusual file access or remote execution attempts.

Data Exfiltration

Data exfiltration is the unauthorized transfer of sensitive data from the organization's network to external locations controlled by the attacker. Detecting abnormal data transfers and analyzing network egress traffic can help identify potential data exfiltration attempts.

8.2 Mapping TTPs to the MITRE ATT&CK Framework

The MITRE ATT&CK Framework is a valuable resource for threat hunters as it categorizes adversary behavior into specific tactics and techniques. By mapping observed TTPs to the ATT&CK Framework, threat hunters gain a structured understanding of an attacker's actions and can identify patterns and trends across various attacks. The ATT&CK Framework includes tactics such as Initial Access, Execution, Persistence, Privilege Escalation, Defense Evasion, Credential Access, Discovery, Lateral Movement, Collection, Exfiltration, and Impact. Within each tactic, there are numerous associated techniques that provide deeper insights into specific attack methods. Further information can be found on https://attack.mitre.org.

9 Threat Hunting Methodologies

9.1 TaHiTI

One methodology that aims to streamline and enhance the process of threat hunting is TaHiTI, which stands for "Targeted Hunting integrating Threat Intelligence." Developed as a collaborative effort among professionals from the Dutch financial sector, the TaHiTI methodology presents a structured and comprehensive approach to conducting effective threat hunting while harnessing the power of threat intelligence. The TaHiTI methodology unfolds across three distinct phases, each designed to address specific aspects of the threat hunting process. These phases not only facilitate the organized execution of threat hunting but also emphasize the integration of threat intelligence to enhance the depth and accuracy of investigations.

Phase 1: Initiate

The Initiate phase of the TaHiTI methodology serves as the catalyst for effective threat hunting endeavors. At its core, this phase revolves around recognizing triggers that prompt the initiation of a hunting investigation. These triggers can originate from various sources, including threat intelligence reports, observed anomalies in security data, or insights gained from incident response activities. By identifying these triggers, security teams can ensure that their threat hunting efforts are aligned with the most relevant and pressing concerns.

Once a trigger is identified, it is transformed into an abstract—a concise overview of the potential threat or concern. This abstract captures the essence of the investigation to be conducted and serves as a foundational piece of information for the subsequent phases. It provides a clear and concise understanding of what the hunting investigation aims to achieve, enabling the hunting team to stay focused on its objectives.

The abstract is then stored in a designated hunting backlog, which acts as a repository for all pending and potential hunting investigations. This backlog facilitates organization, collaboration, and prioritization of investigations. It serves as a central hub where security professionals can access and evaluate different abstracts, selecting the most relevant ones for further investigation.

Phase 2: Hunt

The Hunt phase represents the heart of the TaHiTI methodology, where the actual investigative work takes place. This phase consists of two primary activities: "define / refine" and "execute."

In the "define / refine" activity, the abstract from the Initiate phase is expanded upon, refined, and solidified into a full-fledged investigation plan. This involves adding necessary details such as the specific data sources to be analyzed, the data analysis techniques to be employed, and the overarching hypothesis that will guide the investigation. The hypothesis is a critical component as it outlines the suspected threat, attack vector, or anomaly that the hunting team aims to uncover. A well-defined hypothesis provides direction and structure, ensuring that the investigation is focused and purposeful.

With the investigation plan in place, the "execute" activity begins. This is where the actual data collection, analysis, and hunting take place. Security professionals utilize various data analysis techniques, which can range from simple querying to more complex statistical analyses. Data sources that are relevant to the hypothesis are examined, anomalies are identified, and potential signs of malicious activity are scrutinized.

Throughout this phase, the integration of threat intelligence becomes evident. Threat intelligence provides context and insights that can aid in understanding the nature of the threat, the methods employed by attackers, and the potential impact on the organization. Threat intelligence resources, such as the MITRE ATT&CK framework, can help guide the investigation and provide additional context to the findings.

