Trellix

WHITE PAPER

Threat Hunting and Detection Engineering:

A Proactive Approach to Cyber Defense

Executive Summary

As cyber threats grow more sophisticated, organizations can no longer rely on reactive security measures alone. Threat intelligence (TI) plays a critical role in proactive defense by providing insights into emerging adversary tactics, enabling security teams to anticipate and hunt for threats before they escalate. Threat hunting moves beyond traditional alert-driven investigations, using hypothesis-driven approaches to uncover stealthy attacks, while detection engineering ensures security controls evolve to detect the latest attack techniques effectively.

This white paper provides a comprehensive framework for integrating threat intelligence, proactive hunting methodologies, and advanced detection engineering into modern security operations. Readers will gain practical insights into building a hypothesis-driven hunting model, leveraging diverse data sources, automating detection logic, and continuously refining security measures based on real-world threats.

Designed for SOC analysts, threat hunters, detection engineers, and security leaders, this paper serves as a guide to enhancing threat visibility, reducing attacker dwell time, and strengthening cyber resilience. By adopting these principles, organizations can shift from reactive defense to a proactive, intelligence-driven security strategy capable of staying ahead of evolving threats.

From Reactive to Proactive

Threat hunting is a cornerstone of a modern cybersecurity strategy, designed to close gaps in traditional automated defenses by leveraging human ingenuity, intuition, and in-depth knowledge of threat behavior. The field is characterized by its dynamic nature, where experienced analysts delve deep into network activity to uncover latent threats that often evade automated detection systems.

First, it's important that we shift our thinking from a reactive to a proactive mindset. Don't wait for an alert from your security controls; instead, actively delve into the telemetry and uncover abnormal activity.

The challenge we face necessitates a departure from conventional, highly optimized standard detections. We must instead adopt a creative approach that combines hypothesis-driven hunting questions to identify potential "leads" or vulnerabilities.

Why do we need threat hunting?

Today's threats are far more sophisticated than ever, taking advantage of legitimate applications and impersonating real users. For these reasons alone, we cannot address everything through detection rules.

What you knew about your organization's threat profile yesterday might not be true today. Hence you need to move fast, and use the experimental approach, run with hypothesis, and ask a lot of "what if?" questions. Not everything will be tedious. As your threat hunting practice matures so will the efficacy of tools to collect data and run hunting queries. Oftentimes simple analytics on top of logs can give a Hunter a good start, but the methodology part is reserved for next chapters of this article.

Understanding Threat Hunting

Definition and purpose of threat hunting

At its core, threat hunting is the pursuit of anomalies and hidden threats within an organization's infrastructure, aiming to identify and neutralize malicious activities that automated tools, such as SIEMs and endpoint detection and response (EDR) solutions, might miss. This process is highly dynamic, utilizing not only tools and technology but also the unique insights and analytical thinking of threat hunters, who leverage threat intelligence and contextual expertise.

Threat hunting is more than just discovering a novel threat; it's about detection refinement and threat intelligence enhancement. By iterating through findings and validating suspicions, threat hunters contribute to improving automated systems by feeding them with new Indicators of Compromise (IOCs) and patterns of behavior that fortify future detection capabilities.

Benefits of proactive threat hunting

- **Early detection:** By uncovering stealthy attacks and surreptitious attackers, threat hunting can prevent damage before it happens, reducing impact and subsequent costs associated with an incident.
- **Learning and adapting from threat actor behavior:** Hunters' deep dives into adversarial behavior contribute to improved threat intelligence and predictive capabilities, enhancing overall security posture.
- **Improvement of incident response:** Hunting can reveal new, actionable insights, making incident response teams more prepared and informed.
- **Discovering what is normal in your environments:** Hunters' deep dives into available telemetry can contribute to raising awareness and creating a baseline of what is normal in the environment. This can improve detection engineering and incident response workflows.

This proactive effort is essential to:

- Reduce dwell time: Shorten the period adversaries remain undetected, minimizing potential damage.
- Mitigate damage from unknown threats: Identify new or evolving tactics, techniques, and procedures (TTPs) that adversaries deploy.
- Strengthen organizational resilience: By recognizing and responding to unfamiliar threat vectors, threat hunters contribute to an organization's adaptive defense posture.

Types of threat hunting

Threat hunting is a multifaceted activity, often categorized into several primary types based on the methodology, technology, and sources of inspiration used in hunting missions. Each type offers unique strengths, tailored to different stages of adversary engagement and visibility levels within the network.

Hypothesis-driven hunting

Hypothesis-driven hunting begins with a premise or educated assumption about possible adversarial behavior. This premise can be drawn from:

- Threat intelligence reports: Insights about recently observed tactics or APT groups' targets, methods, or intent.
- Sector-specific attacks: Industries like finance, healthcare, or energy might be targeted by sector-specific threats, prompting hypotheses around sector-specific vulnerabilities or TTPs.
- Organizational changes: Network expansions, software updates, or structural changes can introduce unique vulnerabilities.
- Emerging threats: As vulnerabilities are discovered and disclosed publicly, threat hunters may explore these weaknesses within their networks to preemptively detect exploits.

