4. Introduction to Threat Hunting & Hunting With Elastic

Threat Hunting Fundamentals

Threat Hunting Definition

The median duration between an actual security breach and its detection, otherwise termed "dwell time", is usually several weeks, if not months. This implies a potential adversarial presence within a network for a span approaching three weeks, a duration that can be significantly impactful.

This alarming fact underscores the growing inefficacy of traditional, defense-oriented cybersecurity tactics. In response, we advocate for a paradigm shift towards a proactive, offensive strategy – the initiation of threat hunting.

`Threat hunting` is an active, human-led, and often hypothesis-driven practice that systematically combs through network data to identify stealthy, advanced threats that evade existing security solutions. This strategic evolution from a conventionally reactive posture allows us to uncover insidious threats that automated detection systems or external entities such as law enforcement might not discern.

The principal objective of threat hunting is to substantially reduce dwell time by recognizing malicious entities at the earliest stage of the cyber kill chain. This proactive stance has the potential to prevent threat actors from entrenching themselves deeply within our infrastructure and to swiftly neutralize them.

The threat hunting process commences with the identification of assets – systems or data – that could be high-value targets for threat actors. Next, we analyze the TTPs (Tactics, Techniques, and Procedures) these adversaries are likely to employ, based on current threat intelligence. We subsequently strive to proactively detect, isolate, and validate any artifacts related to the abovementioned TTPs and any anomalous activity that deviates from established baseline norms.

During the hunting endeavor, we regularly employ Threat Intelligence, a vital component that aids in formulating effective hunting hypotheses, developing counter-tactics, and executing protective measures to prevent system compromise.

Key facets of threat hunting include:

- An offensive, `proactive` strategy that prioritizes threat anticipation over reaction, based on hypotheses, attacker TTPs, and intelligence.
- An offensive, `reactive` response that searches across the network for artifacts related to a `verified` incident, based on evidence and intelligence.
- A solid, practical comprehension of threat landscape, cyber threats, adversarial TTPs, and the cyber kill chain.
- Cognitive empathy with the attacker, fostering an understanding of the adversarial mindset.
- A profound knowledge of the organization's IT environment, network topology, digital assets, and normal activity.
- Utilization of high-fidelity data and tactical analytics, and leveraging advanced threat hunting tools and platforms.

The Relationship Between Incident Handling & Threat Hunting

So, how does threat hunting intersect with the various phases of Incident Handling?

- In the `Preparation` phase of incident handling, a threat hunting team must set up robust, clear rules of engagement. Operational protocols must be established, outlining when and how to intervene, the course of action in specific scenarios, and so forth. Organizations may choose to weave threat hunting into their existing incident handling policies and procedures, obviating the need for separate threat hunting policies and procedures.
- During the `Detection & Analysis` phase of incident handling, a threat hunter's acumen is indispensable. They can augment investigations, ascertain whether the observed indicators of compromise (IoCs) truly signify an incident, and further, their adversarial mindset can help uncover additional artifacts or IoCs that might have been missed initially.
- In the `Containment, Eradication, and Recovery` phase of incident handling, the role of a hunter can be diverse. Some organizations might expect hunters to perform tasks within the Containment, Eradication, and Recovery stages. However, this is not a universally accepted practice. The specific roles and responsibilities of the hunting team will be stipulated in the procedural documents and security policies.
- Regarding the `Post-Incident Activity` phase of incident handling, hunters, with their extensive expertise spanning various IT domains and IT Security, can contribute significantly. They can proffer recommendations to fortify the organization's overall security posture.

We tried to shed light on the symbiotic relationship between incident handling and threat hunting. Whether these processes should be integrated or function independently is a

strategic decision, contingent upon each organization's unique threat landscape, risk, etc.

A Threat Hunting Team's Structure

The construction of a threat hunting team is a strategic and meticulously planned process that requires a diverse range of skills, expertise, and perspectives. It is crucial that each member of the team offers a unique set of competencies that, when combined, provide a holistic and comprehensive approach to identifying, mitigating, and eliminating threats.

The ideal threat hunting team composition typically includes the following roles:

- `Threat Hunter`: The core role within the team, threat hunters are cybersecurity professionals with a deep understanding of the threat landscape, cyber adversaries' Tactics, Techniques, and Procedures (TTPs), and sophisticated threat detection methodologies. They proactively search for Indicators of Compromise (IoCs) and are proficient in using a variety of threat hunting tools and platforms.
- `Threat Intelligence Analyst`: These individuals are responsible for gathering and analyzing data from a variety of sources, including open-source intelligence, dark web intelligence, industry reports, and threat feeds. Their job is to understand the current threat landscape and predict future trends, providing valuable insights to threat hunters.
- `Incident Responders`: When threat hunters identify potential threats, incident responders step in to manage the situation. They investigate the incident thoroughly and they are also responsible for containment, eradication, and recovery actions, and they ensure that the organization can quickly resume normal operations.
- `Forensics Experts`: These are the team members who delve deep into the technical details of an incident. They are proficient in digital forensics and incident response (DFIR), capable of analyzing malware, reverse engineering attacks, and providing detailed incident reports.
- `Data Analysts/Scientists`: They play a pivotal role in examining large datasets, using statistical models, machine learning algorithms, and data mining techniques to uncover patterns, correlations, and trends that can lead to actionable insights for threat hunters.
- `Security Engineers/Architects`: Security engineers are responsible for the overall design of the organization's security infrastructure. They ensure that all systems, applications, and networks are designed with security in mind, and they often work closely with threat hunters to implement tools and techniques that facilitate threat hunting, as well as kill-chain defenses.
- `Network Security Analyst`: These professionals specialize in network behavior and traffic patterns. They understand the normal ebb and flow of network activity and can quickly identify anomalies indicative of a potential security breach.

- `SOC Manager`: The Security Operations Center (SOC) manager oversees the operations of the threat hunting team, ensuring smooth coordination among team members and effective communication with the rest of the organization.

When Should We Hunt?

In the realm of cybersecurity, threat hunting should not be seen as a sporadic or reactionary practice, but rather as a sustained, forward-thinking activity. Nevertheless, there are specific instances that call for an immediate and intense threat hunting operation. Here's a more intricate breakdown of these instances:

- `When New Information on an Adversary or Vulnerability Comes to Light`: The cybersecurity landscape is always evolving, with fresh intel on potential threats and system vulnerabilities being uncovered regularly. If there's a newly discovered adversary or a vulnerability associated with an application that our network utilizes, this calls for an immediate threat hunting session. It's imperative to decipher the adversary's modus operandi and scrutinize the vulnerability to evaluate the possible risk to our systems. For instance, if we stumble upon a previously unknown vulnerability in a widely utilized application, we'd promptly kickstart a threat hunting initiative to seek out any signs of exploitation.
- `When New Indicators are Associated with a Known Adversary`: Often, cybersecurity intelligence sources release new Indicators of Compromise (IoCs) tied to specific adversaries. If these indicators are associated with an adversary known for targeting networks akin to ours or if we've been a past target of the same adversary, we need to launch a threat hunting initiative. This aids in detecting any traces of the adversary's activities within our system, allowing us to ward off potential breaches.
- `When Multiple Network Anomalies are Detected`: Network anomalies might sometimes be harmless, caused by system glitches or valid alterations. However, several anomalies appearing concurrently or within a short period might hint at a systemic issue or an orchestrated attack. In such cases, it's crucial to carry out threat hunting to pinpoint the root cause of these anomalies and address any possible threats. For instance, if we observe odd network traffic behavior or unexpected system activities, we'd initiate threat hunting to probe these anomalies.
- `During an Incident Response Activity`: Upon the detection of a confirmed security incident, our Incident Response (IR) team will concentrate on containment, eradication, and recovery. Yet, while the IR process is in motion, it's vital to simultaneously conduct threat hunting across the network. This enables us to expose any connected threats that might not be readily visible, understand the full extent of the compromise, and avert further harm. For example, during a confirmed malware infiltration, while the IR team is dealing with the infected system, threat hunting can assist in identifying other potentially compromised systems.

- `Periodic Proactive Actions`: Beyond the scenarios mentioned above, it's crucial to note that threat hunting should not be simply a reactive task. Regular, proactive threat hunting exercises are key to discovering latent threats that may have slipped past our security defenses. This guarantees a continual monitoring strategy, bolstering our overall security stance and minimizing the prospective impact of an attack.

In a nutshell, the ideal time for threat hunting is always the present. A proactive stance on threat hunting lets us detect and neutralize threats before they can inflict substantial damage.

The Relationship Between Risk Assessment & Threat Hunting

Risk assessment, as an essential facet of cybersecurity, enables a comprehensive understanding of the potential vulnerabilities and threat vectors within an organization. In the context of threat hunting, risk assessment serves as a key enabler, allowing us to prioritize our hunting activities and focus our efforts on the areas of greatest potential impact.

To begin with, risk assessment entails a systematic process of identifying and evaluating risks based on potential threat sources, existing vulnerabilities, and the potential impact should these vulnerabilities be exploited. It involves a series of steps including asset identification, threat identification, vulnerability identification, risk determination, and finally, risk mitigation strategy formulation.

In the threat hunting process, the information gleaned from a thorough risk assessment can guide our activities in several ways:

- `Prioritizing Hunting Efforts`: By recognizing the most critical assets (often referred to as 'crown jewels') and their associated risks, we can prioritize our threat hunting efforts on these areas. Assets could include sensitive data repositories, mission-critical applications, or key network infrastructure.
- `Understanding Threat Landscape`: The threat identification step of the risk assessment allows us to understand the threat landscape better, including the Tactics, Techniques, and Procedures (TTPs) used by potential threat actors. This understanding assists us in developing our hunting hypotheses, which are essential for proactive threat hunting.
- `Highlighting Vulnerabilities`: Risk assessment helps to highlight vulnerabilities in our systems, applications, and processes. Knowing these weaknesses enables us to look for exploitation indicators in these areas. For instance, if we know a particular application has a vulnerability that allows for privilege escalation, we can look for anomalies in user privilege levels.

- `Informing the Use of Threat Intelligence`: Threat intelligence is often used in threat hunting to identify patterns of malicious behavior. Risk assessment helps inform the application of threat intelligence by identifying the most likely threat actors and their preferred methods of attack.
- `Refining Incident Response Plans`: Risk assessment also plays a critical role in refining Incident Response (IR) plans. Understanding the likely risks helps us anticipate and plan for potential breaches, ensuring a swift and effective response.
- `Enhancing Cybersecurity Controls`: Lastly, the risk mitigation strategies derived from risk assessment can directly feed into enhancing existing cybersecurity controls and defenses, further strengthening the organization's security posture.

The technicalities of employing risk assessment for threat hunting include the use of advanced tools and techniques. These range from automated vulnerability scanners and penetration testing tools to sophisticated threat intelligence platforms. For instance, SIEM (Security Information and Event Management) systems can be used to aggregate and correlate events from various sources, providing a holistic view of the organization's security status and aiding in threat hunting.

In essence, risk assessment and threat hunting are deeply intertwined, each augmenting the other to create a more robust and resilient cybersecurity posture. By regularly conducting comprehensive risk assessments, we can better focus our threat hunting activities, thereby reducing dwell time, mitigating potential damage, and enhancing our overall cybersecurity defense.

Enable step-by-step solutions for all questions
✨

Questions

Answer the question(s) below
to complete this Section and earn cubes!

+ 1 Threat hunting is used ... Choose one of the following as your answer: "proactively", "reactively", "proactively and reactively".

+10 Streak pts

Submit

+ 1 Threat hunting and incident handling are two processes that always function independently. Answer format: True, False.

+10 Streak pts

Submit

+ 1 Threat hunting and incident response can be conducted simultaneously. Answer format: True, False.

+10 Streak pts

Submit

The Threat Hunting Process

Below is a brief description of the threat hunting process:

- `Setting the Stage`: The initial phase is all about planning and preparation. It includes laying out clear targets based on a deep understanding of the threat landscape, our business's critical requirements, and our threat intelligence insights. The preparation phase also encompasses making certain our environment is ready for effective threat hunting, which might involve enabling extensive logging across our systems and ensuring threat hunting tools, such as SIEM, EDR, IDS, are correctly set up. Additionally, we stay informed about the most recent cyber threats and familiarize ourselves with threat actor profiles.
    - `Example`: During the planning and preparation phase, a threat hunting team might conduct in-depth research on the latest threat intelligence reports, analyze industry-specific vulnerabilities, and study the tactics, techniques, and procedures (TTPs) employed by threat actors. They may also identify critical assets and systems within the organization that are most likely to be targeted. As part of the preparation, extensive logging mechanisms can be implemented across servers, network devices, and endpoints to capture relevant data for analysis. Threat hunting tools like SIEM, EDR, and IDS are configured to collect and correlate logs, generate alerts, and provide visibility into potential security incidents. Additionally, the team stays updated on emerging cyber threats by monitoring threat feeds, subscribing to relevant security mailing lists, and participating in information sharing communities.
- `Formulating Hypotheses`: The next step involves making educated predictions that will guide our threat hunting journey. These hypotheses can stem from various sources, like recent threat intelligence, industry updates, alerts from security tools, or even our professional intuition. We strive to make these hypotheses testable to guide us where to search and what to look for.
    - `Example`: A hypothesis might be that an attacker has gained access to the network by exploiting a particular vulnerability or through phishing emails. This hypothesis could be derived from recent threat intelligence reports that highlight similar attack vectors. It could also be based on an alert triggered by an intrusion detection system indicating suspicious network traffic patterns. The hypothesis should be specific and testable, such as "An advanced persistent threat (APT) group is leveraging a known vulnerability in the organization's web server to establish a command-and-control (C2) channel."

- `Designing the Hunt`: Upon crafting a hypothesis, we need to develop a hunting strategy. This includes recognizing the specific data sources that need analysis, the methodologies and tools we'll use, and the particular indicators of compromise (IoCs) or patterns we'll hunt for. At this point, we might also create custom scripts or queries and utilize dedicated threat hunting tools.
    - `Example`: The threat hunting team may decide to analyze web server logs, network traffic logs, DNS logs, or endpoint telemetry data. They define the search queries, filters, and correlation rules to extract relevant information from the collected data. The team also leverages threat intelligence feeds and open-source intelligence (OSINT) to identify specific indicators of compromise (IoCs) associated with the suspected threat actor or known attack techniques. This may involve crafting custom scripts or queries to search for IoCs or using specialized threat hunting platforms that automate the process.
- `Data Gathering and Examination`: This phase is where the active threat hunt occurs. It involves collecting necessary data, such as log files, network traffic data, endpoint data, and then analyzing this data using the predetermined methodologies and tools. Our goal is to find evidence that either supports or refutes our initial hypothesis. This phase is highly iterative, possibly involving refinement of the hypothesis or the investigation approach as we uncover new information.
    - `Example`: The threat hunting team might examine web server access logs to identify unusual or unauthorized access patterns, analyze network traffic captures to detect suspicious communications with external domains, or investigate endpoint logs to identify anomalous behavior or signs of compromise. They apply data analysis techniques such as statistical analysis, behavioral analysis, or signature-based detection to identify potential threats. They might employ tools like log analyzers, packet analyzers, or malware sandboxes to extract information from the collected data and uncover hidden indicators of compromise.
- `Evaluating Findings and Testing Hypotheses`: After analyzing the data, we need to interpret the results. This could involve confirming or disproving the hypothesis, understanding the behavior of any detected threats, identifying affected systems, or determining the potential impact of the threat. This phase is crucial, as it will inform the next steps in terms of response and remediation.
    - `Example`: The threat hunting team might discover a series of failed login attempts from an IP address associated with a known threat actor, confirming the hypothesis of an attempted credential brute-force attack. They might also find evidence of suspicious outbound network connections to known malicious domains, supporting the hypothesis of a command-and-control (C2) communication channel. The team conducts deeper investigations to understand the behavior of the identified threats, assess the scope of the compromise, and determine the potential impact on the organization's systems and data.
- `Mitigating Threats`: If we confirm a threat, we must undertake remediation actions. This could involve isolating affected systems, eliminating malware, patching

vulnerabilities, or modifying configurations. Our goal is to eradicate the threat and limit any potential damage.

    - `Example`: If the threat hunting team identifies a compromised system communicating with a C2 server, they may isolate the affected system from the network to prevent further data exfiltration or damage. They may deploy endpoint protection tools to remove malware or perform forensic analysis on the compromised system to gather additional evidence and determine the extent of the breach. Vulnerabilities identified during the threat hunting process can be patched or mitigated to prevent future attacks. Network configurations can be adjusted to restrict unauthorized access or to strengthen security controls.