The last thing we do in the hunt phase is to check if our initial guess (hypothesis) was right. When we finish investigating, we see if what we suspected, like malicious activity, actually happened. There are three possible outcomes:

- we confirm our guess was right (we found something bad and start dealing with it),
- we realize our guess was wrong (no bad stuff found),
- or we're not sure because the evidence isn't clear. If we're not sure, we can go back to the start and think about our investigation plan again.

We might change things like what we're looking for or how we're looking. This helps us make our investigation better. Sometimes, we might not have all the information we need to be sure. In that case, we can consider the investigation as not successful, but even from this, we can learn important things to make future investigations better.

Phase 3: Finalize

The Finalize phase marks the culmination of the TaHiTI methodology and focuses on documenting and disseminating the results of the threat hunting investigation.

During this phase, the findings, conclusions, and insights gained from the investigation are meticulously documented. This documentation includes not only the specific findings of potential threats or anomalies but also the analysis techniques employed and the rationale behind the conclusions drawn. Recommendations are also an integral part of the documentation, suggesting actions and improvements that the organization can undertake to enhance its security posture.

The documented results are not isolated; they are intended to be shared and integrated with other relevant security processes. This phase emphasizes the collaborative nature of cybersecurity, as the results of threat hunting investigations have implications for incident response, security monitoring, threat intelligence, vulnerability management, and other areas. By effectively communicating findings and recommendations to these processes, organizations can ensure that the insights gained from threat hunting are leveraged to enhance overall security effectiveness.

In conclusion, the TaHiTI methodology's three phases—Initiate, Hunt, and Finalize—provide a comprehensive and systematic approach to conducting effective threat hunting. By starting with triggers, focusing on well-defined hypotheses, leveraging threat intelligence, and documenting results for integration with other processes, organizations can proactively detect and mitigate threats, thereby strengthening their cybersecurity posture.

Source:

- TaHiTI

9.2 PEAK Framework

PEAK Framework is a modernized approach to threat hunting. The name stands for "Prepare, Execute, and Act with Knowledge," and it introduces a fresh perspective on how to conduct threat hunting effectively. It incorporates three main types of hunts: Hypothesis-Driven, Baseline, and Model-Assisted Threat Hunts (M-ATH).

- **Hypothesis-Driven Hunts:** In this approach, hunters formulate hypotheses about potential threats and their activities within the organization's network. They then use data and analysis to validate or debunk these hypotheses. (source)
- **Baseline Hunts:** Here, hunters establish a baseline of "normal" behavior within the organization's systems and networks. They then look for deviations from this baseline that could indicate malicious activity. (source)

- **Model-Assisted Threat Hunts (M-ATH)**: This approach combines human expertise with machine learning techniques. Hunters create models of both known good and known malicious behavior using machine learning. These models are then used to identify activity that aligns with or deviates from them. (source)

The PEAK process consists of three stages: Prepare, Execute, and Act. In the Prepare phase, hunters plan their hunts, gather information, and conduct research. The Execute phase involves deep data analysis, and the Act phase focuses on documentation, communication, and automation. Throughout the entire process, knowledge plays a key role. This knowledge can be derived from organizational expertise, threat intelligence, prior hunting experience, or the findings of the current hunt. The PEAK Framework offers flexibility, allowing hunters to tailor their approach to suit the specific situation they are dealing with. This adaptability enables security teams to respond effectively to different threats.

In summary, the PEAK Framework represents a holistic and innovative approach to threat hunting that incorporates different types of hunts and integrates knowledge, human expertise, and machine learning techniques to enhance the efficiency and effectiveness of threat detection and response.

Source:

- Introducing the PEAK Threat Hunting Framework | Splunk
- Splunk - David Bianco blog

9.3 Query-Based Threat Hunting

Query-Based Threat Hunting is a proactive cybersecurity approach that involves searching through large volumes of data to identify potential threats, anomalies, or suspicious activities within an organization's network, systems, or applications. Unlike traditional reactive methods that rely on predefined signatures, query-based hunting empowers security teams to formulate custom queries that target specific indicators of compromise or patterns associated with advanced threats.