Example:

A financial organization receives intelligence suggesting a new APT group is using custom PowerShell scripts for data exfiltration. The hypothesis might involve searching for unusual PowerShell activity or command-line patterns associated with credential access and data staging.

Hypothesis-driven hunts are powerful because they encourage creativity, enabling hunters to think like attackers and explore potential attack paths. This approach also builds upon threat intelligence to contextualize and anticipate adversarial behavior, often exposing previously unknown vulnerabilities.

IOC-based hunting

IOC-based hunting is a reactive, evidence-led approach, focusing on the detection of known threat artifacts within the environment. Common indicators of compromise include:

- File hashes: Known malicious file hashes can reveal malware or compromised files.
- IP addresses and domains: Identifying communication with malicious IPs or command-and-control (C2) domains can expose active intrusions.
- Email addresses and URLs: Phishing or spear-phishing attempts often rely on specific email addresses or URLs that are associated with adversary campaigns.

Curated threat intelligence feeds and historical incident data serve as essential resources for IOC-based hunting. By systematically combing through logs, endpoint, and network telemetry for these indicators, threat hunters provide a crucial layer of defense. However, this type of hunting also has limitations:

- Reactive nature: Since IOCs are based on past or known attacks, IOC-based hunting may miss new or unknown threats.
- Reliance on timely threat intelligence: IOCs can rapidly become outdated, as adversaries change tactics and infrastructure.

Despite these limitations, IOC-based hunting remains invaluable for detecting ongoing or historical threats and augmenting threat visibility.

TTP-based hunting (behavioral hunting)

Another advanced hunting approach is TTP-based hunting, which emphasizes the detection of threat actor tactics, techniques, and procedures (TTPs). By understanding how attackers operate, hunters can look beyond specific indicators and identify patterns or sequences of behavior that suggest malicious intent. The MITRE ATT&CK framework is widely used to map and hunt for these TTPs, allowing hunters to recognize techniques commonly employed by adversaries, such as privilege escalation or lateral movement, and act accordingly.

Example:

TTP-based hunting: Identifying lateral movement

In the past, a technology firm initiated a proactive threat-hunting exercise after detecting an unusual spike in failed logins across their domain controllers. Utilizing the MITRE ATT&CK framework, the team focused on identifying TTPs associated with lateral movement, particularly the use of administrative shares and remote services.

Approach:

The team extracted and correlated log data from Windows Event Logs and network monitoring systems. Key events included:

1. Event ID 4624 (Logon Success) to identify successful authentications.
2. Event ID 5140 (Access to Shared Object) to monitor administrative share usage.

Example Detection Logic:

```
Get-EventLog -LogName Security -InstanceId 4624 | Where-Object { $_.Message
-like '*Workstation Name: \\\\*' }
```

During the analysis, the team discovered that a compromised administrative account was actively being used to access multiple systems in rapid succession. Further investigation revealed the use of psexec.exe, a legitimate Microsoft tool often abused by attackers, to facilitate remote command execution across the network. Login activity logs also showed suspicious patterns, such as authentications originating from geographically distant IP addresses that were inconsistent with the organization's usual access locations.

The attackers were employing legitimate administrative tools such as PowerShell and WMI. These tools allowed them to blend into normal operations while carrying out malicious activities. One critical observation was their use of administrative shares, such as \\ADMIN$\, to copy malicious payloads between systems and execute commands remotely.

Case Studies:

Case Study 1

In 2024, a financial institution observed subtle anomalies in its network traffic that evaded existing security measures. To proactively identify potential threats, the organization's cybersecurity team initiated a hypothesis-driven threat hunting mission.

Step 1: Hypothesis formation

The team hypothesized that an APT actor had infiltrated the network and was employing stealthy techniques to evade and maintain persistence with the goal of exfiltrating sensitive data. They speculated that the adversary might be using legitimate administrative tools to blend in with normal operations, a tactic known as "Living Off The Land (LOTL)."

Step 2: Hunting methodology

- Data collection: Aggregated logs from endpoints, servers, and network devices, focusing on processes initiating network connections, especially those associated with administrative tools.
- Behavioral analysis: Utilized User and Entity Behavior Analytics (UEBA) to establish baselines for normal behavior and identify deviations.
- Threat intelligence integration: Correlated findings with threat intelligence feeds to identify known IOCs and TTPs associated with APT groups.

Step 3: Findings

The hunt revealed that the Windows utility rundll32.exe was being used to execute malicious DLLs, a technique aligned with MITRE ATT&CK technique T1218.011.