- `After the Hunt`: Once the threat hunting cycle concludes, it's crucial to document and share the findings, methods, and outcomes. This might involve updating threat intelligence platforms, enhancing detection rules, refining incident response playbooks, or improving security policies. It's also vital to learn from each threat hunting mission to enhance future efforts.
    - `Example`: Once the threat hunting cycle concludes, the team documents the findings, methodologies, and outcomes of the investigation. They update threat intelligence platforms with newly discovered indicators of compromise (IoCs) and share relevant information with other teams or external partners to enhance the collective defense against threats. They may improve detection rules within security tools based on the observed attack patterns and refine incident response playbooks to streamline future incident handling. Lessons learned from the hunt are incorporated into security policies and procedures, and training programs are adjusted to enhance the organization's overall security posture.
- `Continuous Learning and Enhancement`: Threat hunting is not a one-time task, but a continuous process of learning and refinement. Each threat hunting cycle should feed into the next, allowing for continuous improvement of hypotheses, methodologies, and tools based on the evolving threat landscape and the organization's changing risk profile.
    - `Example`: After each threat hunting cycle, the team reviews the effectiveness of their hypotheses, methodologies, and tools. They analyze the results and adjust their approach based on lessons learned and new threat intelligence. For example, they might enhance their hunting techniques by incorporating machine learning algorithms or behavioral analytics to detect more sophisticated threats. They participate in industry conferences, attend training sessions, and collaborate with other threat hunting teams to stay updated on the latest attack techniques and defensive strategies.

Threat hunting is a delicate balance of art and science. It demands technical prowess, creativity, and a profound understanding of both the organization's environment and the broader threat landscape. The most successful threat hunting teams are those that learn from each hunt and constantly hone their skills and processes.

The Threat Hunting Process VS Emotet

Let's see how the abovementioned threat hunting process could have been applied to hunt for emotet malware within an organization.

- `Setting the Stage`: During the planning and preparation phase, the threat hunting team extensively researches the Emotet malware's tactics, techniques, and procedures (TTPs) by studying previous attack campaigns, analyzing malware samples, and reviewing threat intelligence reports specific to Emotet. They gain a deep understanding of Emotet's infection vectors, such as malicious email attachments or links, and the exploitation of vulnerabilities in software or operating systems. The team identifies critical assets and systems that are commonly targeted by Emotet, such as endpoints with administrative privileges or email servers.
- `Formulating Hypotheses`: Hypotheses in the context of Emotet threat hunting might be based on known Emotet IoCs or patterns observed in previous attacks. For example, a hypothesis could be that Emotet is using a new phishing technique to distribute malicious payloads via compromised email accounts. This hypothesis could be derived from recent threat intelligence reports highlighting similar Emotet campaigns or based on alerts triggered by email security systems detecting suspicious email attachments. The hypothesis should be specific, such as "Emotet is using compromised email accounts to send phishing emails with malicious Word documents containing macros."
- `Designing the Hunt`: In the design phase, the threat hunting team determines the relevant data sources and collection methods to validate or invalidate the Emotet-related hypotheses. They may decide to analyze email server logs, network traffic logs, endpoint logs, or sandboxed malware samples. They define search queries, filters, and correlation rules to extract information related to Emotet's specific characteristics, such as email subject lines, attachment types, or network communication patterns associated with Emotet infections. They leverage threat intelligence feeds to identify Emotet-related IoCs, such as known command-and-control (C2) server addresses or file hashes associated with Emotet payloads.
- `Data Gathering and Examination`: During the active threat hunting phase, the team collects and analyzes data from various sources to detect Emotet-related activities. For example, they might examine email server logs to identify patterns of suspicious email attachments or analyze network traffic captures to detect communication with known Emotet C2 servers. They apply data analysis techniques, such as email header analysis, network traffic pattern analysis, or behavioral analysis, to identify potential Emotet infections. They utilize tools like email forensics software, network packet analyzers, or sandbox environments to extract relevant information from the collected data and uncover hidden indicators of Emotet activity.
- `Evaluating Findings and Testing Hypotheses`: In this phase, the team evaluates the findings from data analysis to confirm or refute the initial Emotet-related hypotheses. For example, they might discover a series of emails with similar subject

lines and attachment types associated with Emotet campaigns, confirming the hypothesis of ongoing Emotet phishing activities. They might also find evidence of network connections to known Emotet C2 servers, supporting the hypothesis of an active Emotet infection. The team conducts deeper investigations to understand the behavior of the identified Emotet infections, assess the scope of the compromise, and determine the potential impact on the organization's systems and data.

- `Mitigating Threats`: If Emotet infections are confirmed, the team takes immediate remediation actions. They isolate affected systems from the network to prevent further spread of the malware and potential data exfiltration. They deploy endpoint protection tools to detect and remove Emotet malware from compromised systems. Additionally, they analyze compromised email accounts to identify and remove unauthorized access. They patch or mitigate vulnerabilities exploited by Emotet to prevent future infections. Network configurations are adjusted to block communication with known Emotet C2 servers or malicious domains.
- `After the Hunt`: Once the Emotet threat hunting cycle concludes, the team documents their findings, methodologies, and outcomes. They update threat intelligence platforms with new Emotet-related IoCs and share relevant information with other teams or external partners to enhance their collective defense against Emotet. They improve detection rules within security tools based on the observed Emotet attack patterns and refine incident response playbooks to streamline future incident handling. Lessons learned from the Emotet hunt are incorporated into security policies and procedures, and training programs are adjusted to enhance the organization's overall defenses against Emotet and similar malware.
- `Continuous Learning and Enhancement`: Threat hunting for Emotet is an ongoing process that requires continuous learning and improvement. After each Emotet threat hunting cycle, the team reviews the effectiveness of their hypotheses, methodologies, and tools. They analyze the results and adjust their approach based on lessons learned and new Emotet-related threat intelligence. For example, they might enhance their hunting techniques by incorporating advanced behavior-based detection mechanisms or machine learning algorithms specifically designed to identify Emotet's evolving TTPs. They actively participate in industry conferences, attend training sessions, and collaborate with other threat hunting teams to stay updated on the latest Emotet attack techniques and defensive strategies.

Enable step-by-step solutions for all questions
✨

Questions

Answer the question(s) below
to complete this Section and earn cubes!

+ 1 It is OK to formulate hypotheses that are not testable. Answer format: True, False.

+10 Streak pts

Submit

Threat Hunting Glossary

Within the domain of cybersecurity and threat hunting, several crucial terms and concepts play a pivotal role. Here's an enriched understanding of these:

- `Adversary`: An adversary, within the realm of Cyber Threat Intelligence (CTI), refers to an entity driven by shared objectives as your organization, albeit unauthorized, seeking to infiltrate your business and satisfy their collection requirements, which may include financial gains, insider information, or valuable intellectual property. These adversaries possess varying levels of technical expertise and are motivated to circumvent your security measures.

Adversaries can be classified into distinct categories, including `cyber criminals`, `insider threats`, `hacktivists`, or `state-sponsored operators`. Each category exhibits unique characteristics and motivations in their pursuit of unauthorized access and exploitation.

- `Advanced Persistent Threat (APT)`: APTs are typically associated with highly organized groups or nation-state entities that possess extensive resources, thereby enabling them to carry out their malicious activities over prolonged periods. While APTs target various sectors, they show a marked preference for high-value targets, which can include governmental organizations, healthcare infrastructures, and defense systems.

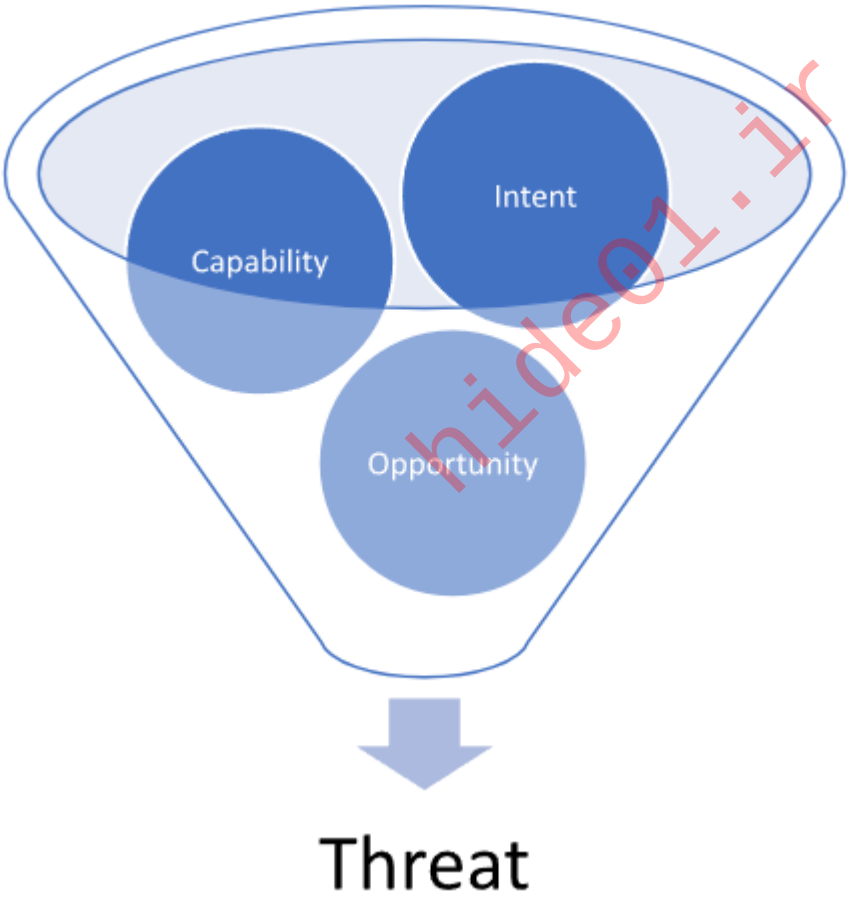
Contrary to what the name might suggest, being labeled as an APT doesn't necessarily imply that the group utilizes technologically advanced techniques. Rather, the 'Advanced' aspect can refer to the sophisticated strategic planning, and 'Persistent' alludes to their dogged persistence in achieving their objectives, backed by substantial resources including, but not limited to, financial backing, manpower, and time.

- `Tactics, Techniques, and Procedures (TTPs)`: A term borrowed from the military, TTPs symbolize the distinct operational patterns or 'signature' of an adversary.
  - `Tactics`: This term describes the strategic objectives and high-level concepts of operations employed by the adversary. Essentially, it addresses the 'why' behind their actions.
  - `Techniques`: These are the specific methods utilized by an adversary to accomplish their tactical objectives, providing the 'how' behind their actions. Techniques don't provide step-by-step instructions but rather describe the general approach to achieving a goal.
  - `Procedures`: These are the granular, step-by-step instructions, essentially the 'recipe' for the implementation of each technique.

Analyzing TTPs offers deep insights into how an adversary penetrates a network, moves laterally within it, and achieves their objectives. Understanding TTPs allows for the creation of Indicators of Compromise (IOCs), which can help detect and thwart future attacks.

- `Indicator`: An indicator, when analyzed in CTI, encompasses both technical data and contextual information. Isolated technical data lacking relevant context holds limited or negligible value for network defenders. Contextual details allow for a comprehensive understanding of the indicator's significance, enabling effective threat analysis and response.

- `Threat`: A threat is a multifaceted concept, consisting of three fundamental factors, intent, capability, and opportunity.

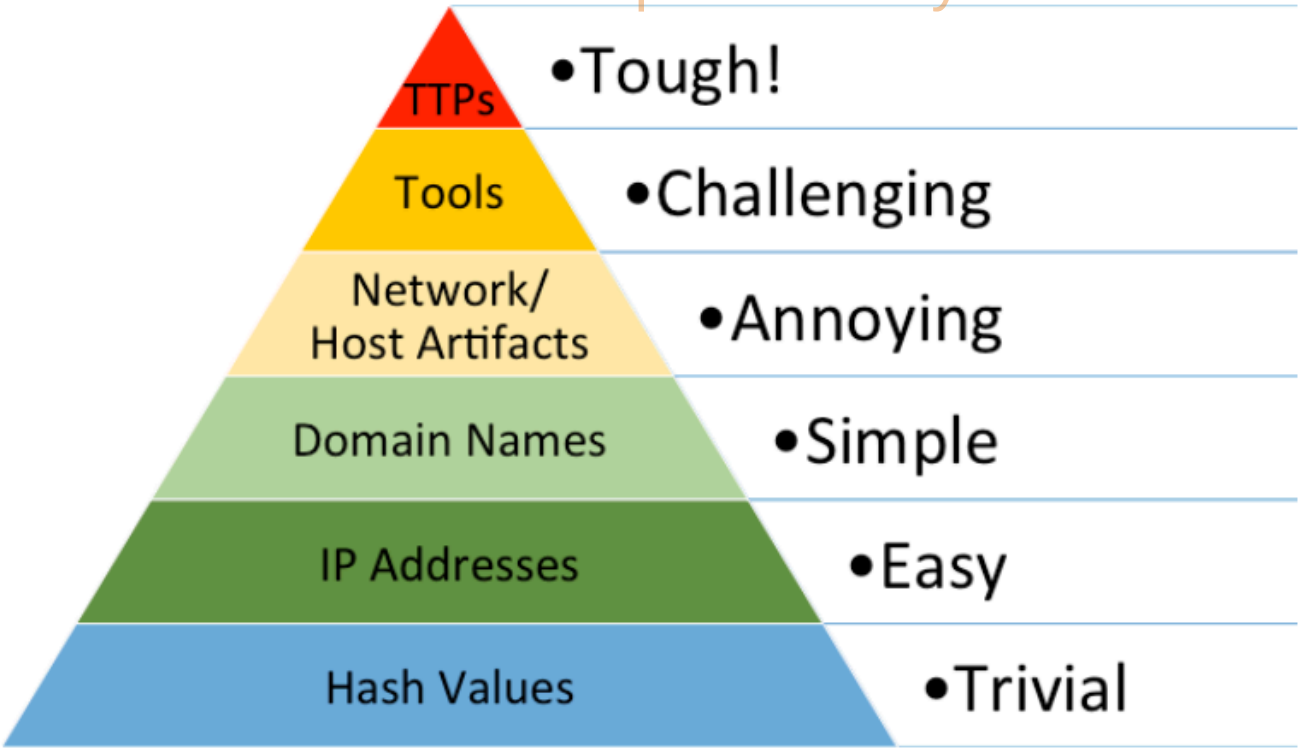


Firstly, `intent` signifies the underlying rationale driving adversaries to target and exploit your network infrastructure. This intent can range from corporate espionage to financial gains through cybercrime, or even targeting your business relationships with other entities.

Secondly, `capability` denotes the tools, resources, and financial backing that adversaries possess to carry out their operations successfully. Their skill level in penetrating your network and the availability of sufficient financial resources determine their capability to sustain ongoing attacks against your organization.

Lastly, `opportunity` refers to conditions or events that provide favorable circumstances for adversaries to execute their operations. This encompasses instances where adversaries acquire relevant email addresses or credentials from your network, as well as their awareness of vulnerabilities in specific software systems.

- `Campaign`: A campaign refers to a collection of incidents that share similar Tactics, Techniques, and Procedures (TTPs) and are believed to have comparable collection requirements. This type of intelligence necessitates substantial time and effort to aggregate and analyze, as businesses and organizations progressively report and uncover related malicious activities.
- `Indicators of Compromise (IOCs)`: IOCs are digital traces or artifacts derived from active or past intrusions. They serve as 'signposts' of a specific adversary or malicious activity. IOCs can include a wide array of elements such as the hashes of malicious files, suspicious IP addresses, URLs, domain names, and names of malicious executables or scripts. Continually tracking, cataloging, and analyzing IOCs can greatly enhance our threat detection capabilities, leading to faster and more effective responses to cyber threats.
- `Pyramid of Pain`: Pyramid of Pain is a critical visualization which presents a hierarchy of indicators that can support us in detecting adversaries. It also showcases the degree of difficulty in acquiring these specific indicators and the subsequent impact of gathering intelligence on them. The Pyramid of Pain concept was brought to life by David Bianco from FireEye in his insightful presentation, Intel-Driven Detection and Response to Increase Your Adversary's Cost of Operations. As we ascend the Pyramid of Pain, obtaining adversary-specific Indicators of Compromise (IOCs) becomes increasingly challenging. However, the flip side is that acquiring these specific IOCs forces the adversary to alter their attack methodologies, a task that is far from simple for them.