This method leverages the power of data analysis and correlation to discover threats that might otherwise go unnoticed. By crafting and executing queries, security analysts can detect emerging threats, insider attacks, lateral movement, data exfiltration, and other malicious behaviors that might bypass traditional security measures.

Crafting effective queries requires a deep understanding of the data sources, the potential threats, and the underlying query language. The syntax varies depending on the data source being queried (e.g., databases, log files, network traffic), and it's important to master the query language specific to each source.

9.3.1 Kusto Query Language (KQL)

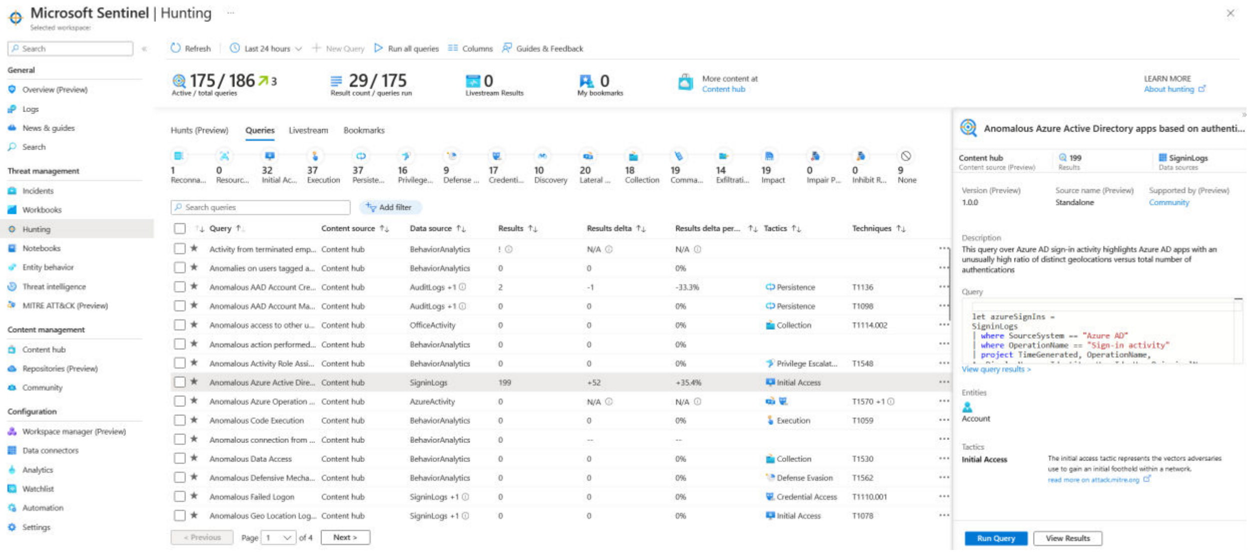
One powerful tool often used for threat hunting is Kusto Query Language (KQL), which is used with Microsoft's Azure Sentinel and Microsoft 365 Defender’s Advanced Hunting module. KQL allows you to query and analyze large datasets to uncover hidden threats and anomalies. Here are the key components of KQL that are commonly used in threat hunting:

1. **Data Source Selection**: Before you start writing KQL queries, you need to decide which data sources you want to analyze. This could include log files, network traffic, system events, and more. Different data sources will require different queries to extract relevant information.
2. **Basic Query Structure**: KQL queries follow a basic structure:
   ‘*datatable* | *where Condition* | *project Columns*’.
   - ‘*datatable*’: Specifies the table or dataset you're querying.
   - ‘*where Condition’*: Specifies the filtering condition to narrow down the data.
   - ‘*project*’ *Columns*’: Specifies the columns you want to include in the query result.
3. **Time Windows**: Incorporating time windows is crucial in threat hunting. You often want to analyze data within a specific timeframe to identify patterns or anomalies. Use the ‘*timestamp’* field to filter data based on time.
4. **Aggregation and Grouping**: To identify trends or anomalies in data, you can aggregate and group data using functions like ‘*summarize*’, ‘*count*’, ‘*avg’*, etc. This can help you spot patterns that might indicate malicious activities.
5. **Joins**: If you're working with multiple datasets, you might need to perform joins to combine data from different tables or sources. Use the ‘*join’* operator to correlate data and uncover relationships.
6. **Subqueries**: Subqueries can be used to nest queries within queries, allowing you to perform complex operations or filtering based on the results of another query.
7. **Pattern Matching:** KQL supports pattern matching using regular expressions. This is useful for identifying specific strings or patterns within textual data, which can be indicative of threats.
8. **Statistical Analysis**: KQL provides statistical functions that can help you identify outliers or unusual behavior within your data. These functions include ‘*percentile’*, ‘*stdev’*, and ‘*bin’*.
9. **Thresholds and Anomalies**: Set threshold values for specific metrics or behaviors. You can then compare incoming data against these thresholds to identify anomalies or deviations from the expected norm.
10. **Visualization**: KQL results can be visualized using tools like Azure Monitor, Kibana, or other visualization platforms. Creating graphs, charts, and dashboards can help you better understand the data and spot trends.
11. **Alert Creation**: In a threat hunting context, you can use KQL to create custom alerts. These alerts can notify you when certain conditions are met, allowing you to respond to potential threats in real-time.

12. **Iterative Refinement**: Threat hunting is an iterative process. As you analyze data and uncover potential threats, you may need to adjust your queries and techniques to dive deeper into the investigation.

Microsoft Sentinel Hunts

Microsoft Sentinel has a built-in proactive security tool for identifying threats within an organization's data. It offers powerful search and query tools to analyze security data, aiding analysts in detecting anomalies and potential risks. The system includes pre-built hunting queries, with a dashboard showcasing tactics and techniques based on the MITRE ATT&CK framework. Analysts can use these queries before, during, and after security incidents to gain insights, monitor ongoing compromises, and prevent future attacks. Custom queries can be created, shared, and refined, while community resources like GitHub provide additional queries and data sources.



Query	Content source	Data source	Results	Results delta	Results delta per...	Tactics	Techniques
Activity from terminated emp...	Content hub	BehaviorAnalytics	!	N/A	N/A		
Anomalies on users tagged a...	Content hub	BehaviorAnalytics	0	0	0%		
Anomalous AAD Account Cre...	Content hub	AuditLogs +1	2	-1	-33.3%	Persistence	T1136
Anomalous AAD Account Ma...	Content hub	AuditLogs +1	0	0	0%	Persistence	T1098
Anomalous access to other u...	Content hub	OfficeActivity	0	0	0%	Collection	T1114.002
Anomalous action performed...	Content hub	BehaviorAnalytics	0	0	0%		
Anomalous Activity Role Assi...	Content hub	BehaviorAnalytics	0	0	0%	Privilege Escalat...	T1548
Anomalous Azure Active Dire...	Content hub	SigninLogs	199	+52	+35.4%	Initial Access	
Anomalous Azure Operation ...	Content hub	AzureActivity	0	N/A	N/A		T1570 +1
Anomalous Code Execution	Content hub	BehaviorAnalytics	0	0	0%	Execution	T1059
Anomalous connection from ...	Content hub	BehaviorAnalytics	0	--	--		
Anomalous Data Access	Content hub	BehaviorAnalytics	0	0	0%	Collection	T1530
Anomalous Defensive Mecha...	Content hub	BehaviorAnalytics	0	0	0%	Defense Evasion	T1562
Anomalous Failed Logon	Content hub	SigninLogs +1	0	0	0%	Credential Access	T1110.001
Anomalous Geo Location Log...	Content hub	SigninLogs +1	0	0	0%	Initial Access	T1078

Anomalous Azure Active Directory apps based on authenti...