Further analysis uncovered unauthorized use of WMI for persistence and lateral movement, corresponding to ATT&CK techniques T1047 and T1053.

Step 4: Outcome

The team confirmed the presence of an APT leveraging legitimate tools to conduct malicious activities. The organization implemented stricter monitoring of administrative tool usage, enhanced endpoint detection capabilities, and updated incident response protocols to address such threats.

Step 5: Remediation

Once the threat was confirmed, the cybersecurity team executed a structured response plan to contain and eliminate the APT actor while strengthening defenses to prevent recurrence.

- **Containment actions:** The organization isolated affected systems to prevent further lateral movement. They temporarily restricted the execution of rundll32.exe for non-essential operations and disabled unauthorized WMI persistence while revoking suspicious administrative privileges.
- **Threat eradication:** The cybersecurity team conducted forensic analysis on compromised hosts to identify and remove all malicious DLLs. They also removed unauthorized scheduled tasks and WMI subscriptions used for persistence, patched exploited vulnerabilities, and enforced endpoint hardening measures.
- **Recovery and system restoration:** Critical systems were restored from clean backups, and compromised machines were rebuilt where necessary to ensure complete removal of threats. The team revalidated system integrity through additional threat hunting cycles.
- **Strengthening defenses:** Stricter monitoring of administrative tool usage was implemented via SIEM alerts. The team updated EDR rules to detect suspicious rundll32.exe executions and abnormal WMI activity. Additionally, security awareness training was conducted for IT and SOC teams on stealthy attack techniques.

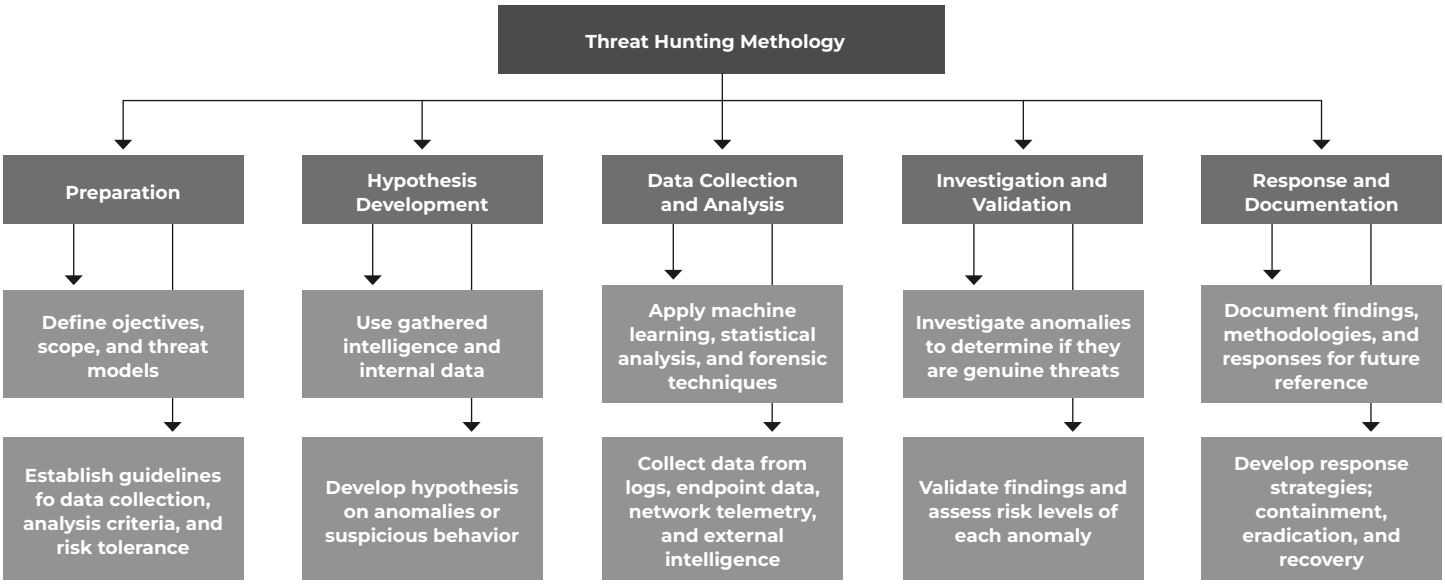

Figure 1: Threat Hunting Methology

Challenges in Threat Hunting

Despite its benefits, threat hunting faces several challenges:

- **Skills gap:** Effective threat hunting requires highly skilled professionals. Whether it is analyzing PCAPs and network flows, or being proficient with logging tools, expertise is in short supply.
- **Standard tools:** Unfortunately there is a complete lack of standardized solutions, tools, technologies, and methodologies across industries and organizations. This requires more from the threat hunter to start being effective.
- **Data overload:** The sheer volume of data can make it difficult to identify relevant information. Closing the skills gap is important.
- **False positives:** Distinguishing between genuine threats and benign anomalies can be challenging. Choose tools that repeatedly demonstrate a low false positive rate.
- **Resource constraints:** Threat hunting can be time-consuming and resource-intensive. Many analysts spend their days simply triaging alerts.
- **Findings dissemination:** Sharing findings to enable prompt action, whether with team members or inputting the findings into existing detection tools.