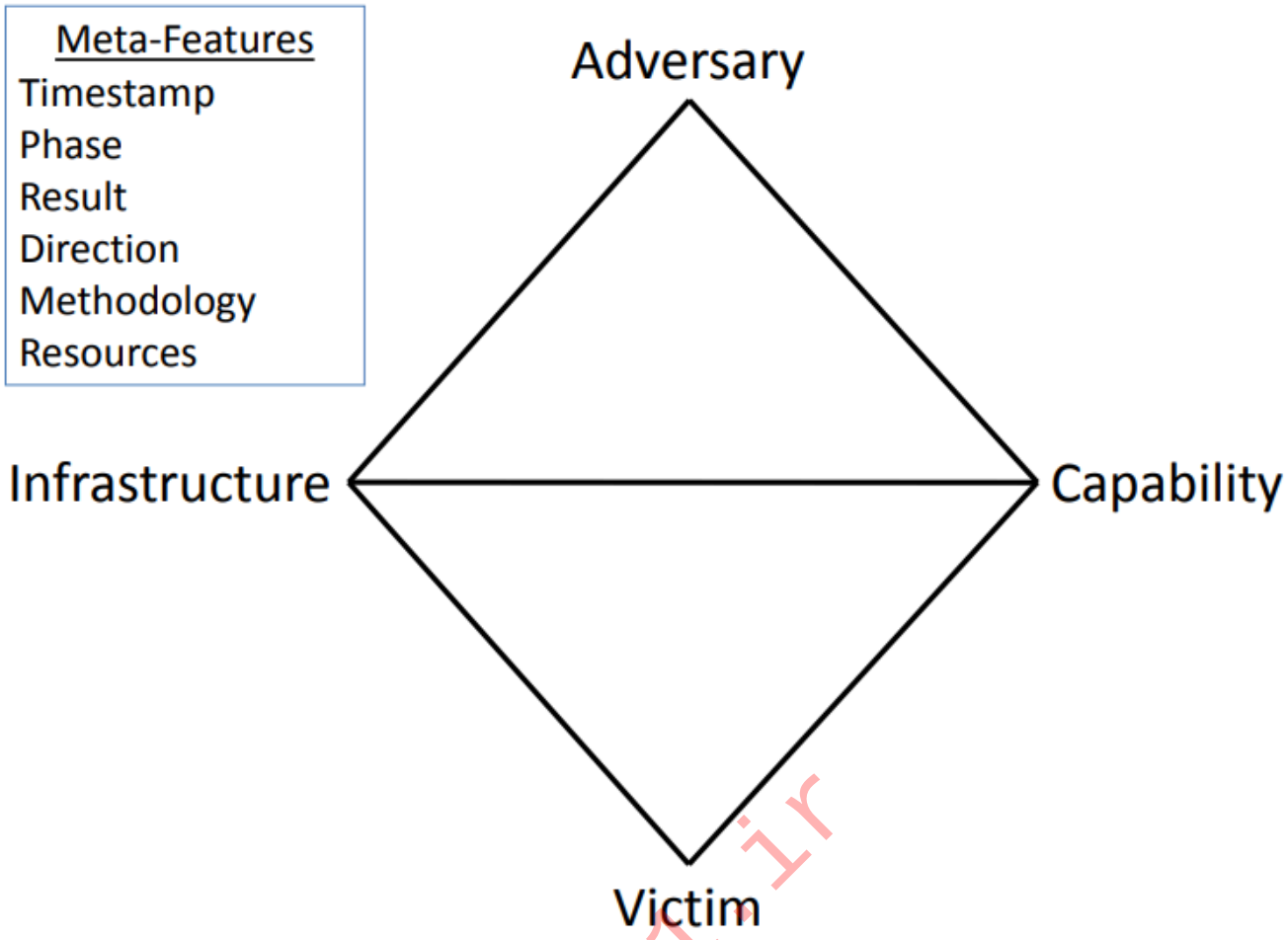
- `Hash Values`: Hash values are the digital fingerprints of files. They are created using algorithms like MD5, SHA-1, or SHA-256 that take an input (or 'message') and return a fixed-size string of bytes. For instance, malware binaries can be identified through their unique hash values. However, a slight change to the file, such as adding a byte or changing a single character, will dramatically alter the hash value, making it an easy-to-change and, therefore, less reliable indicator.
- `IP Addresses`: IP addresses are unique identifiers for devices on a network. They can be used to track the source of network traffic or a potential attack. However, adversaries often use tactics such as IP spoofing, VPNs, proxies, or TOR networks to hide their true IP addresses, making this level of indicator easy to change and somewhat unreliable.
- `Domain Names`: Domains are used to identify one or more IP addresses. For example, the domain name www.example.com represents about a dozen IP addresses. Malicious actors often use domain generation algorithms (DGAs) to produce a large number of pseudo-random domain names to evade detection. They can also use dynamic DNS services to quickly change the IP addresses associated with a domain.
- `Network/Host Artifacts`:
  - `Network Artifacts`: These are residual traces of an attacker's activities within the network infrastructure. They can be found in network logs, packet captures, netflow data, or DNS request logs, to name a few. Examples might include certain patterns in network traffic, unique packet headers, or unusual protocol usage. Network artifacts are challenging for an attacker to modify without impacting the effectiveness or stealth of their operation.
  - `Host Artifacts`: On the other hand, host artifacts refer to remnants of malicious activity left on individual systems or endpoints. These could be found within system logs, file systems, registry keys, list of running processes, loaded DLLs, or

even in volatile memory. For instance, unusual entries in the Windows Registry, unique file paths, or suspicious running processes could all be considered host artifacts. These indicators are also fairly hard for an adversary to alter without affecting their intrusion campaign or revealing their presence.

    - Analyzing these artifacts can provide valuable insights into an adversary's tools, techniques, and procedures (TTPs), and help in the detection and prevention of future attacks. However, the higher position of Network and Host Artifacts in the Pyramid of Pain indicates that they are harder to utilize for detection, and also harder for the attacker to change or obfuscate.

- `Tools`: Tools refer to the software used by adversaries to conduct their attacks. This could include malware, exploits, scripts, or command and control (C2) frameworks. Identifying the tools used by an adversary can provide valuable insight into their capabilities and intentions. However, sophisticated adversaries often use custom tools or modify existing ones to evade detection.
- `TTPs (Tactics, Techniques, and Procedures)`: This is the pinnacle of the Pyramid of Pain. TTPs refer to the specific methods used by adversaries to conduct their attacks. Tactics describe the adversary's overall objectives, techniques describe the actions taken to achieve those objectives, and procedures are the exact steps taken to execute the techniques. Identifying an adversary's TTPs can provide the most valuable insight into their operations and are the most difficult for an adversary to change without significant cost and effort. Examples might include the use of spear-phishing emails for initial access (tactic), exploitation of a specific software vulnerability (technique), and the specific steps taken to exploit that vulnerability (procedure).
- `Diamond Model`: The Diamond Model of Intrusion Analysis is a conceptual framework designed to illustrate the fundamental aspects of a cyber intrusion. This model, developed by Sergio Caltagirone, Andrew Pendergast, and Christopher Betz, aims to provide a more structured approach to understand, analyze, and respond to cyber threats.

The model is structured around four key components, represented as vertices of a diamond:



- `Adversary`: This represents the individual, group, or organization responsible for the cyber intrusion. It's important to understand their capabilities, motivations, and intent to effectively defend against their attacks.
- `Capability`: This represents the tools, techniques, and procedures (TTPs) that the adversary uses to carry out the intrusion. This could include malware, exploits, and other malicious tools, as well as the specific methods used to deploy these tools.
- `Infrastructure`: This represents the physical and virtual resources that the adversary uses to facilitate the intrusion. It can include servers, domain names, IP addresses, and other network resources used to deliver malware, control compromised systems, or exfiltrate data.
- `Victim`: This represents the target of the intrusion, which could be an individual, organization, or system. Understanding the victim's vulnerabilities, the value of their assets, and their potential exposure to threats is crucial for effective defense.

These four components are connected by bidirectional arrows, representing the dynamic relationships and interactions between them. For example, an adversary uses capabilities through an infrastructure to target a victim. This model allows for the capture of complex relationships and the construction of robust strategies for threat detection, mitigation, and prediction.

Comparing this to the Cyber Kill Chain model, we can see that the Diamond Model provides a more detailed view of the cyber intrusion ecosystem. While the Cyber Kill Chain focuses

more on the stages of an attack (from reconnaissance to actions on objectives), the Diamond Model provides a more holistic view of the components involved in the intrusion and their interrelationships.

Let's consider a technical example to illustrate the Diamond Model:
Suppose a financial institution (Victim) is targeted by a cybercriminal group (Adversary). The group uses spear-phishing emails (Capability) sent from a botnet (Infrastructure) to deliver a banking Trojan. When a recipient clicks on a malicious link in the email, the Trojan is installed on their system, allowing the cybercriminals to steal sensitive financial data.

In this scenario, the Diamond Model helps to highlight the interplay between the different components of the intrusion. By analyzing these components and their interactions, the financial institution can gain a deeper understanding of the threat they're facing and develop more effective strategies for mitigating this and future threats. This could involve strengthening their email security protocols, monitoring for signs of the specific banking Trojan, or implementing measures to detect and respond to unusual network activity associated with the botnet.

Overall, the Diamond Model provides a complementary perspective to the Cyber Kill Chain, offering a different lens through which to understand and respond to cyber threats. Both models can be useful tools in the arsenal of a cybersecurity professional.

Threat Intelligence Fundamentals

Cyber Threat Intelligence Definition

`Cyber Threat Intelligence (CTI)` represents a vital asset in our arsenal, providing essential insights to fortify our defenses against cyberattacks. The primary objective of our CTI team is to transition our defense strategies from merely reactive measures to a more proactive, anticipatory stance. They contribute crucial insights to our Security Operations Center (SOC).

Four fundamental principles make CTI an integral part of our cybersecurity strategy:

- `Relevance`: The cyber world is awash with diverse sources of information, from social media posts and security vendor reports to shared insights from similar organizations. However, the true value of this information lies in its relevance to our organization. For instance, if there is a reported vulnerability in a software that we, or our trusted partner organizations, do not use, the urgency to implement defensive measures is naturally diminished.
- `Timeliness`: Swift communication of intelligence to our defense team is crucial for the implementation of effective mitigation measures. The value of information depreciates over time - freshly discovered data is more valuable, and 'aged' indicators lose their relevance as they might no longer be used by the adversary or may have been resolved by the affected organization.
- `Actionability`: Data under analysis by a CTI analyst should yield actionable insights for our defense team. If the intelligence doesn't offer clear directives for action, its value diminishes. Intelligence must be scrutinized until it yields relevant, timely, and actionable insights for our network defense. Unactionable intelligence can lead to a self-perpetuating cycle of non-productive analysis, often referred to as a "self-licking ice cream cone".
- `Accuracy`: Before disseminating any intelligence, it must be verified for accuracy. Incorrect indicators, misattributions, or flawed Tactics, Techniques, and Procedures (TTPs) can result in wastage of valuable time and resources. If the accuracy of any information is uncertain, it should be labeled with a confidence indicator, ensuring that our defense team is aware of potential inaccuracies.

When these four factors synergize, the intelligence gleaned allows us to:

- Gain insights into potential adversary operations and campaigns that might be targeting our organization.
- Enrich our data pool through analysis by CTI analysts and other network defenders.
- Uncover adversary TTPs, enabling the development of effective mitigation measures and enhancing our understanding of adversary behavior.
- Provide decision-makers within our organization with pertinent information for informed, impactful decision-making related to business operations.

The Difference Between Threat Intelligence & Threat Hunting

Threat Intelligence and Threat Hunting represent two distinct, yet intrinsically interconnected, specialties within the realm of cybersecurity. While they serve separate functions, they both contribute significantly to the development of a comprehensive security analyst. However, it's important to note that they are not substitutes for each other.

Threat Intelligence (Predictive): The primary aim here is to anticipate the adversary's moves, ascertain their targets, and discern their methods of information acquisition. The adversary has a specific objective, and as a team involved in Threat Intelligence, our mission is to predict:

- The location of the intended attack
- The timing of the attack
- The operational strategies the adversary will employ
- The ultimate objectives of the adversary

Threat Hunting (Reactive and Proactive): Yes, the two terms are opposites, but they encapsulate the essence of Threat Hunting. An initiating event or incident, whether it occurs within our network or in a network of a similar industry, prompts our team to launch an operation to ascertain whether an adversary is present in the network, or if one was present and evaded detection.

Ultimately, Threat Intelligence and Threat Hunting bolster each other, strengthening our organization's overall network defense posture. As our Threat Intelligence team analyzes adversary activities and develops comprehensive adversary profiles, this information can be shared with our Threat Hunting analysts to inform their operations. Conversely, the findings from Threat Hunting operations can equip our Threat Intelligence analysts with additional data to refine their intelligence and enhance the accuracy of their predictions.

Criteria Of Cyber Threat Intelligence

What truly makes Cyber Threat Intelligence (CTI) valuable? What issues does it resolve? As discussed earlier, for CTI to be effective, it must be Actionable, Timely, Relevant, and Accurate. These four elements form the foundation of robust CTI that ultimately provides visibility into adversary operations. Additionally, well-constructed CTI brings forth secondary benefits, such as:

- Understanding of threats to our organization and partner entities
- Potential insights into our organization's network
- Enhanced awareness of potential problems that may have gone unnoticed

Furthermore, from a leadership standpoint, high-quality CTI aids in fulfilling the business objective of minimizing risk as much as possible. As intelligence about an adversary targeting our business is gathered and analyzed, it empowers leadership to adequately assess the risk, formulate a contingency action plan if an incident occurs, and ultimately frame the problem and disseminate the information in a coherent and meaningful way.

As this information is compiled, it transforms into intelligence. This intelligence can then be classified into three different categories, each having varying degrees of relevance for different teams within our organization. These categories are:

- Strategic Intelligence
- Operational Intelligence
- Tactical Intelligence

In the diagram below, the ideal intersection is right at the core. At this convergence juncture, the Cyber Threat Intelligence (CTI) analyst is equipped to offer the most comprehensive and detailed portrait of the adversary and their modus operandi.

`Strategic Intelligence` is characterized by:

- Being consumed by C-suite executives, VPs, and other company leaders
- Aiming to align intelligence directly with company risks to inform decisions
- Providing an overview of the adversary's operations over time
- Mapping TTPs and Modus Operandi (MO) of the adversary
- Striving to answer the Who? and Why?
- `Example`: A report containing strategic intelligence might outline the threat posed by APT28 (also known as Fancy Bear), a nation-state actor linked to the Russian government. This report could cover the group's past campaigns, its motivations (such as political espionage), targets (like governments, military, and security organizations), and long-term strategies. The report might also explore how the group adapts its tactics and tools over time, based on historical data and the geopolitical context.

`Operational Intelligence` is characterized by:

- Also including TTPs of an adversary (similar to strategic intelligence)
- Providing information on adversary campaigns
- Offering more detail than what's found in strategic intelligence reports
- Being produced for mid-level management personnel
- Working towards answering the How? and Where?
- `Example`: A report containing operational intelligence can provide detailed analysis of a ransomware campaign conducted by the REvil group. It would include how the group gains initial access (like through phishing or exploiting vulnerabilities), its lateral movement tactics (such as credential dumping and exploiting Windows admin tools), and its methods of executing the ransomware payload (maybe after hours to maximize damage and encrypt as many systems as possible).

`Tactical Intelligence` is characterized by:

- Delivering immediate actionable information
- Being provided to network defenders for swift action
- Including technical details on attacks that have occurred or could occur in the near future
- `Example`: A report containing tactical intelligence could include specific IP addresses, URLs, or domains linked to the REvil command and control servers, hashes of known REvil ransomware samples, specific file paths, registry keys, or mutexes associated with REvil, or even distinctive strings within the ransomware code. This type of information can be directly used by security technologies and incident responders to detect, prevent, and respond to specific threats.

It's crucial to understand that there's a degree of overlap among these three types of intelligence. That's why we represent the intelligence in a Venn diagram. Tactical intelligence contributes to forming an operational picture and a strategic overview. The converse is also true.