Content hub	199	SigninLogs
Content source (Preview)	Results	Data sources

Version (Preview)	Source name (Preview)	Supported by (Preview)
1.0.0	Standalone	Community

Description

This query over Azure AD sign-in activity highlights Azure AD apps with an unusually high ratio of distinct geolocations versus total number of authentications

Query

```
let azureSignIns =
SigninLogs
| where SourceSystem == "Azure AD"
| where OperationName == "Sign-in activity"
| project TimeGenerated, OperationName,
```

View query results >

Entities

Account

Tactics

Initial Access — The initial access tactic represents the vectors adversaries use to gain an initial foothold within a network. read more on attack.mitre.org

Source:

KQL

- Kusto Query Language (KQL) overview - Azure Data Explorer | Microsoft Learn
- SQL to Kusto query translation - Azure Data Explorer | Microsoft Learn
- Splunk to Kusto map - Azure Data Explorer | Microsoft Learn

Threat Hunting:

- Hunting capabilities in Microsoft Sentinel | Microsoft Learn
- Home · Azure/Azure-Sentinel Wiki · GitHub
- Threat hunting with Microsoft Sentinel - Training | Microsoft Learn

9.3.2 Splunk Search Processing Language (SPL)

Splunk is a popular log management and analysis platform. It uses the Splunk Search Processing Language (SPL) for querying and analyzing log data. SPL provides a wide range of functions and operators tailored for security and operational analysis. Threat hunting with Splunk's Search Processing Language (SPL) involves using SPL queries to search, analyze, and detect potential security threats and anomalies in your data. Here are key components of SPL that are frequently used for threat hunting:

1. **Search Command (‘*search*’)**: The search command is the core of every SPL query. It allows you to specify the data you want to search within your indexes.
2. **Time Range (‘*earliest*’ and ‘*latest*’)**: Specifying the time range helps you narrow down your search to a specific period when the threat or incident might have occurred.
3. **Filters and Conditions (‘*where*’, ‘*eval*’, ‘*match*’, etc.)**: These components are essential for filtering your data to focus on relevant events. Use where to define conditions, eval to create calculated fields, and match for pattern matching.
4. **Field Extraction (‘*rex*’ and ‘*erex*’)**: Use the rex and erex commands to extract specific fields from raw log data, making it easier to analyze and search for relevant information.
5. **Aggregation (‘*stats*’, ‘*chart*’, ‘*timechart*’)**: Aggregation commands allow you to summarize data, create charts, and visualize patterns in your data. stats provides statistical summaries, while chart and timechart create visual representations of data.
6. **Pipelining (‘|’)**: The pipeline operator (|) allows you to chain multiple commands together, enabling sequential data processing and analysis.
7. **Lookup (‘*lookup*’)**: The lookup command lets you enrich your data with additional information from external sources, such as threat intelligence feeds or reference datasets.
8. **Join (‘*join*’)**: The join command helps you combine data from multiple sources based on common fields, enabling cross-referencing and correlation.
9. **Top and Rare (‘*top*’ and ‘*rare*’)**: These commands help you identify the most frequent or infrequent events, which can be valuable for uncovering patterns or anomalies.
10. **Alerting (‘| *outputalert*’)**: While not a primary SPL component, you can use the | outputalert command to trigger alerts based on specific conditions, allowing for real-time threat detection.
11. **Timechart (‘*timechart*’)**: This command helps you visualize data over time, which is especially useful for identifying temporal patterns in your data.
12. **Transaction (‘*transaction*’)**: The transaction command groups related events together, which can help you understand the flow of activities or potential attack sequences.
13. **Eval (‘*eval*’)**: The eval command lets you create calculated fields and perform operations on existing fields, facilitating custom analysis.
14. **Subsearches (‘[| *search* ...]’)**: Subsearches allow you to embed one search within another, enabling more complex queries and correlation between different data sets.
15. **Stats (‘*stats*’)**: The stats command provides various statistical functions (e.g., count, sum, avg) to analyze and summarize data.