Best Practices for Effective Threat Hunting

Building a hypothesis-driven hunting model

Effective threat hunting requires a proactive and structured approach to identify and mitigate emerging threats before they can cause harm. A best practice is to adopt a hypothesis-driven hunting model, where hunts begin with well-defined questions or assumptions about potential attacker behaviors. For example, a hunter might explore, "Are we seeing any instances of PowerShell scripts executing encoded commands during non-business hours?" to detect potential living-off-the-land (LOTL) attacks. By focusing on specific threat scenarios, this method ensures hunts are targeted, measurable, and more likely to produce actionable results rather than generating noise.

Leveraging threat intelligence to inform hunts

Threat intelligence plays a pivotal role in refining these hypotheses by providing real-world insights into attacker tactics and campaigns. For instance, if a recent threat report highlights a surge in Cobalt Strike beacon activity targeting financial institutions, a hunter could pivot their investigation to search for suspicious DNS queries, periodic network beaconing, or unrecognized parent-child process chains within their own environment. Integrating relevant and actionable threat intelligence allows defenders to stay ahead of emerging attack trends and identify threats before they escalate into full-blown incidents.

Collecting and analyzing logs from diverse sources

Data diversity is equally critical to effective threat hunting. Modern attackers often blend tactics across multiple layers of infrastructure, making it essential to analyze logs from network traffic, endpoints, identity platforms, and cloud services simultaneously. For example, a hunter investigating potential data exfiltration might correlate endpoint telemetry showing mass file compression with network logs revealing large outbound transfers to uncommon IP ranges. This multi-source approach reduces blind spots and improves the accuracy of threat detection.

Collaboration between SOC analysts and threat hunters

Successful hunts are not just about tools and data, they require collaboration between SOC analysts and threat hunters. SOC teams often detect early warning signs, such as multiple failed login attempts or suspicious account behavior, which can serve as critical starting points for deeper threat hunts. For example, if a SOC analyst observes repeated failed logins followed by a successful attempt, hunters could investigate further for signs of or credential stuffing, enriching the investigation with broader threat context.

Continuous learning from hunting exercises and updating playbooks

Continuous learning loops and documentation are essential for long-term success. Each hunt should conclude with a debrief where insights are shared across teams and detection rules are updated based on findings. For example, if a hunt uncovers a novel phishing payload using HTML smuggling, detection engineers can create and test YARA rules targeting the specific obfuscation techniques observed. By regularly updating playbooks, detection rules, and sharing findings across teams, organizations build a resilient, knowledge-driven defense strategy.

Ultimately, a mature threat hunting practice combines structured methodology, collaboration, and continuous iteration, empowering defenders to shift from reactive defense to proactive cyber resilience.

Understanding Detection Engineering

In this section we aim to walk you through the process of detection engineering and its value. Detection engineering is the practice of designing, implementing, and refining security detections to effectively identify and respond to threats within an organization's environment. At its core, detection engineering focuses on producing high-quality detections that reliably identify malicious activities, reduce false positives, and improve response times for defenders.

A fundamental distinction in detection engineering is between signature-based detection and behavioral detection. In signature-based detection the threats are characterized by known-malicious indicators (like IP addresses, domain names and file hashes). Behavioral-based detections require deeper understanding of the methods used by threats, and behavioral patterns are used to identify potential threats.

Detection engineering is the process of understanding adversarial TTPs, identifying and extracting patterns and developing detection rules and models for detection. Detection engineering works with the input provided by threat intel teams, and requires deep understanding of the detection tradecraft, including availability of telemetry and potential impact of new detections. Custom detection rules are essential for tailoring detection strategies to an organization's unique requirements, addressing specific risks and operational nuances that standard rules may miss.

Good detections are specific and provide enough context so that defenders can expedite alert triage and response. MITRE ATT&CK is the de facto standard framework for classifying detections. A confidence level that classifies the detection logic by its (expected) accuracy is of major importance. This allows the detection engineering team to classify logic accordingly: logic that is focused on accuracy (low FPs) will be categorized as higher confidence; and logic focused on recall (low FNs) would be categorized as lower confidence. Members of Palantir's security team published an Alerting and Detection Strategies Framework that provides a very prescriptive approach to getting started with detection engineering: .