How To Go Through A Tactical Threat Intelligence Report

Interpreting threat intelligence reports loaded with tactical intelligence and Indicators of Compromise (IOCs) is a task that requires a structured methodology to optimize our responsiveness as SOC analysts or threat hunters. Let's delve into a procedural, in-depth process using a theoretical scenario involving a threat intelligence report on an elaborate Emotet malware campaign:

- `Comprehending the Report's Scope and Narrative`: The initial phase of interpreting the report involves comprehending its broader context. Suppose our report elucidates an ongoing Emotet campaign directed towards businesses in our sector. The report may offer macro-level insights about the attackers' objectives and the types of entities in their crosshairs. By grasping the narrative, we can assess the pertinence of the threat to our own business.
- `Spotting and Classifying the IOCs`: Tactical intelligence typically encompasses a list of IOCs tied to the threat. In the context of our Emotet scenario, these might include IP addresses linked to command-and-control (C2) servers, file hashes of the Emotet payloads, email addresses or subject lines leveraged in phishing campaigns, URLs of deceptive websites, or distinct Registry alterations by the malware. We should partition these IOCs into categories for more comprehensible understanding and actionable results: Network-based IOCs (IPs, domains), Host-based IOCs (file hashes, registry keys), and Email-based IOCs (email addresses, subject lines). Furthermore, IOCs could also contain Mutex names generated by the malware, SSL certificate hashes, specific API calls enacted by the malware, or even patterns in network traffic (such as specific User-Agents, HTTP headers, or DNS request patterns). Moreover, IOCs can be augmented with supplementary data. For instance, IP addresses can be supplemented with geolocation data, WHOIS information, or associated domains.
- `Comprehending the Attack's Lifecycle`: The report will likely depict the Tactics, Techniques, and Procedures (TTPs) deployed by the attackers, correspondingly mapped to the MITRE ATT&CK framework. For the Emotet campaign, it might commence with a spear-phishing email (Initial Access), proceed to execute the payload (Execution), establish persistence (Persistence), execute defense evasion tactics (Defense Evasion), and ultimately exfiltrate data or deploy secondary payloads (Command and Control). Comprehending this lifecycle aids us in forecasting the attacker's moves and formulating an effective response.
- `Analysis and Validation of IOCs`: Not all IOCs hold the same utility or accuracy. We need to authenticate them, typically by cross-referencing with additional threat intelligence sources or databases such as VirusTotal or AlienVault's OTX. We also need to contemplate the age of IOCs. Older ones may not be as pertinent if the attacker has modified their infrastructure or tactics. Moreover, contextualizing IOCs is critical for their correct interpretation. For example, an IP address employed as a C2 server may also host legitimate websites due to IP sharing in cloud environments. Analysts should also consider the source's reliability and whether the IOC has been whitelisted in the past. Ultimately, understanding the false positive rate is crucial to avoid alert fatigue.
- `Incorporating the IOCs into our Security Infrastructure`: Once authenticated, we can integrate these IOCs into our security solutions. This might involve updating firewall rules with malicious IP addresses or domains, incorporating file hashes into our endpoint detection and response (EDR) solution, or creating new IDS/IPS signatures. For email-based IOCs, we can update our email security gateway or anti-spam solution. When implementing IOCs, we should consider the potential impact on business operations. For example, blocking an IP address might affect a

business-critical service. In such cases, alerting rather than blocking might be more appropriate. Additionally, all changes should be documented and approved following change management procedures to maintain system integrity and avoid unintentional disruptions.

- `Proactive Threat Hunting`: Equipped with insights from the report, we can proactively hunt for signs of the Emotet threat in our environment. This might involve searching logs for network connections to the C2 servers, scanning endpoints for the identified file hashes, or checking email logs for the phishing email indicators. Threat hunting shouldn't be limited to searching for IOCs. We should also look for broader signs of TTPs described in the report. For instance, Emotet often employs PowerShell for execution and evasion. Therefore, we might hunt for suspicious PowerShell activity, even if it doesn't directly match an IOC. This approach aids in detecting variants of the threat not covered by the specific IOCs in the report.
- `Continuous Monitoring and Learning`: After implementing the IOCs, we must continually monitor our environment for any hits. Any detection should trigger a predefined incident response process. Furthermore, we should utilize the information gleaned from the report to enhance our security posture. This could involve user education around the phishing tactics employed by the Emotet group or improving our detection rules to catch the specific evasion techniques employed by this malware. While we should unquestionably learn from each report, we should also contribute back to the threat intelligence community. If we discover new IOCs or TTPs, these should be shared with threat intelligence platforms and ISACs/ISAOs (Information Sharing and Analysis Centers/Organizations) to aid other organizations in defending against the threat.

This meticulous, step-by-step process, while tailored to our Emotet example, can be applied to any threat intelligence report containing tactical intelligence and IOCs. The secret is to be systematic, comprehensive, and proactive in our approach to maximize the value we derive from these reports.

Enable step-by-step solutions for all questions
✨

Questions

Answer the question(s) below
to complete this Section and earn cubes!

+ 1 It’s useful for the CTI team to provide a single IP with no context to the SOC team.
Answer format: True, False.

+10 Streak pts

Submit

+ 1 When an incident occurs on the network and the CTI team is made aware, what should they do? Choose one of the following as your answer: "Do Nothing", "Reach out to the Incident Handler/Incident Responder", "Provide IOCs on all research being conducted, regardless if the IOC is verified".

+10 Streak pts

Submit

+ 1 When an incident occurs on the network and the CTI team is made aware, what should they do? Choose one of the following as your answer: "Provide IOCs on all research being conducted, regardless if the IOC is verified", "Do Nothing", "Provide further IOCs and TTPs associated with the incident".

+10 Streak pts

Submit

+ 1 Cyber Threat Intelligence, if curated and analyzed properly, can ... ? Choose one of the following as your answer: "be used for security awareness", "be used for fine-tuning network segmentation", "provide insight into adversary operations".

+10 Streak pts

Submit

Hunting For Stuxbot

Threat Intelligence Report: Stuxbot

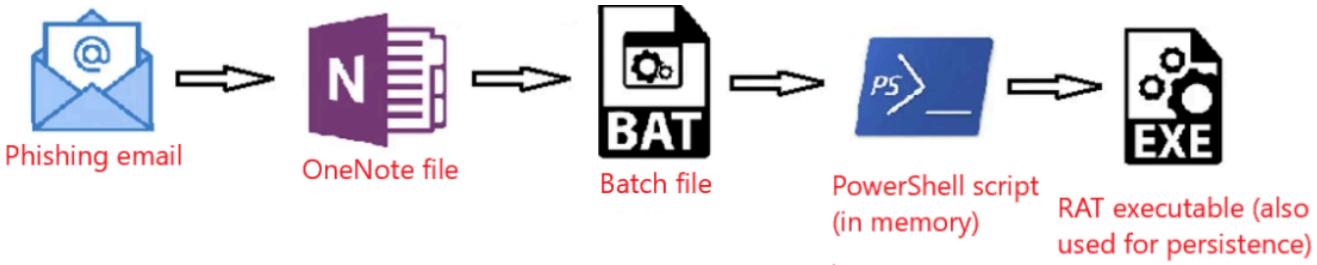
The present Threat Intelligence report underlines the immediate menace posed by the organized cybercrime collective known as "Stuxbot". The group initiated its phishing campaigns earlier this year and operates with a broad scope, seizing upon opportunities as they arise, without any specific targeting strategy – their motto seems to be anyone, anytime. The primary motivation behind their actions appears to be espionage, as there have been no indications of them exfiltrating sensitive blueprints, proprietary business information, or seeking financial gain through methods such as ransomware or blackmail.

- Platforms in the Crosshairs: `Microsoft Windows`
- Threatened Entities: `Windows Users`
- Potential Impact: `Complete takeover of the victim's computer / Domain escalation`
- Risk Level: `Critical`

The group primarily leverages opportunistic-phishing for initial access, exploiting data from social media, past breaches (e.g., databases of email addresses), and corporate websites. There is scant evidence suggesting spear-phishing against specific individuals.

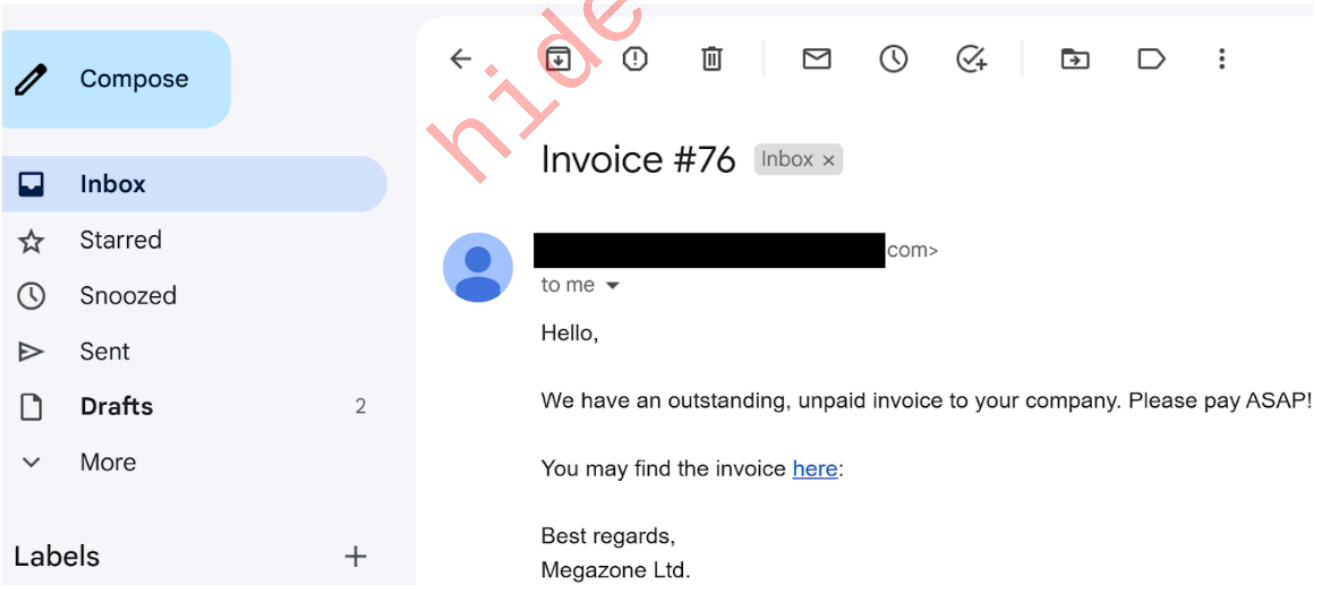
The document compiles all known Tactics Techniques and Procedures (TTPs) and Indicators of Compromise (IOCs) linked to the group, which are currently under continuous refinement. This preliminary sketch is confidential and meant exclusively for our partners, who are strongly advised to conduct scans of their infrastructures to spot potential successful breaches at the earliest possible stage.

In summary, the attack sequence for the initially compromised device can be laid out as follows:



Initial Breach

The phishing email is relatively rudimentary, with the malware posing as an invoice file. Here's an example of an actual phishing email that includes a link leading to a OneNote file:



Our forensic investigation into these attacks revealed that the link directs to a OneNote file, which has consistently been hosted on a file hosting service (e.g., Mega.io or similar platforms).

This OneNote file masquerades as an invoice featuring a 'HIDDEN' button that triggers an embedded batch file. This batch file, in turn, fetches PowerShell scripts, representing stage 0 of the malicious payload.

`RAT Characteristics`

The RAT deployed in these attacks is modular, implying that it can be augmented with an infinite range of capabilities. While only a few features are accessible once the RAT is staged, we have noted the use of tools that capture screen dumps, execute Mimikatz, provide an interactive `CMD shell` on compromised machines, and so forth.

`Persistence`

All persistence mechanisms utilized to date have involved an EXE file deposited on the disk.

`Lateral Movement`

So far, we have identified two distinct methods for lateral movement:

- Leveraging the original, Microsoft-signed PsExec
- Using WinRM

`Indicators of Compromise (IOCs)`

The following provides a comprehensive inventory of all identified IOCs to this point.

OneNote File:

- https://transfer.sh/get/kNxU7/invoice.one
- https://mega.io/dl9o1Dz/invoice.one

Staging Entity (PowerShell Script):

- https://pastebin.com/raw/AvHtdKb2
- https://pastebin.com/raw/gj58DKz

Command and Control (C&C) Nodes:

- 91.90.213.14:443
- 103.248.70.64:443
- 141.98.6.59:443

Cryptographic Hashes of Involved Files (SHA256):

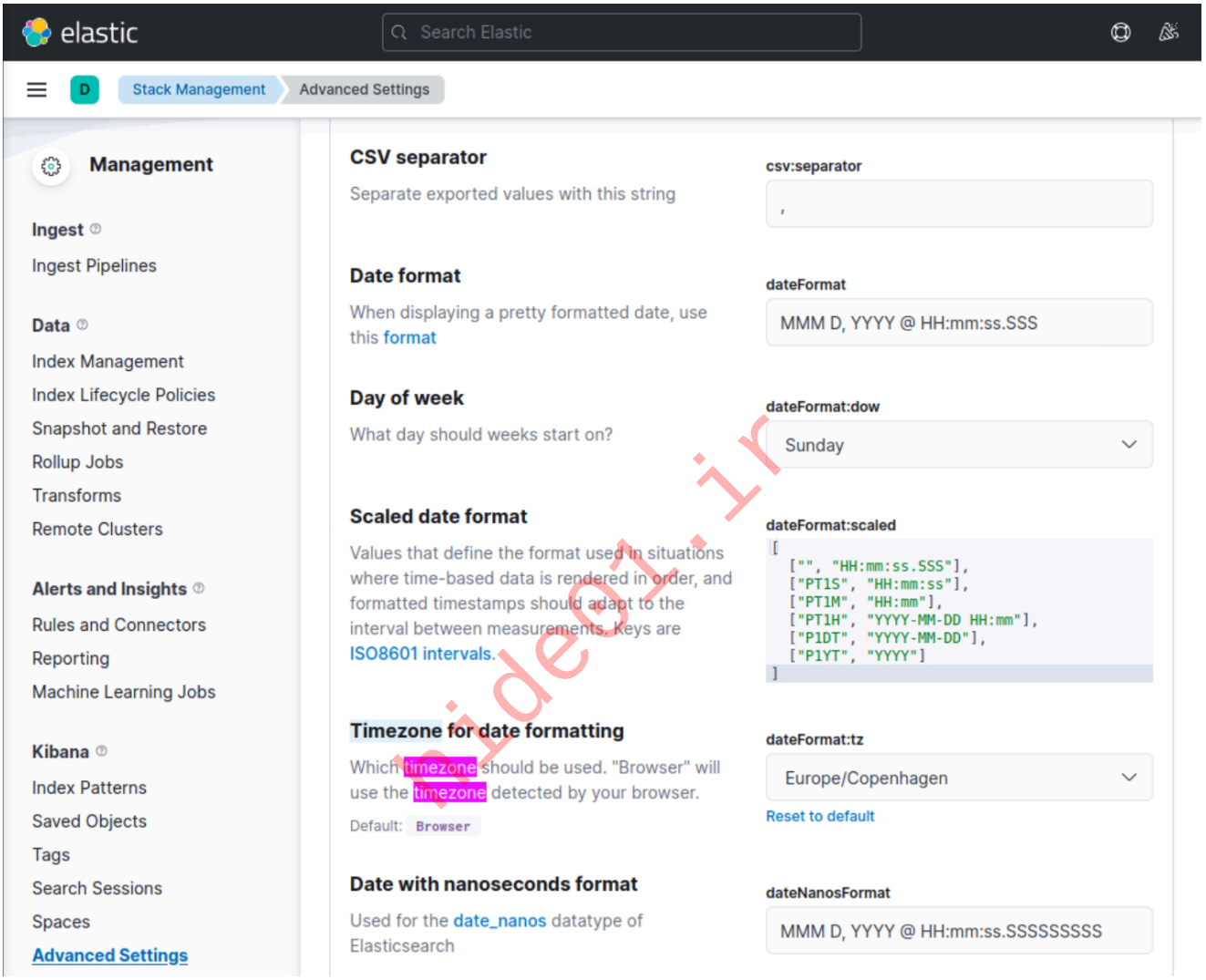
- 226A723FFB4A91D9950A8B266167C5B354AB0DB1DC225578494917FE53867EF2
- C346077DAD0342592DB753FE2AB36D2F9F1C76E55CF8556FE5CDA92897E99C7E
- 018D37CBD3878258C29DB3BC3F2988B6AE688843801B9ABC28E6151141AB66D4

Hunting For Stuxbot With The Elastic Stack

Navigate to the bottom of this section and click on `Click here to spawn the target system!`

Now, navigate to `http://[Target IP]:5601`, click on the side navigation toggle, and click on "Discover". Then, click on the calendar icon, specify "last 15 years", and click on "Apply".

Please also specify a `Europe/Copenhagen` timezone, through the following link `http://[Target IP]:5601/app/management/kibana/settings`.