16. **Regular Expressions (‘*rex*’)**: Regular expressions can be used with the rex command to extract specific patterns from text fields, useful for identifying specific threat indicators.
17. **GeoIP (‘*iplocation*’)**: The iplocation command helps you translate IP addresses into geographical information, which can be valuable for identifying unusual geolocation patterns.

Source:

- Threat Hunting with Splunk: Hands-on Tutorials for the Active Hunter | Splunk
- GitHub - olafhartong/ThreatHunting: A Splunk app mapped to MITRE ATT&CK to guide your threat hunts
- Splunk - ThreatHunting toolkit

Common Threat Hunting Queries

When conducting Common Threat Hunting Queries, security analysts use a range of predefined queries that are designed to identify known threat indicators or behaviors. These queries serve as starting points for investigating specific types of threats. Some common examples include:

- **Unusual Network Traffic**: Queries that detect unusual patterns in network traffic, such as spikes in data transfers to unfamiliar IP addresses.
- **Brute Force Attacks**: Queries that identify multiple failed login attempts from a single IP address within a short time frame.
- **Data Exfiltration**: Queries that detect large volumes of data being transferred out of the organization's network.
- **Suspicious Processes**: Queries that flag unusual or unauthorized processes running on endpoints.
- **Command and Control (C2) Communication**: Queries that uncover communications with known malicious domains or IP addresses.
- **Unpatched Software**: Queries that highlight systems running outdated or vulnerable software.

These queries can be adapted and customized based on an organization's specific environment and threat landscape.

Query Optimization

Query Optimization is the process of refining and fine-tuning queries to improve their efficiency and accuracy. Effective query optimization helps reduce the number of false positives and enhances the likelihood of detecting genuine threats. Here are some strategies for query optimization:

- **Use Indexing**: Leverage database indexes to speed up query execution.

- **Limit Data Scope**: Narrow down the data being queried to relevant time frames or specific hosts.
- **Minimize Complex Operations**: Avoid complex operations that can slow down queries, such as excessive joins or subqueries.
- **Test and Iterate**: Continuously test and refine queries based on real-world results and feedback from threat hunting operations.
- **Leverage Aggregation**: Use aggregation functions to summarize and group data for easier analysis.
- **Utilize Caching**: Store frequently executed queries' results in cache for faster retrieval.

By consistently optimizing queries, threat hunters can maximize their efficiency and improve the overall effectiveness of their threat detection efforts.

Hunting query samples: Microsoft-365-Defender-Hunting-Queries

10 Threat Hunting playbook for Mitre tactics

The Threat Hunting playbook offers a structured approach for security professionals to effectively identify, analyze, and respond to potential security threats. The playbook outlines a set of sequential steps to guide threat hunting activities, providing a clear framework for proactive threat detection and mitigation. Each step should include defined objectives, tactics, assumptions, and actionable instructions, enhancing your organization's readiness to address emerging security risks. By following a playbook, the security team can systematically uncover hidden threats and vulnerabilities, enabling more targeted incident response and strengthening the overall security posture.

The following sample provides a brief insight into a threat hunting playbook, offering a concise overview of essential steps for the identification and mitigation of potential threats. By employing methodologies drawn from the MITRE ATT&CK framework, the playbook guides security practitioners through a structured approach, enabling the systematic detection and response to emerging risks within an organization's environment.

Sample

Tactic: Execution

Objective: Identify and prevent the execution of malicious code within the environment to thwart attackers' efforts.

Description: This playbook focuses on detecting and responding to tactics, techniques, and procedures (TTPs) used by threat actors to execute malicious code in your environment.

Assumption: Attackers may use various techniques to execute malicious code, such as exploiting vulnerabilities, social engineering, and leveraging legitimate tools.

Playbook Steps:

Step 1: Define Scope

- Clearly define the scope of your threat hunting activity, including the systems, applications, and users to be monitored for potential execution attempts.
- Identify relevant data sources, such as process logs, network traffic logs, and endpoint data.