A resilient detection engineering process is rooted in continuous improvement, pace of execution, and solid release process. The ability to create, modify, and release detection logic with ease is fundamental. New detections are usually needed to adapt to evolving TTPs used by threat actors. It is also key to refine existing rules based on the understanding of what is common in the environment to reduce noise. Adaptability is the name of the game. Best practices from the software engineering world like version control, peer review, continuous integration, automated testing, observability, and solid continuous delivery pipelines provide a good basis for a sound detection engineering process that supports iterative improvement over time. Adaptability can be achieved by adopting detection-as-code approaches and modern DevOps models. Key stages of an iterative detection engineering flow are shown below.

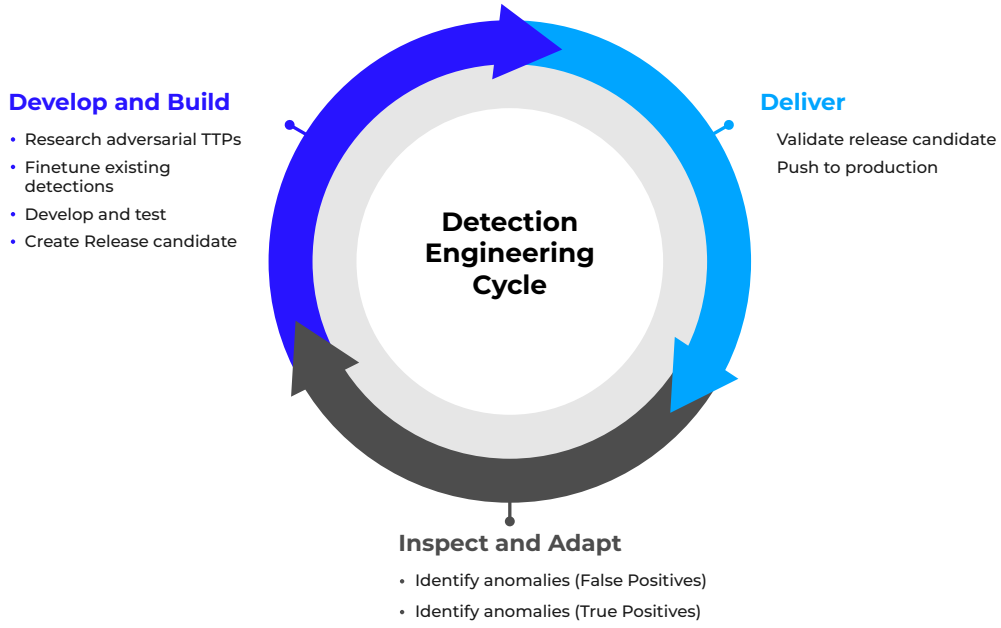

Figure 2: Detection Engineering Process

Best Practices for Detection Engineering

Detection-as-code

Best practices from the software engineering realm like usage of version control systems, peer review processes, automated testing and continuous integration contribute to detections with built-in quality. Consistency, relevancy, uniformity, and readability are also good characteristics of great detections.

Threat-informed detections

Detections based on traditional IOCs like file hashes, IP addresses, and URLs require automatic processing of threat intel feeds. These are frequently strong indicators but tend to lose value very quickly (decay time) as it is easier for adversaries to change these indicators.

Detections based on behavioral patterns need to be refined based on the understanding of adversarial TTPs. Broad detections can be added for compliance or coverage.

Threat-Informed detections based on the understanding of adversarial goals, methods and tools are more resilient in time and help to detect unknown threats that leverage known TTPs.

Telemetry analysis and customer-centric improvements

A threat-driven mindset serves as guidance and motivation for driving continuous improvement. In enterprise threat hunting teams, the work doesn't end when a new detection is deployed. Even highly effective detections require ongoing refinement and maintenance to ensure they provide maximum value in detecting real threats. Continuous telemetry analysis is essential to assess detection efficacy, reduce noise, and adapt to evolving attacker behaviors.

Pace of execution and peace of mind

Developing on cadence with well established rhythms and shorter iterations releases pressure from the team. No discussions nor surprises about deadlines or code freezes. It increases the predictability of the team's deliverables both internally and externally. It is another best practice borrowed from the software engineering world.

Release on demand

Following the short cycles of research and development, it is important to get the support from the organization to get the detection assets released periodically. This requires coordination with the teams in charge of the production deployment and content operations. Predefined release dates and times add predictability with multiple stakeholders.

It is important to have a fast track process that allows a push to production immediately, when emergency releases are needed.

Expect the unexpected

When working against cyber-threats, the only constant is change! New threats, exploits, vulnerabilities, and high profile attacks will happen every day. The detection engineering team needs a criteria and process to handle emerging threats. And this should be part of the planned work for the week.

Adversary emulation

Performed both by internal and external teams, adversary emulation is a great method to assess the detection efficacy and identify gaps.