`The Available Data`

The cybersecurity strategy implemented is predicated on the utilization of the Elastic stack as a SIEM solution. Through the "Discover" functionality we can see logs from multiple sources. These sources include:

- `Windows audit logs` (categorized under the index pattern windows*)
- `System Monitor (Sysmon) logs` (also falling under the index pattern windows*, more about Sysmon here)
- `PowerShell logs` (indexed under windows* as well, more about PowerShell logs here)

- `Zeek logs`, a network security monitoring tool (classified under the index pattern zeek*)

Our available threat intelligence stems from March 2023, hence it's imperative that our Kibana setup scans logs dating back at least to this time frame. Our "windows" index contains around 118,975 logs, while the "zeek" index houses approximately 332,261 logs.

`The Environment`

Our organization is relatively small, with about 200 employees primarily engaged in online marketing activities, thus our IT resource requirement is minimal. Office applications are the primary software in use, with Gmail serving as our standard email provider, accessed through a web browser. Microsoft Edge is the default browser on our company laptops. Remote technical support is provided through TeamViewer, and all our company devices are managed via Active Directory Group Policy Objects (GPOs). We're considering a transition to Microsoft Intune for endpoint management as part of an upcoming upgrade from Windows 10 to Windows 11.

`The Task`

Our task centers around a threat intelligence report concerning a malicious software known as "Stuxbot". We're expected to use the provided Indicators of Compromise (IOCs) to investigate whether there are any signs of compromise in our organization.

`The Hunt`

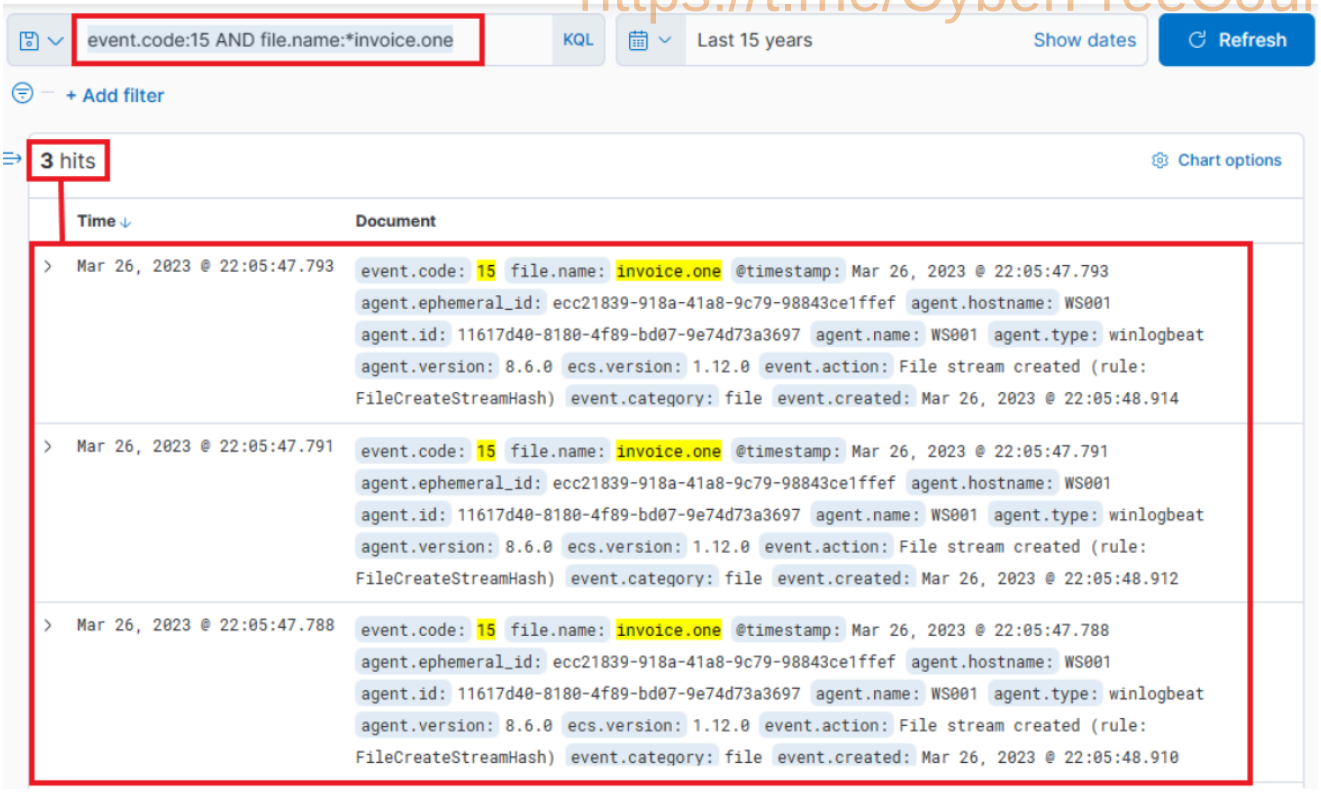
The sequence of hunting activities is premised on the hypothesis of a successful phishing email delivering a malicious OneNote file. If our hypothesis had been the successful execution of a binary with a hash matching one from the threat intelligence report, we would have undertaken a different sequence of activities.

The report indicates that initial compromises all took place via "invoice.one" files. Despite this, we must continue to conduct searches on other IOCs as the threat actors may have introduced different delivery techniques between the time the report was created and the present. Back to the "invoice.one" files, a comprehensive search can be initiated based on Sysmon Event ID 15 (FileCreateStreamHash), which represents a browser file download event. We're assuming that a potentially malicious OneNote file was downloaded from Gmail, our organization's email provider.

Our search query should be the following.

Related fields: winlog.event_id or event.code and file.name

```
event.code:15 AND file.name:*invoice.one
```



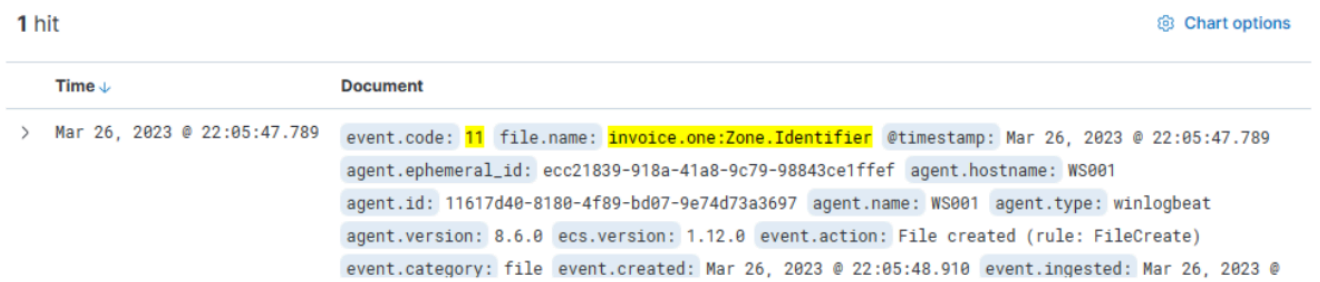
While this development could imply serious implications, it's not yet confirmed if this file is the same one mentioned in the report. Further, signs of execution have not been probed. If we extend the event log to display its complete content, it'll reveal that MSEdge was the application (as indicated by `process.name` or `process.executable`) used to download the file, which was stored in the Downloads folder of an employee named Bob.

The timestamp to note is: `March 26, 2023 @ 22:05:47`

We can corroborate this information by examining Sysmon Event ID 11 (File create) and the "invoice.one" file name. This method is especially effective when browsers aren't involved in the file download process. The query is similar to the previous one, but the asterisk is at the end as the file name includes only the filename with an additional Zone Identifier, likely indicating that the file originated from the internet.

Related fields: winlog.event_id or event.code and file.name

```
event.code:11 AND file.name:invoice.one*
```



It's relatively easy to deduce that the machine which reported the "invoice.one" file has the hostname WS001 (check the `host.hostname` or `host.name` fields of the Sysmon Event ID 11 event we were just looking at) and an IP address of 192.168.28.130, which can be confirmed by checking any network connection event (Sysmon Event ID 3) from this machine (execute the following query `event.code:3 AND host.hostname:WS001` and check the `source.ip` field).

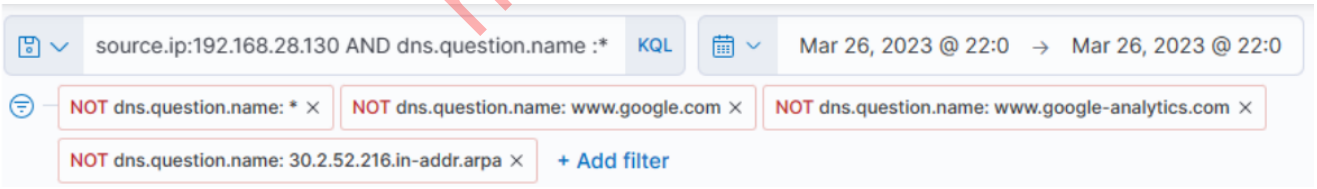
If we inspect network connections leveraging Sysmon Event ID 3 (Network connection) around the time this file was downloaded, we'll find that Sysmon has no entries. This is a common configuration to avoid capturing network connections created by browsers, which could lead to an overwhelming volume of logs, particularly those related to our email provider.

This is where Zeek logs prove invaluable. We should filter and examine the DNS queries that Zeek has captured from WS001 during the interval from `22:05:00` to `22:05:48`, when the file was downloaded.

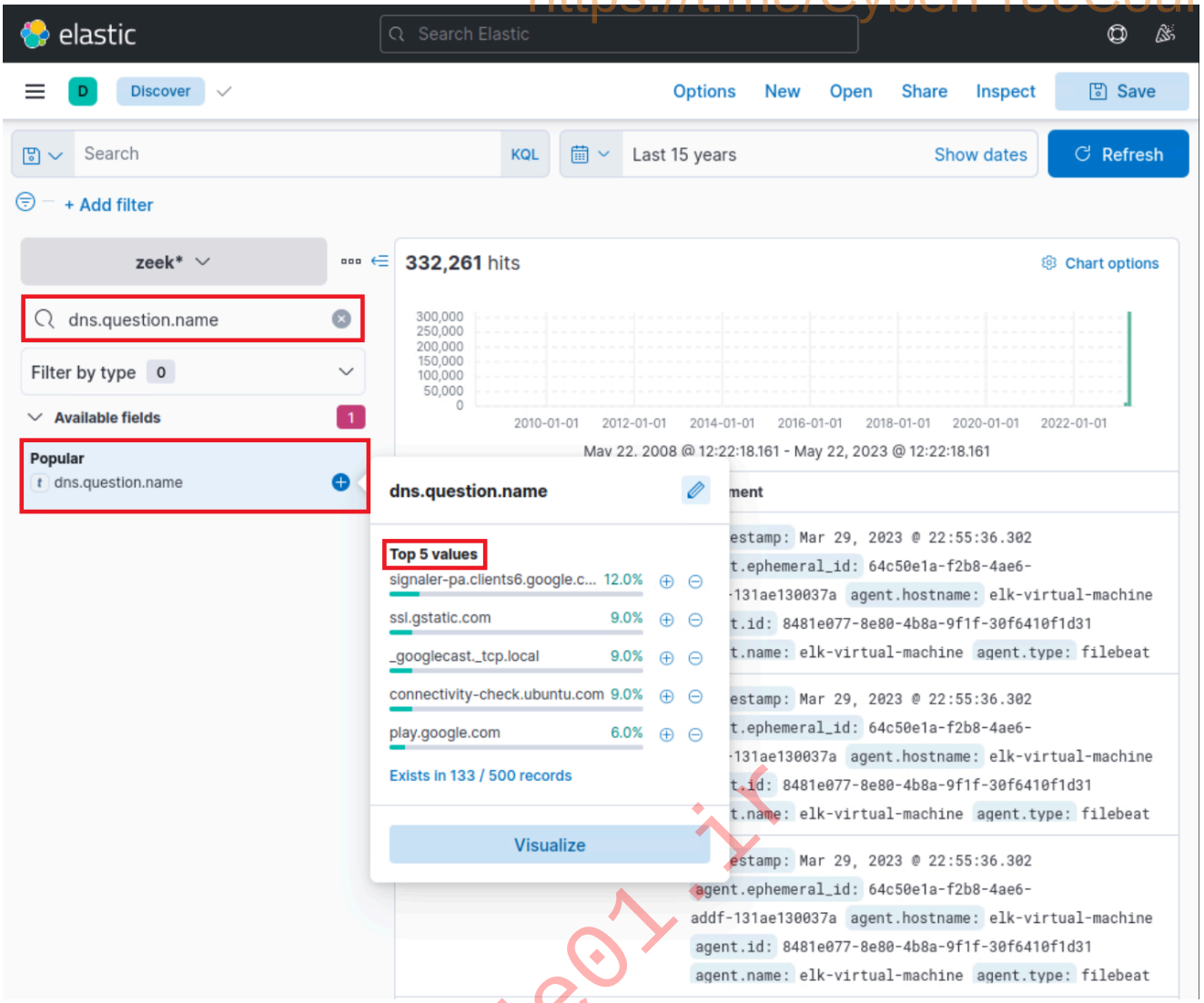
Our `Zeek query` will search for a source IP matching 192.168.28.130, and since we're querying about DNS queries, we'll only pick logs that have something in the `dns.question.name` field. Note that this will return a lot of common noise, like google.com, etc., so it's necessary to filter that out. Here's the query and some filters.

Related fields: source.ip and dns.question.name

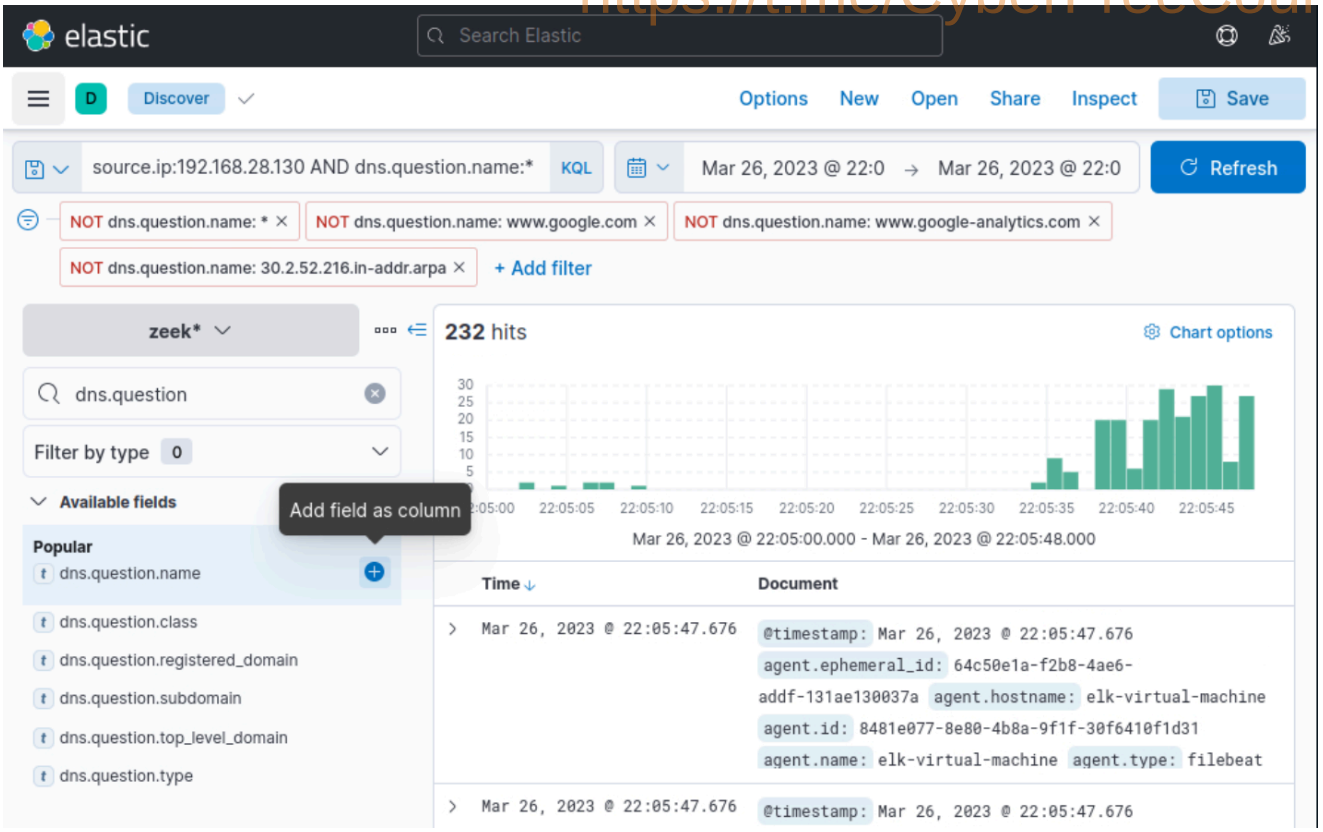
```
source.ip:192.168.28.130 AND dns.question.name:*
```



We can easily identify major sources of noise by looking at the most common values that Kibana has detected (click on a field as follows), and then apply a filter on the known noisy ones.