Step 2: Develop Queries

- Craft SPL queries to filter and extract data related to process execution, file creation, and network activity.
- Create queries to identify suspicious processes, unusual command-line arguments, and processes with parent-child chains indicative of process injection.

Step 3: Analyze Results

- Execute the developed queries on the collected data to identify potential instances of malicious code execution.
- Analyze the results for anomalies, patterns, and deviations from the expected behavior.
- Focus on detecting scripts executed through scripting languages and memory-based attacks.

Step 4: Take Action

- Upon detecting suspicious activity, initiate an incident response process according to your organization's procedures.
- Isolate affected systems to prevent further propagation of malicious code.
- Collect additional data and artifacts for forensic analysis to understand the extent of the compromise.

Step 5: Report

- Create detailed incident reports documenting the detected execution attempts, techniques used, and affected systems.
- Include recommendations for remediation and future prevention strategies.
- Share the findings with relevant stakeholders, such as the incident response team, security management, and system administrators.

Source:

- Threat Hunter Playbook
- Threat Hunting Playbooks for Mitre Tactics

11 Threat Hunting with Machine Learning

Machine learning can play an important role in threat hunting by automating and enhancing the process of detecting anomalies and patterns that might indicate malicious behavior. Here's how machine learning is typically utilized in threat hunting:

- **Anomaly Detection**: Machine learning models can be trained on historical data to learn what "normal" behavior looks like within a system or network. These models can then be used to identify deviations from the norm, which might indicate potential threats. For example, if a user suddenly starts accessing resources they've never accessed before, or if a system starts communicating with unusual external servers, these anomalies can be flagged for further investigation.
- **Behavioral Analysis**: Machine learning algorithms can analyze the behavior of users, applications, and devices to establish baselines of behavior. Any deviations from these baselines can be treated as potential threats. By continuously updating these baselines, machine learning models can adapt to evolving attack techniques and tactics.
- **Threat Intelligence Integration**: Machine learning algorithms can be trained using threat intelligence data, which includes information about known malicious entities, attack patterns, and IoCs. By leveraging this data, machine learning models can identify similarities between observed behaviors and known attack patterns.
- **Data Correlation**: Threat hunting often involves analyzing data from various sources, such as network traffic, logs, and endpoints. Machine learning algorithms can help correlate and analyze these diverse datasets to identify patterns and connections that might not be apparent to human analysts.
- **Reducing False Positives**: Machine learning models can help reduce the number of false positive alerts generated by security systems. By learning to distinguish between normal and abnormal behaviors, these models can prioritize alerts that are more likely to be genuine threats, allowing security teams to focus their efforts more effectively.
- **Automated Response**: In some cases, machine learning can enable automated responses to certain types of threats. For instance, if a machine learning model detects a clearly malicious activity, it can trigger an automated response, such as blocking the suspicious IP address or quarantining a compromised device.
- **Continuous Learning**: Machine learning models can adapt over time by incorporating new data and feedback from security analysts. This adaptive learning allows the models to improve their accuracy in detecting both known and novel threats.

It's important to note that while machine learning can significantly enhance threat hunting, it is not a standalone solution. Effective threat hunting still requires human expertise to interpret results, investigate findings, and make decisions based on context.

As cyber threats continue to evolve, threat hunting powered by machine learning will remain an essential component of a comprehensive cybersecurity strategy. However, organizations must carefully design and implement their threat hunting processes, considering factors such as data quality, model accuracy, and the potential for adversarial attacks against machine learning systems.

Source:

- Splunk - Threat Hunting With ML
- Splunk Machine Learning Toolkit - User Guide
- Splunk Machine Learning Toolkit
- Splunk Machine Learning Toolkit - Cheat Sheet
- Splunk - Anomaly Mining in Windows Event Logs

12 Useful links

- Coursera - Threat Hunting course
- GitHub - A3sal0n/CyberThreatHunting: A collection of resources for Threat Hunters