Challenges in Detection Engineering

Timely response

Effective detection engineering against new and emerging threats requires practical and timely solutions. Delivering great solutions requires understanding of key TTPs and identifying behavioral patterns that are less prone to change than traditional IOCs like file hashes or IPs.

Delivering Context

Effective detection engineering should produce detections that deliver as much context as possible (to be used by SOC analysts when alert triggers). Again, delivering great solutions requires understanding of key TTPs, to put SOC analysts on track for faster triage and investigation.

Roadmap vs. reality

Keeping a roadmap and at the same time responding to emerging trends can be challenging. In cybersecurity, urgent things usually win over important things. While engineering teams strive to be proactive, often they are forced to be reactive due to urgent matters .

Communicating value and progress

One of the main challenges for detection engineering teams is to communicate and make visible the value of their work. The MITRE ATT&CK Navigator is a tool that can be used for this purpose. MITRE ATT&CK Navigator displays information using the matrix format from ATT&CK, with tactics and techniques.This is a convenient way to deliver an overall summary, however it is harder to visualize the deltas from release to release. TTPs are the heart of the MITRE ATT&CK framework and ultimately require an in-depth conversation about adversarial TTPs to truly convey value.

Access to actionable threat intelligence

As it has been pointed out many times in this document, great detection engineering requires understanding of adversarial TTPs. Knowledge and understanding of adversarial TTPs is not easily derived from a list of file hashes, IPs, or URLs. Actionable threat intelligence delivers insights about adversaries' objectives and modus operandi with practical observations (e.g. from incident response reports). Two sources to start with include the Trellix Advanced Research Center and an open source threat intel such as this Github repository. Curiosity plus accessible examples can spark the next wave of security researchers!

Case Study 2

Efficient Detection Engineering Process Reduces False Positive Rates

A large healthcare organization faced challenges with high false positive rates in its Security Information and Event Management (SIEM) system, leading to alert fatigue among analysts. To enhance detection accuracy, the security team embarked on refining their detection engineering processes.

They began by reviewing existing detection rules to identify overly broad criteria that contributed to false positives. By developing profiles for typical user and system behaviors, they could distinguish between legitimate activities and potential threats. They also implemented machine learning models to analyze patterns and adapt to evolving behaviors, reducing reliance on static rules.

Data analysis

The team conducted a thorough analysis of historical alert data to identify common characteristics of false positives. They collaborated closely with IT and business units to understand legitimate workflows and adjust detection parameters accordingly. A continuous feedback loop was established, allowing analysts to provide input on alert accuracy and enabling ongoing refinement of detection logic.

As a result of these efforts, the organization reduced false positive alerts from 60% to 15% over six months. This significant improvement allowed analysts to focus on genuine threats, ultimately enhancing the organization's security posture.

Tools and Technologies for Threat Hunting and Detection Engineering

Access to enterprise logs (security data lake)

Data access is fundamental for threat hunting and detection engineering. Endpoint logs (such as Process Creation, Network Access, and File System), Proxy logs and application logs are some of the most valuable data sources for both threat hunting and detection engineering. A simple query to retrieve process creation events related to the execution of PowerShell can be the start of the threat hunting and/or detection engineering experience. You can get started with open source tools such as an ELK stack (Elasticsearch, Logstash, and Kibana), or commercial tools including Splunk, AWS, and Google Chronicle.

SIEM and XDR

Technologies such as SIEM and XDR work as enterprise log aggregators. Usually SIEM, XDR, as well as EDR and NDR products come with built-in detections, interfaces and/or languages for querying the data, hunting for threats, and data visualization capabilities (dashboards).

Dashboards, queries, and notebooks

A practical approach to threat hunting and detection engineering involves creation and monitoring of dashboards to keep track of detection trends. Data visualization focused on the most common and less common detection are sources for launching both threat hunting sessions as well as detection engineering fine-tuning efforts. During analysis, direct access to the datasets using query languages (like SQL, KQL, etc.) provide more flexibility than the dashboards to collect all the details needed. The use of notebooks (e.g. Jupyter notebooks) can help to create structure for the analysis supporting programmatic data analysis and processing.

The MITRE ATT&CK matrix

The MITRE ATT&CK matrix is a great resource for giving shape and structure both threat hunting and detection engineering efforts. It not only offers a catalog of adversarial TTPs but also a catalog of data sources, threat actor profiles, and software tools. It has become the de facto standard for exchanging knowledge about adversarial TTPs.

Languages and rulesets

Sigma is an open-source language for defining detection logic. There are several open repositories of Sigma rules, and it's common to see security researchers sharing their research. There are tools to convert generic Sigma rules to specific SIEM platforms.

Case Study 3

Appropriate Use of MITRE ATT&CK to Address Coverage

A multinational corporation sought to assess and improve its security coverage against known adversary behaviors. The security team decided to leverage the MITRE ATT&CK framework to identify gaps and enhance their detection capabilities.