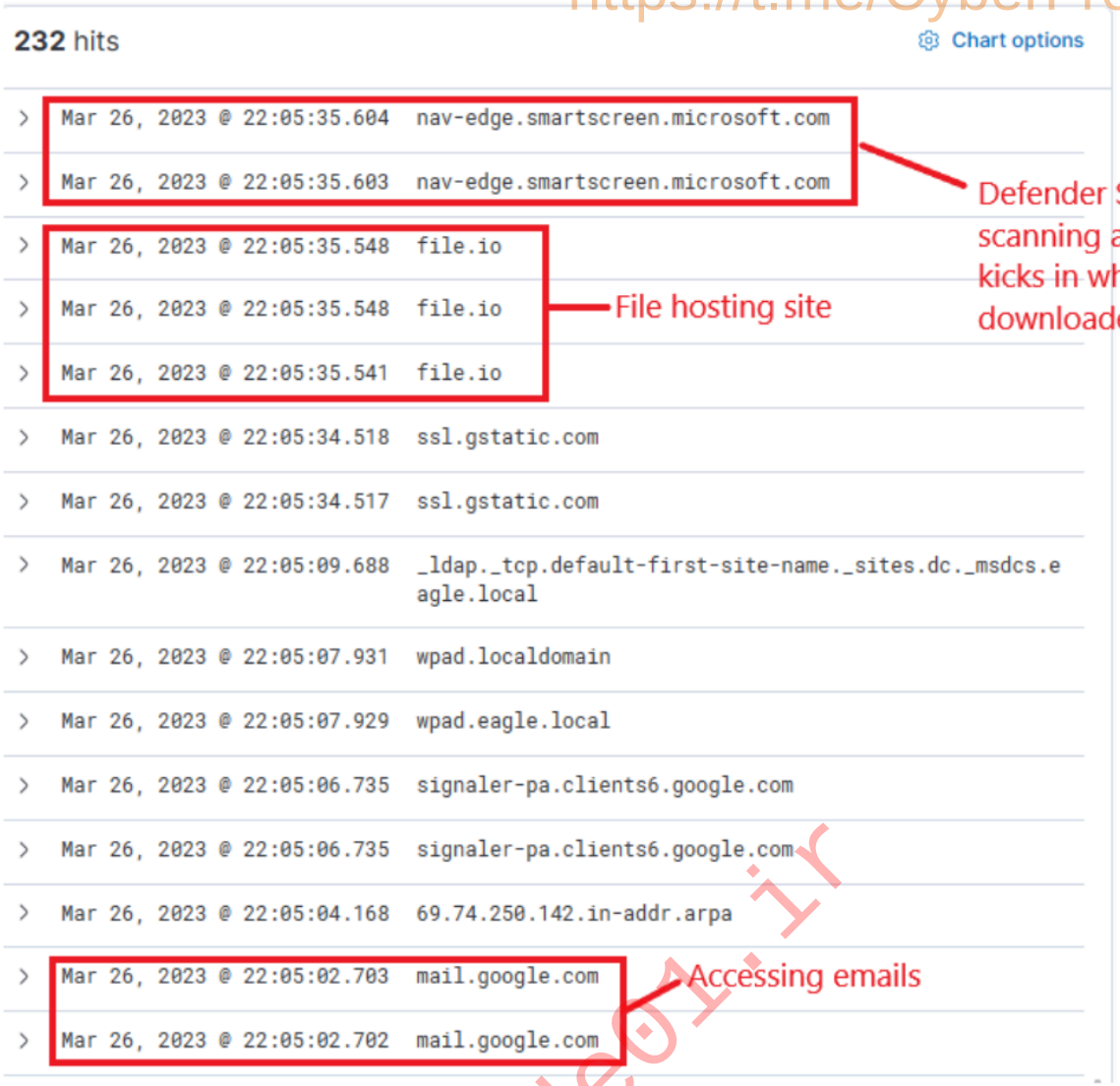
As part of our search process, since we're interested in DNS names, we'd like to display only the `dns.question.name` field in the result table. Please note the specified time `March 26th 2023 @ 22:05:00` to `March 26th 2023 @ 22:05:48`.



232 hits

Time	dns.question.name
Mar 26, 2023 @ 22:05:47.676	ad-delivery.net
Mar 26, 2023 @ 22:05:47.676	ad-delivery.net
Mar 26, 2023 @ 22:05:47.553	crt.usertrust.com
Mar 26, 2023 @ 22:05:47.237	track.venatusmedia.com
Mar 26, 2023 @ 22:05:47.236	track.venatusmedia.com
Mar 26, 2023 @ 22:05:47.193	52.208.241.202.in-addr.arpa

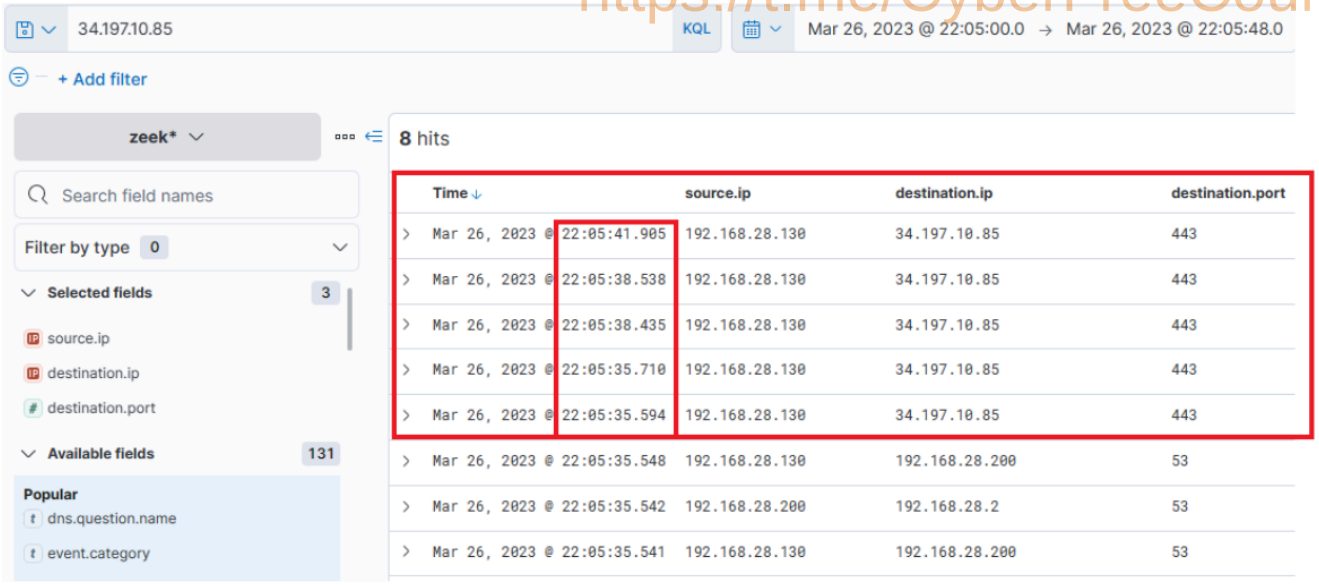
Scrolling down the table of entries, we observe the following activities.



From this data, we infer that the user accessed Google Mail, followed by interaction with "file.io", a known hosting provider. Subsequently, Microsoft Defender SmartScreen initiated a file scan, typically triggered when a file is downloaded via Microsoft Edge. Expanding the log entry for file.io reveals the returned IP addresses (`dns.answers.data` or `dns.resolved_ip` or `zeek.dns.answers` fields) as follows.

`34.197.10.85`, `3.213.216.16`

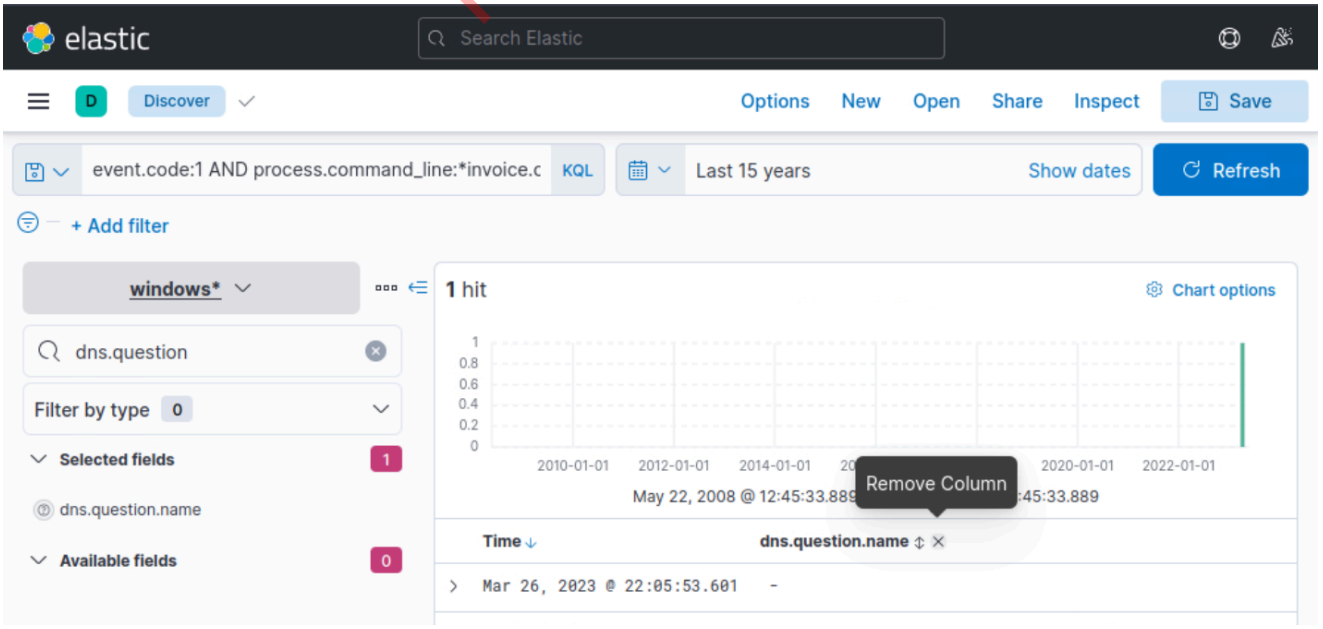
Now, if we run a search for any connections to these IP addresses during the same timeframe as the DNS query, it leads to the following findings.



This information corroborates that a user, Bob, successfully downloaded the file "invoice.one" from the hosting provider "file.io".

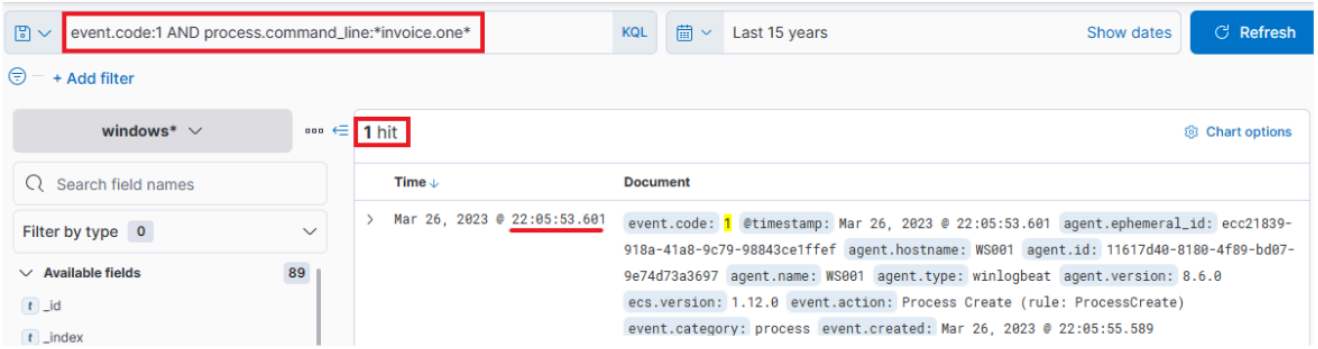
At this juncture, we have two choices: we can either cross-reference the data with the Threat Intel report to identify overlapping information within our environment, or we can conduct an Incident Response (IR)-like investigation to trace the sequence of events post the OneNote file download. We choose to proceed with the latter approach, tracking the subsequent activities.

Hypothetically, if "invoice.one" was accessed, it would be opened with the OneNote application. So, the following query will flag the event, if it transpired. **Note**: The time frame we specified previously should be removed, setting it to, say, 15 years again. The `dns.question.name` column should also be removed.



Related fields: winlog.event_id or event.code and process.command_line

```
event.code:1 AND process.command_line:*invoice.one*
```



Indeed, we find that the OneNote file was accessed shortly after its download, with a delay of roughly 6 seconds. Now, with OneNote.exe in operation and the file open, we can speculate that it either contains a malicious link or a malevolent file attachment. In either case, OneNote.exe will initiate either a browser or a malicious file. Therefore, we should scrutinize any new processes where OneNote.exe is the parent process. The corresponding query is the following. Sysmon Event ID 1 (Process creation) is utilized.

Related fields: winlog.event_id or event.code and process.parent.name

```
event.code:1 AND process.parent.name:"ONENOTE.EXE"
```


The results of this query present three hits. However, one of these (the bottom one) falls outside the relevant time frame and can be dismissed. Evaluating the other two results:

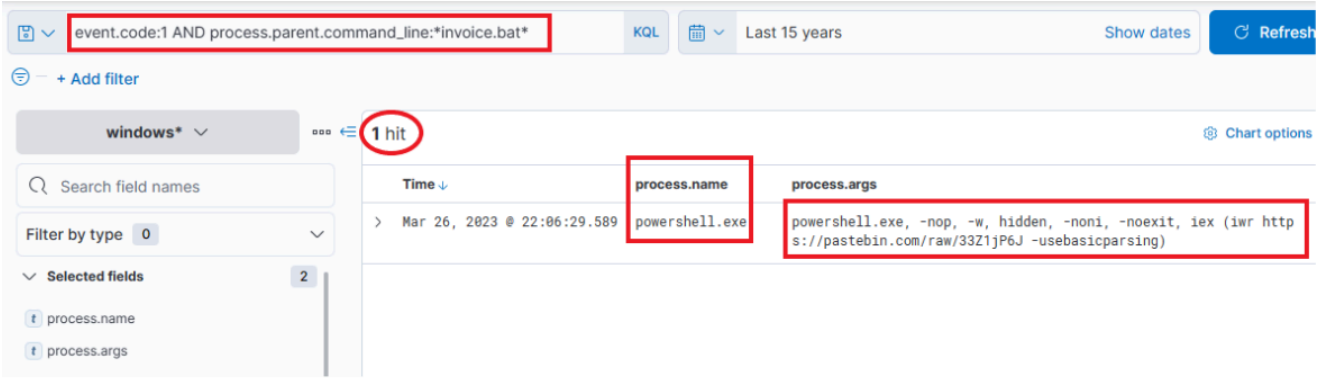
- The middle entry documents (when expanded) a new process, OneNoteM.exe, which is a component of OneNote and assists in launching files.
- The top entry reveals "cmd.exe" in operation, executing a file named "invoice.bat". Here is the view upon expanding the log.

4	t process.command_line	C:\WINDOWS\system32\cmd.exe /c ""C:\Users\bob\AppData\Local\Temp\OneNote\16.0\Exported\{EC284AA9-1F31-4DC4-B3C5-3EEE8137EBC3}\NT\0\invoice.bat" "
	t process.entity_id	{3f3a32cd-a5c4-6420-e101-000000001a00}
3	t process.executable	C:\Windows\System32\cmd.exe
	t process.hash.md5	8a2122e8162dbef04694b9c3e0b6cdee
	t process.hash.sha256	b99d61d874728edc0918ca0eb10eab93d381e7367e377406e65963366c874450
	t process.name	cmd.exe
	t process.parent.args	C:\Program Files\Microsoft Office\Root\Office16\ONENOTE.EXE, C:\Users\bob\Downloads\invoice.one
	# process.parent.args_count	2
2	t process.parent.command_line	"C:\Program Files\Microsoft Office\Root\Office16\ONENOTE.EXE" "C:\Users\bob\Downloads\invoice.one"
	t process.parent.entity_id	{3f3a32cd-a5a1-6420-cc01-000000001a00}
1	t process.parent.executable	C:\Program Files\Microsoft Office\root\Office16\ONENOTE.EXE

Now we can establish a connection between "OneNote.exe", the suspicious "invoice.one", and the execution of "cmd.exe" that initiates "invoice.bat" from a temporary location (highly likely due to its attachment inside the OneNote file). The question now is, has this batch script instigated anything else? Let's search if a parent process with a command line argument pointing to the batch file has spawned any child processes with the following query.

Related fields: winlog.event_id or event.code and process.parent.command_line

```
event.code:1 AND process.parent.command_line:*invoice.bat*
```



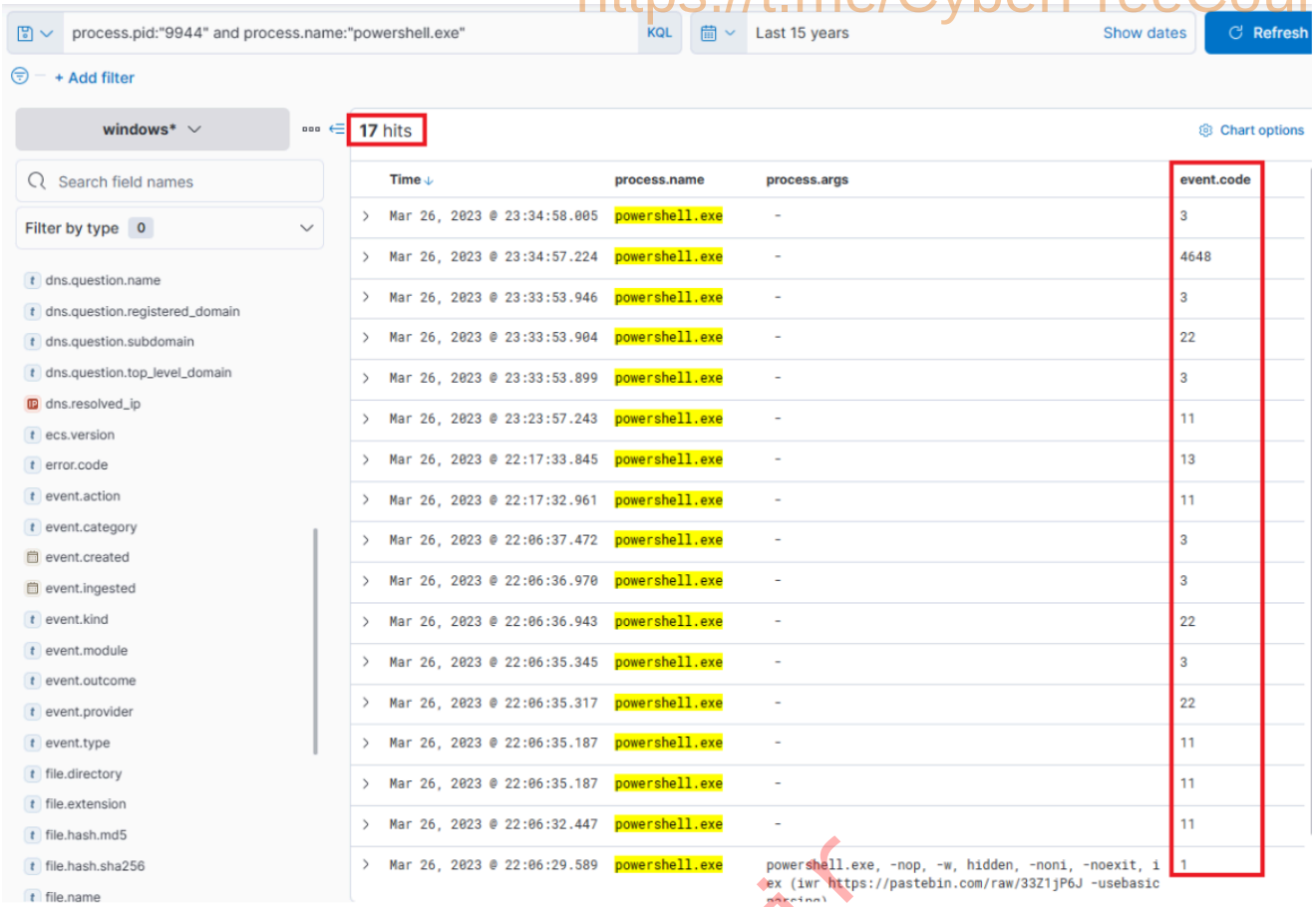
This query returns a single result: the initiation of PowerShell, and the arguments passed to it appear conspicuously suspicious (note that we have added `process.name`, `process.args`, and `process.pid` as columns)! A command to download and execute content from Pastebin, an open text hosting provider! We can try to access and see if the content, which the script attempted to download, is still available (by default, it won't expire!).

Indeed, it is! This is referred to in the Threat Intelligence report, stating that a PowerShell Script from Pastebin was downloaded.

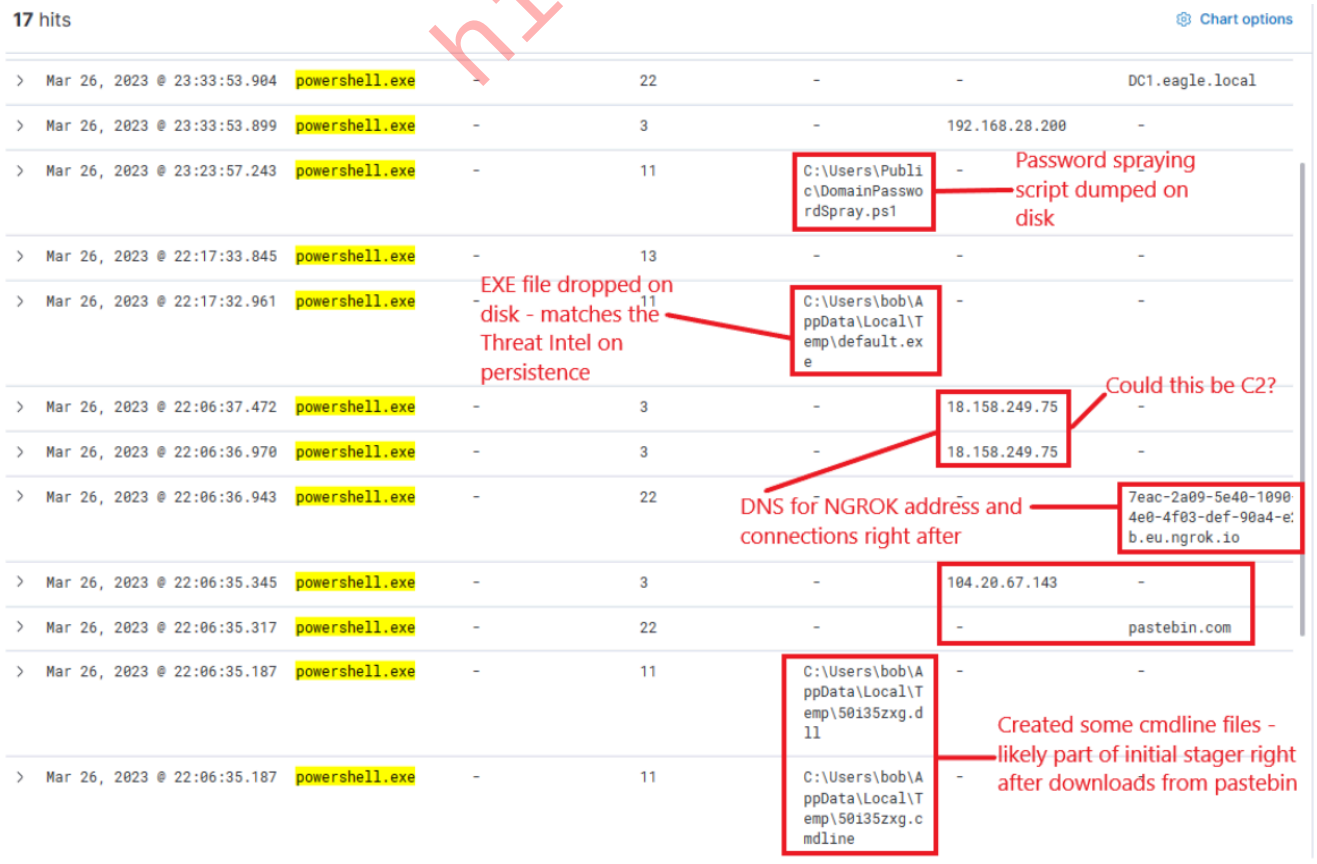
To figure out what PowerShell did, we can filter based on the process ID and name to get an overview of activities. Note that we have added the `event.code` field as a column.

Related fields: process.pid and process.name

```
process.pid:"9944" and process.name:"powershell.exe"
```



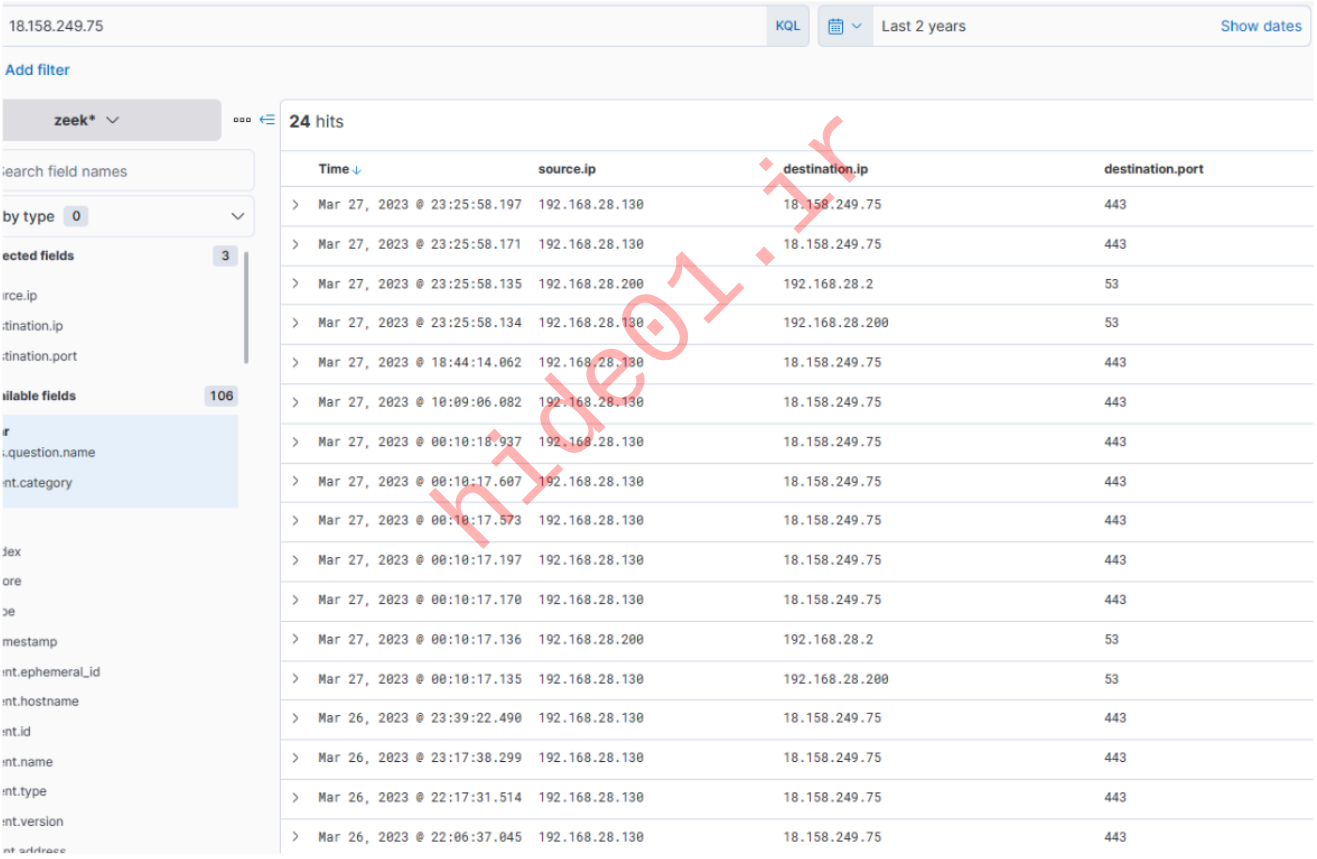
Immediately, we can observe intriguing output indicating file creation, attempted network connections, and some DNS resolutions leverarging Sysmon Event ID 22 (DNSEvent). By adding some additional informative fields (`file.path` , `dns.question.name` , and `destination.ip`) as columns to that view, we can expand it.



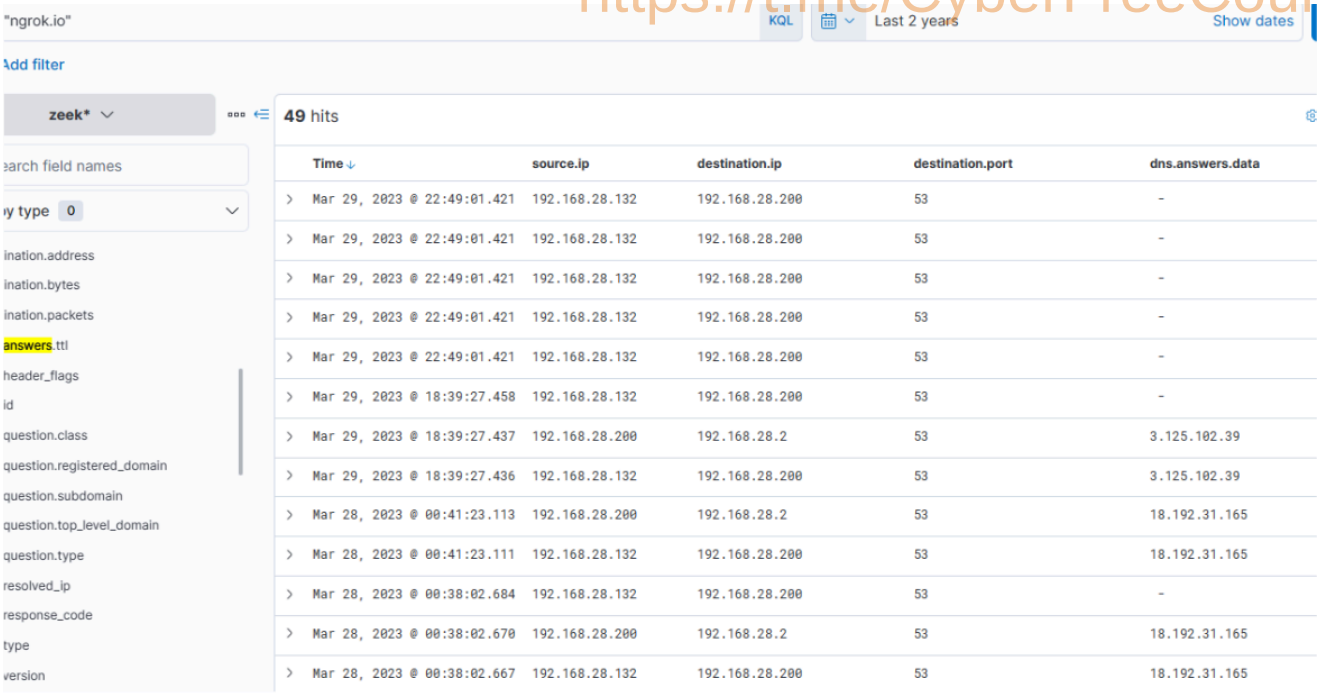
Now, this presents us with rich data on the activities. Ngrok was likely employed as C2 (to mask malicious traffic to a known domain). If we examine the connections above the DNS resolution for Ngrok, it points to the destination IP Address 443, implying that the traffic was encrypted.

The dropped EXE is likely intended for persistence. Its distinctive name should facilitate determining whether it was ever executed. It's important to note the timestamps – there is some time lapse between different activities, suggesting it's less likely to have been scripted but perhaps an actual human interaction took place (unless random sleep occurred between the executed actions). The final actions that this process points to are a DNS query for DC1 and connections to it.

Let's review Zeek data for information on the destination IP address `18.158.249.75` that we just discovered. Note that the `source.ip`, `destination.ip`, and `destination.port` fields were added as columns.



Intriguingly, the activity seems to have extended into the subsequent day. The reason for the termination of the activity is unclear... Was there a change in C2 IP? Or did the attack simply halt? Upon inspecting DNS queries for "ngrok.io", we find that the returned IP (`dns.answers.data`) has indeed altered. Note that the `dns.answers.data` field was added as a column.