The first step was mapping existing security controls and detection mechanisms to the ATT&CK framework, providing a comprehensive view of coverage across various tactics and techniques. A gap analysis revealed areas with limited or no coverage, and the team prioritized addressing techniques commonly used by threat actors targeting their industry.

To close these gaps, the team developed and implemented new detection rules and alerts, ensuring alignment with ATT&CK techniques. They validated their improvements by conducting adversary emulation exercises using tools like Atomic Red Team, testing the effectiveness of the new detections.

The organization achieved comprehensive visibility into adversary tactics and techniques, leading to the development of targeted detections. This proactive approach resulted in earlier detection of attempted breaches and a more robust defense posture.

Threat hunting methodology

A structured threat hunting methodology enables hunters to systematically approach their objectives, assess findings, and refine hypotheses over time. A typical methodology includes the following steps:

1. **Preparation:** Define objectives, scope, and potential threat models. Establish clear guidelines on data collection, analysis criteria, and risk tolerance.
2. **Hypothesis development:** Based on gathered intelligence, internal data, and organizational insights, develop hypotheses that guide the search for anomalies or suspicious behavior.
3. **Data collection and analysis:** Gather relevant data from diverse sources, including logs, endpoint data, network telemetry, and external intelligence feeds. Employ machine learning, statistical analysis, or forensic techniques to pinpoint deviations.
4. **Investigation and validation:** Delve into detected anomalies to ascertain if they represent genuine threats. Validate findings and assess the level of risk associated with each anomaly.
5. **Response and documentation:** Develop response strategies for detected threats, including containment, eradication, and recovery actions. Document and share findings, methodologies, and responses to inform future hunts.

There are several open repositories of platform/product specific detection rulesets.For example, YARA-L 2.0 (by Google) is a language used to create rules for searching enterprise log data. It's possible to ingest enterprise logs in databases, and then use the database query language (e.g., plain SQL) to define the hunting/detection queries.

Detection as code

This aspect has been covered in the best practices for detection engineering, but also applies to threat hunting. Leveraging Github repositories, and similar control version systems, to document and maintain hunting queries has multiple benefits including knowledge sharing; but it is also important to document the scope and results of the hunting exercises.

The Future of Threat Hunting and Detection Engineering

The role of AI and ML in enhancing threat detection

Artificial Intelligence (AI) and Machine Learning (ML) are revolutionizing threat hunting and detection engineering by enabling faster, more accurate anomaly detection in massive data sets. Traditional rule-based detection often struggles to keep up with evolving attacker techniques, but ML-driven models can identify subtle deviations from baseline behavior, such as an employee account suddenly accessing large amounts of sensitive data outside of normal working hours or from a different geographical region. AI-powered behavioral analytics also enhance threat hunting by clustering similar attack behaviors across different environments, allowing security teams to proactively detect emerging attack patterns before they are widely known. For example, ML can be used to detect credential stuffing attacks by analyzing login attempts across multiple accounts and flagging IPs with high failure rates followed by a successful login. Additionally, adversarial AI techniques—where red teams train ML models against simulated attack scenarios—help refine detection mechanisms against advanced adversaries who may attempt to evade AI-powered defenses. As cyber threats grow more sophisticated, the integration of AI and ML will not replace human threat hunters but will serve as an indispensable force multiplier, enabling them to focus on high-impact investigations instead of drowning in low-priority alerts.

The role of cloud-native detection mechanisms

With the rapid adoption of cloud computing, traditional detection methods must evolve to address the unique challenges of cloud environments. Unlike on-premises infrastructures, where visibility is typically centralized, cloud environments are highly dynamic, with ephemeral workloads, serverless functions, and decentralized access controls. Cloud-native detection mechanisms leverage API-driven monitoring, real-time telemetry, and identity-based analytics to detect threats that traditional endpoint and network-based detections may miss. For example, in AWS or Azure, threat hunters can analyze CloudTrail and identity provider logs to detect anomalous API calls, such as an attacker attempting to escalate privileges using an unused IAM role. Similarly, cloud workload protection platforms (CWPPs)

use runtime behavioral analysis to detect container-based attacks, such as cryptojacking operations running in Kubernetes clusters. Another emerging trend is the use of security data lakes, where vast amounts of cloud logs are ingested into platforms like AWS Security Lake or Google Chronicle, allowing for scalable and efficient detection engineering. By shifting towards cloud-native detection, security teams can better monitor and respond to threats in increasingly complex hybrid and multi-cloud environments.