The newly discovered IP also indicates that connections continued consistently over the following days.

3.125.102.39

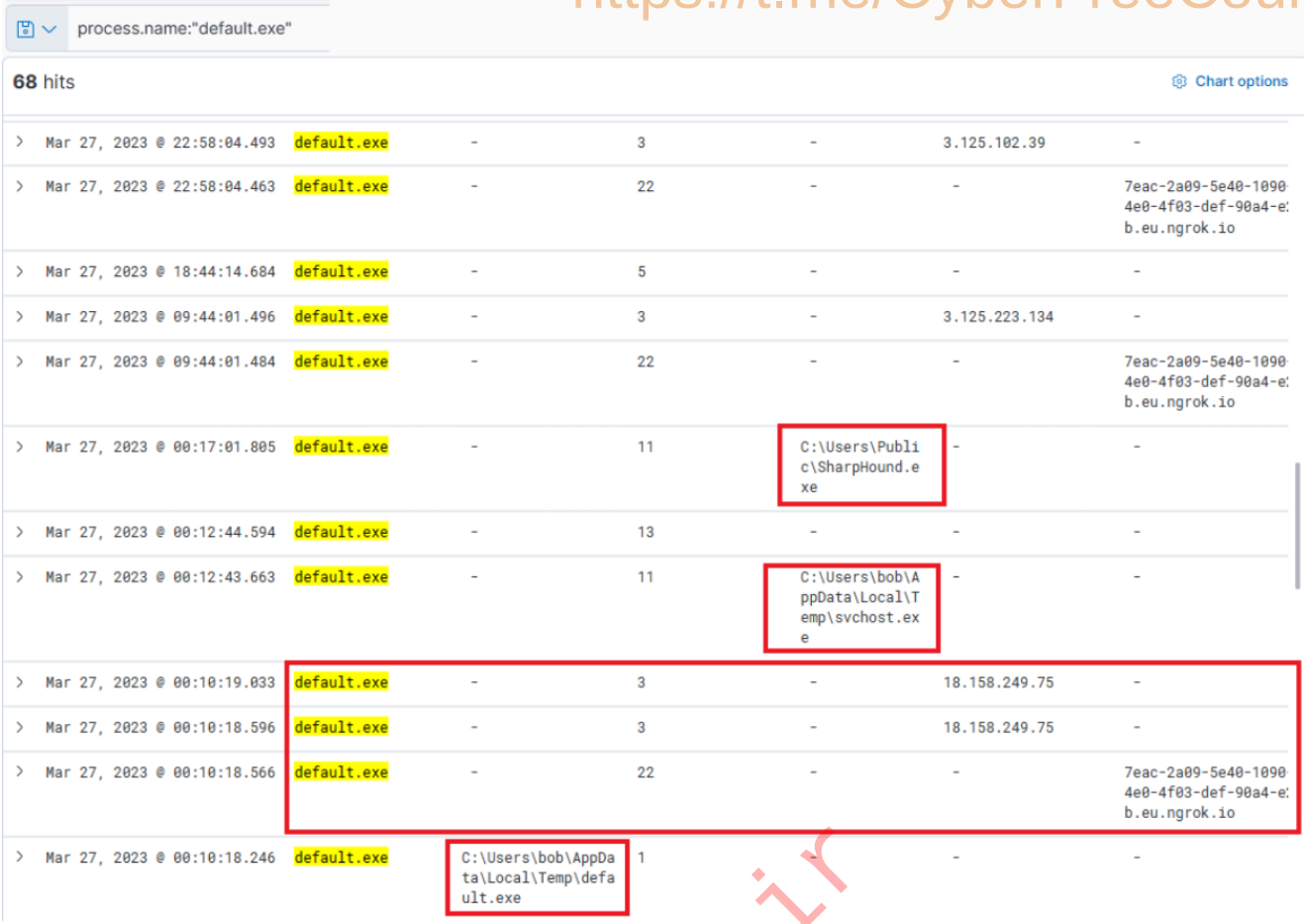
29 hits

Time ↓	source.ip	destination.ip	destination.port
Mar 29, 2023 @ 18:39:27.507	192.168.28.132	3.125.102.39	443
Mar 29, 2023 @ 18:39:27.488	192.168.28.132	3.125.102.39	443
Mar 29, 2023 @ 18:39:27.437	192.168.28.200	192.168.28.2	53
Mar 29, 2023 @ 18:39:27.436	192.168.28.132	192.168.28.200	53
Mar 28, 2023 @ 00:57:23.900	192.168.28.130	3.125.102.39	443
Mar 28, 2023 @ 00:34:35.855	192.168.28.132	3.125.102.39	443
Mar 28, 2023 @ 00:34:11.644	192.168.28.132	3.125.102.39	443
Mar 28, 2023 @ 00:18:14.780	192.168.28.132	3.125.102.39	443
Mar 28, 2023 @ 00:18:13.374	192.168.28.132	3.125.102.39	443
Mar 28, 2023 @ 00:18:13.353	192.168.28.132	3.125.102.39	443
Mar 28, 2023 @ 00:18:13.241	192.168.28.132	3.125.102.39	443
Mar 28, 2023 @ 00:18:12.783	192.168.28.132	3.125.102.39	443
Mar 28, 2023 @ 00:18:12.723	192.168.28.132	3.125.102.39	443
Mar 28, 2023 @ 00:18:12.641	192.168.28.200	192.168.28.2	53
Mar 28, 2023 @ 00:18:12.640	192.168.28.132	192.168.28.200	53
Mar 28, 2023 @ 00:11:27.043	192.168.28.130	3.125.102.39	443
Mar 27, 2023 @ 23:23:31.933	192.168.28.130	3.125.102.39	443

Thus, it's apparent that there is sustained network activity, and we can deduce that the C2 has been accessed continually. Now, as for the earlier uploaded executable file "default.exe" – did that ever execute? By probing the Sysmon logs for a process with that name, we can ascertain this. Note that the `process.name`, `process.args`, `event.code`, `file.path`, `destination.ip`, and `dns.question.name` fields were added as columns.

Related field: process.name

```
process.name:"default.exe"
```



Indeed, it has been executed – we can instantly discern that the executable initiated DNS queries for Ngrok and established connections with the C2 IP addresses. It also uploaded two files "svchost.exe" and "SharpHound.exe". SharpHound is a recognized tool for diagramming Active Directory and identifying attack paths for escalation. As for svchost.exe, we're unsure – is it another malicious agent? The name implies it attempts to mimic the legitimate svchost file, which is part of the Windows Operating System.

If we scroll up there's further activity from this executable, including the uploading of "payload.exe", a VBS file, and repeated uploads of "svchost.exe".

At this juncture, we're left with one question: did SharpHound execute? Did the attacker acquire information about Active Directory? We can investigate this with the following query (since it was an on-disk executable file).

Related field: process.name

```
process.name:"SharpHound.exe"
```

4 hits

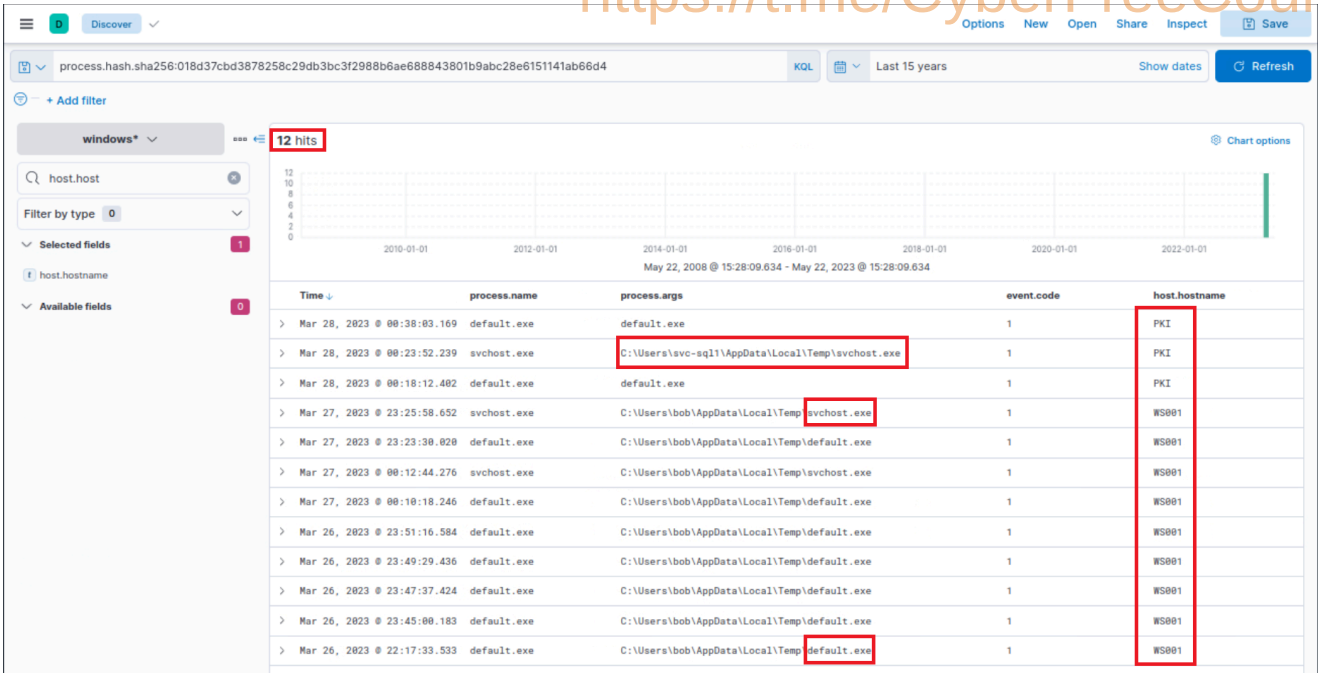
	Time ↓	process.name	process.args
>	Mar 27, 2023 @ 00:19:30.147	SharpHound.exe	-
>	Mar 27, 2023 @ 00:19:30.119	SharpHound.exe	Sharphound.exe, -c, all
>	Mar 27, 2023 @ 00:17:58.409	SharpHound.exe	-
>	Mar 27, 2023 @ 00:17:58.000	SharpHound.exe	sharphound.exe, -collectionmethod, all

Indeed, the tool appears to have been executed twice, roughly 2 minutes apart from each other.

It's vital to note that Sysmon has flagged "default.exe" with a file hash (`process.hash.sha256` field) that aligns with one found in the Threat Intel report. This leads us to question whether this executable has been detected on other devices within the environment. Let's conduct a broad search. Note that the `host.hostname` field was added as a column.

Related field: process.hash.sha256

```
process.hash.sha256:018d37cbd3878258c29db3bc3f2988b6ae688843801b9abc28e6151141ab66d4
```



Files with this hash value have been found on WS001 and PKI, indicating that the attacker has also breached the PKI server at a minimum. It also appears that a backdoor file has been placed under the profile of user "svc-sql1", suggesting that this user's account is likely compromised.

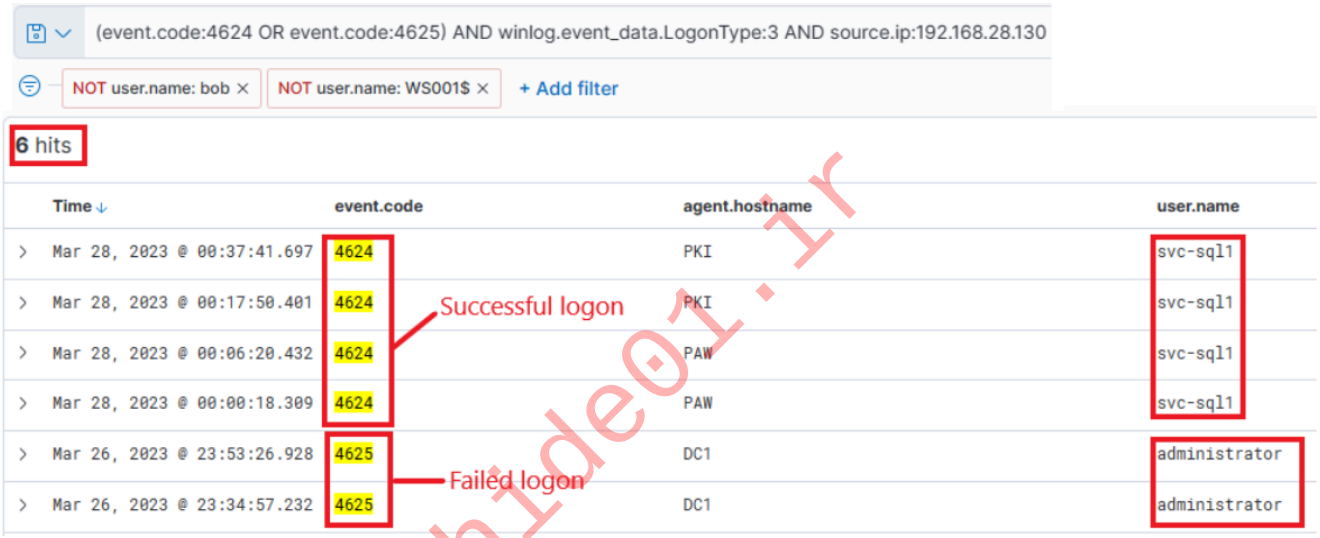
Expanding the first instance of "default.exe" execution on PKI, we notice that the parent process was "PSEXESVC", a component of PSExec from SysInternals – a tool often used for executing commands remotely, frequently utilized for lateral movement in Active Directory breaches.

Further down the same log, we notice "svc-sql1" in the `user.name` field, thereby confirming the compromise of this user.

How was the password of "svc-sql1" compromised? The only plausible explanation from the available data so far is potentially the earlier uploaded PowerShell script, seemingly designed for Password Bruteforcing. We know that this was uploaded on WS001, so we can check for any successful or failed password attempts from that machine, excluding those for Bob, the user of that machine (and the machine itself).

Related fields: winlog.event_id or event.code, winlog.event_data.LogonType, and source.ip

```
(event.code:4624 OR event.code:4625) AND winlog.event_data.LogonType:3 AND
source.ip:192.168.28.130
```



The results are quite intriguing – two failed attempts for the administrator account, roughly around the time when the initial suspicious activity was detected. Subsequently, there were numerous successful logon attempts for "svc-sql1". It appears they attempted to crack the administrator's password but failed. However, two days later on the 28th, we observe successful attempts with svc-sql1.

At this stage, we have amassed a significant amount of information to present and initiate a comprehensive incident response, in accordance with company policies.

Please allow 3-5 minutes for Kibana to become available after spawning the target of the questions below.

Skills Assessment

Hunting For Stuxbot (Round 2)

Recently uncovered details shed light on the operational strategy of Stuxbot's newest iteration.

1. The newest iterations of Stuxbot are exploiting the `C:\Users\Public` directory as a conduit for deploying supplementary utilities.
2. The newest iterations of Stuxbot are utilizing registry run keys as a mechanism to ensure their sustained presence within the infected system.
3. The newest iterations of Stuxbot are utilizing PowerShell Remoting for lateral movement within the network and to gain access to domain controllers.

The Available Data

The cybersecurity strategy implemented is predicated on the utilization of the Elastic stack as a SIEM solution. Through the "Discover" functionality we can see logs from multiple sources. These sources include:

- `Windows audit logs` (categorized under the index pattern windows*)
- `System Monitor (Sysmon) logs` (also falling under the index pattern windows*, more about Sysmon here)
- `PowerShell logs` (indexed under windows* as well, more about PowerShell logs here)
- `Zeek logs`, a network security monitoring tool (classified under the index pattern zeek*)

The Tasks

Navigate to the bottom of this section and click on `Click here to spawn the target system!`

Now, navigate to `http://[Target IP]:5601`, click on the side navigation toggle, and click on "Discover". Then, click on the calendar icon, specify "last 15 years", and click on "Apply".

`Hunt 1`: Create a KQL query to hunt for "Lateral Tool Transfer" to `C:\Users\Public`. Enter the content of the `user.name` field in the document that is related to a transferred tool that starts with "r" as your answer.

`Hunt 2`: Create a KQL query to hunt for "Boot or Logon Autostart Execution: Registry Run Keys / Startup Folder". Enter the content of the `registry.value` field in the document that is related to the first registry-based persistence action as your answer.

`Hunt 3`: Create a KQL query to hunt for "PowerShell Remoting for Lateral Movement". Enter the content of the `winlog.user.name` field in the document that is related to

PowerShell remoting-based lateral movement towards DC1.