Threat intelligence and its impact on proactive hunting

Threat intelligence has transitioned from a reactive tool to a cornerstone of proactive threat hunting, providing real-time insights into adversary TTPs. High-fidelity threat intelligence enables security teams to pivot from simply blocking known indicators of compromise (IOCs) to actively searching for adversaries before they strike. For instance, if threat intelligence identifies a surge in activity from an APT group exploiting a zero-day vulnerability in a widely used SaaS platform, threat hunters can proactively query logs for any similar activity within their own environment. Additionally, threat intelligence feeds enriched with machine learning help identify emerging attacker infrastructure, allowing defenders to hunt for domains with characteristics similar to known command-and-control (C2) servers before they are operationalized in attacks. Another innovative approach is threat intelligence operationalization, where structured threat intelligence is automatically integrated into detection engineering pipelines, ensuring that hunting teams are continuously adapting to the latest threats. By leveraging real-time threat intelligence, security teams can reduce dwell time and mitigate threats before they escalate into full-blown incidents.

The importance of threat hunting in a zero-trust architecture

Zero-trust security models assume that no entity—whether inside or outside the network—should be automatically trusted, making continuous anomaly detection a fundamental requirement. In a zero-trust environment, threat hunting becomes a key strategy for detecting identity-based attacks, where adversaries compromise legitimate accounts to bypass traditional security controls. For example, if a zero-trust-enabled system detects an employee logging in from an unrecognized device in a foreign country while simultaneously accessing sensitive files through a VPN, threat hunters can investigate further for signs of session hijacking or token theft. Similarly, by analyzing micro-segmentation logs, hunters can identify lateral movement attempts, such as an attacker attempting to access a restricted database by exploiting an over-permissioned service account. Advanced analytics, including user and entity behavior analytics (UEBA), further enhance zero-trust threat hunting by dynamically assessing risk based on deviations from normal behavioral patterns. As organizations continue to adopt zero-trust frameworks, integrating threat hunting with adaptive access controls will be critical in ensuring that even sophisticated stealth attacks are uncovered before they can cause significant damage.

Conclusion

Relying solely on reactive security measures leaves organizations vulnerable to threats that bypass defenses. Proactive threat detection through structured threat hunting and detection engineering reduces attacker dwell time and uncovers hidden threats before they escalate. Instead of waiting for alerts, hunters should investigate behavioral anomalies, such as abnormal account activity or unexpected data transfers, to identify stealthy adversaries early.

Attackers constantly adapt, making continuous improvement in threat hunting and detection engineering essential. Static detection rules quickly become outdated, requiring security teams to manually refine detection logic based on emerging threats like fileless malware and living-off-the-land attacks. Regular purple teaming, real-world threat intelligence integration, and automated detection updates ensure defenses stay relevant and effective.

A strong security posture depends on skilled analysts, well-defined processes, and advanced technology working together. AI and automation enhance detection but cannot replace human expertise in understanding complex attack patterns. Clear methodologies, structured feedback loops, and real-time threat intelligence empower teams to act decisively. Organizations that invest holistically in people, processes, and technology will be best positioned to detect, respond to, and prevent emerging cyber threats.

For more information on the emerging threats or to read the latest cyber security trends report, check out the Advanced Research Center.

Contributors

Alejandro Houspanossian is a Sr. Staff Security Researcher at Trellix with more than 18 years experience in software engineering and threat detection. Alejandro has helped create numerous renowned security products including McAfee Investigator, MVISION EDR, and Trellix EDR. Alejandro has presented his work at renowned conferences such as the SANS Blue Team Summit, SANS Threat Hunting Summit, Ekoparty, CyberGen, and BHack Brasil. He holds a degree in Computer Science from Universidad Nacional del Centro de la Provincia de Buenos Aires in Argentina.

Tomer Shloman is a Senior Security Researcher with Trellix Advanced Research Center where he specializes in threat hunting, malware analysis, and reverse engineering. With over 15 years of experience, he focuses on identifying and tracking sophisticated threats. He leverages his expertise to enhance detection and develop proactive defense strategies. Tomer is passionate about disrupting cyber threats and sharing insights through research and collaboration. When he's not hunting adversaries, he enjoys mentoring and speaking at industry conferences.

Ilya Kolmanovich is the Senior Director of Security Research at Trellix, where he leads global teams within the XDR business function. Overseeing both Product Research for EDR & XDR solutions and an advanced Data Science group, Ilya ensures Trellix stays at the forefront of cyber defense. His team focuses on identifying and responding to emerging threats, developing cutting-edge detection capabilities that protect customers worldwide.

Joe Malenfant is the Director of Product Marketing for Trellix's endpoint security portfolio. WIth over 15 years experience spanning endpoint, network, cloud, identity, and IoT, Joe is always eager to dig into the complexity of a technology. in an effort to distill it and be consumable by the masses. Prior to joining Trellix he held leadership positions at Ping Identity, Vectra AI, and Cisco where he launched their first EDR solution. Joe holds an MBA and undergraduate degrees in business.

Copyright © 2025 Musarubra US LLC