



White Paper

# Managed Intelligence™

Shaping a Threat Hunt Program to Operationalize Data, Resource Accordingly, and Protect the Business

## Actionable Intelligence

**Deriving actionable intelligence to enhance organizational security is a challenge faced by all global companies** and often further complicated by intertwined networks resulting from mergers and acquisitions. With the volumes of data, it's important to shape a threat hunting program to be able to consume and operationalize data collected from various sources.

## Background

In today's threat landscape, standard tools and predefined compliance and risk policies are typically far too conventional to provide the appropriate risk deterrence against the threats an organization faces.

While many organizations have a plethora of tools through their network and application infrastructure that push data into a SIEM, they may not be appropriately scoped to collect the data necessary to detect likely threats.

Beyond detecting threats, organizations also need to identify risks and have the ability to address them. Understandably, not all organizations have the necessary resources to address risks; they might not even know the risks exist.

There are often cases where a tool deployment or policy change is the ultimate solution to an organization's security problems. However, threat actors have evolved just as defenders have. Tools can be subverted, and policies may be misconfigured, allowing attackers access to a network while the organization has a false sense of security.

Enter threat hunting, the proactive function of an organization's security program. The Threat Hunt team is mandated with discovering threat actors that have already bypassed currently deployed tools and policies.

This article describes how Nisos develops such a program and implements the proper metrics that will eventually allow an organization to operationalize many streams of intelligence to properly mitigate risk for the business.

## Starting Out - Identify the Threat

Threat actors vary based on TTPs, targets, and motives, summarized in the below chart:

Threat Actor	Methods	Motives
Nation State	Highly advanced and difficult to detect. May utilize human enabled operations in order to obtain access into networks.	Espionage, sabotage, cyber war
Criminal Group	Often use "spray and pray" tactics. Will seek out targets of opportunity. May use commoditized exploit kits to develop tools.	Financial
Hacktivist	Denial of Service attacks, webpage defacement, hacking into social media accounts.	Political, Ideological
Insider Threat	Utilizes accesses granted. No tools needed. Can blend in with the noise.	Financial, Revenge
Script Kiddie	Simple to use tools freely available online.	Curiosity, fun, clout, financial.

Different organizations are targets of different threat actors. For example, a large R&D firm involved in developing weapons systems may be targeted by a nation state, while a large restaurant chain may not. Organizations may identify their likeliest threat based on previous breaches that involved their industry.

Organizations may also want to identify their "Crown Jewels." Information considered to be of such value that any sort of breach or compromise will be catastrophic to the business, either through loss of money, loss of client/public trust or legal action from regulatory agencies. In the world of GDPR, breaches are not only embarrassing, they are expensive.

## Coverage Map - Collecting What Matters

Once an organization identifies the threat actor(s), it should create a coverage map with the purpose of identifying collection gaps. This does not have to be difficult; the MITRE ATT&CK Framework is a thorough matrix that covers tactics and techniques utilized by threat actors. In addition, MITRE has a list of threat actors along with the techniques and tactics they use. The team can easily map the MITRE ATT&CK framework to the organization's security controls as demonstrated in the example below.

### Example Coverage Map - Criminal Organization

An organization identifies financially motivated criminal groups as their likeliest threat actor, singling out the groups FIN5, FIN6, FIN7 and FIN8 for specific attention. Using the information MITRE has on those groups and mapping detection capabilities to the MITRE ATT&CK Framework, they created the following coverage map:

Initial Access	Execution	Persistence	Privilege Escalation	Defense Evasion	Credential Access	Discovery	Lateral Movement	Collection	Command and Control	Exfiltration	Impact
11 Items	34 Items	62 Items	32 Items	69 Items	21 Items	23 Items	18 Items	13 Items	22 Items	9 Items	16 Items
Drive-by Compromise	AppleScript	Access Token Manipulation	Account Manipulation	Account Manipulation	Account Manipulation	Account Discovery	AppleScript	Commonly Used Port	Automated Exfiltration	Account Access Removal	
Exploit Public-Facing Application	CMSTP	Accessibility Features	Accessibility Features	Binary Padding	Brute Force	Application Window Discovery	Application Deployment Software	Automated Collection	Communication Through Removable Media	Data Destruction	
External Remote Services	Compiled HTML File	AppCert DLLs	AppCert DLLs	AppCert DLLs	Brute Force	Browser Bookmark Discovery	Component Object Model and Distributed COM	Clipboard Data	Connection Proxy	Data Encrypted for Impact	
Hardware Additions	Component Object Model and Distributed COM	AppCert DLLs	AppCert DLLs	Clear Command History	Credentials from Web Browsers	File and Directory Discovery	Exploitation of Remote Services	Data from Information Repositories	Custom Command and Control Protocol	Defacement	
Replication Through Removable Media	Control Panel Items	Application Shimming	Application Shimming	Code Signing	Credentials in Registry	Network Service Scanning	Internal Spearphishing	Data from Local System	Custom Cryptographic Protocol	Disk Structure Wipe	
Spearphishing Attachment	Dynamic Data Exchange	Authentication Package	Authentication Package	Compiled HTML File	Component Object Model Hijacking	Network Sniffing	Logon Scripts	Data from Removable Media	Data Encoding	Endpoint Denial of Service	
Spearphishing Link	Execution through API	Bootkit	Dylib Hijacking	Component Firmware	Component Object Model Hijacking	Password Policy Discovery	Pass the Hash	Data from Removable Media	Data Obfuscation	Firmware Corruption	
Spearphishing via Service	Execution through Module Load	Browser Extensions	Elevated Execution with Prompt	Connection Proxy	Control Panel Items	Peripheral Device Discovery	Pass the Ticket	Data Staged	Domain Fronting	Inhibit System Recovery	
Supply Chain Compromise	Exploitation for Client Execution	Change Default File Association	Emond	Control Panel Items	Hooking	Permission Groups Discovery	Remote File Copy	Email Collection	Domain Generation Algorithms	Network Denial of Service	
Trusted Relationship	Graphical User Interface	Component Firmware	Exploitation for Privilege Escalation	Declassify/Decode Files or Information	Input Prompt	Process Discovery	Remote Services	Input Capture	Fallback Channels	Resource Hijacking	
Valid Accounts	InstallUtil	Extra Window Memory Injection	Disabling Security Tools	Disabling Security Tools	Kerberoasting	Query Registry	Replication Through Removable Media	Man in the Browser	Multi-hop Proxy	Scheduled Transfer	
	Launchctl	Create Account	DLL Search Order Hijacking	DLL Search Order Hijacking	LLMNR/NBNS Poisoning and Relay	Security Software Discovery	Screen Capture	Screen Capture	Multi-Stage Channels	Stored Data Manipulation	
	Local Job Scheduling	DLL Search Order Hijacking	DLL Side-Loading	DLL Side-Loading	Execution Guardrails	Software Discovery	Shared Webroot	Video Capture	Multiband Communication	System Shutdown/Reboot	
	LSASS Driver	Dylib Hijacking	Hooking	Hooking	Exploitation for Defense Evasion	System Information Discovery	SSH Hijacking		Port Knocking	Transmitted Data Manipulation	
	Malware	Emond	Image File Execution Options Injection	Extra Window Memory Injection	Private Keys	System Network Configuration Discovery	Taint Shared Content		Remote Access Tools		
	PowerShell	External Remote Services	File System Permissions Weakness	File System Permissions Weakness	Securityd Memory	System Network Connections Discovery	Third-party Software		Remote File Copy		
	Regsvcs/Regasm	File System Permissions Weakness	Launch Daemon	Launch Daemon	Stall Web Session Cookie	System Owner/User Discovery	Windows Admin Shares		Standard Application Layer Protocol		
	Regsvr32	Hidden Files and Directories	Parent PID Spoofing	File System Logical Offsets	Two Factor Authentication Interception	System Service Discovery	Windows Remote Management		Standard Cryptographic Protocol		
	Rundll32	Hooking	Path Interception	Gatekeeper Bypass	Group Policy Modification	System Time Discovery	Virtualization/Sandbox Evasion		Standard Non-Application Layer Protocol		
	Scheduled Task	Image File Execution Options Injection	Port Monitors	Hidden Files and Directories	Hidden Users				Uncommonly Used		
	Scripting	Kernel Modules and Extensions	PowerShell Profile	Hidden Window	Hidden Window						
	Service Execution	Launch Agent	Launch Daemon	Launch Daemon	Launch Daemon						
	Signed Binary Proxy Execution	Launchctl	Launchctl	Launchctl	Launchctl						
	Signed Script Proxy Execution	Launchctl	Launchctl	Launchctl	Launchctl						
	Source	Launchctl	Launchctl	Launchctl	Launchctl						
	Space after Filename	Launchctl	Launchctl	Launchctl	Launchctl						
	Third-party Software	Launchctl	Launchctl	Launchctl	Launchctl						
	Trap	Launchctl	Launchctl	Launchctl	Launchctl						
	Trusted Developer Utilities	Launchctl	Launchctl	Launchctl	Launchctl						
	User Execution	Launchctl	Launchctl	Launchctl	Launchctl						
	Windows Management Instrumentation	Launchctl	Launchctl	Launchctl	Launchctl						

(Sample Coverage Map. Above created using: <https://mitre-attack.github.io/attack-navigator/enterprise/>)

Note that specific TTPs are selected, allowing the organization to identify and address the gaps that matter. This may not always be possible due to a variety of reasons. The amount of logging necessary may be too much and impractical, or the TTP may require a tool that would require a significant financial investment that is not in budget.

## Cyber Analytics Repository

A Cyber Analytics Repository (CAR) is a catalog of product-specific queries used to hunt TTPs identified in the coverage map. For example, if an organization uses Splunk the query will be in SPL, if an endpoint detection and response agent (Endgame for example) is used the query will be in EQL, etc. The CAR may also include Use Cases from which to create alerting rules. Use Cases can be described as the type of activity we expect from threat actors. An example Use Case may be PowerShell execution from a malicious word document. Alerts are then based on this Use Case.

One popular CAR template is available from MITRE's GitHub page: <https://github.com/mitre-attack/car>. It includes queries for several TTPs in the MITRE ATT&CK framework.

Below is an example of the Powershell page from MITRE:

### CAR-2014-04-003: Powershell Execution

PowerShell is a scripting environment included with Windows that is used by both attackers and administrators. Execution of PowerShell scripts in most Windows versions is opaque and not typically secured by antivirus which makes using PowerShell an easy way to circumvent security measures. This analytic detects execution of PowerShell scripts.

PowerShell can be used to hide monitored command line execution such as:

- net use
- sc start

**Submission Date:** 2014/04/11

**Information Domain:** Host

**Data Subtypes:** Process

**Analytic Type:** TTP

**Applicable Platforms:** Windows

**Contributors:** MITRE

#### ATT&CK Detection

Technique	Tactic	Level of Coverage
PowerShell	Execution	High
Scripting	Defense Evasion	Moderate

#### Data Model References

Object	Action	Field
process	create	exe
process	create	parent_exe

#### Implementations

##### Pseudocode

Look for versions of PowerShell that were not launched interactively.

```
process = search Process:Create
powershell = filter process where (exe == "powershell.exe" AND parent_exe != "explorer.exe")
output powershell
```

##### Splunk, Sysmon native

Splunk version of the above pseudocode.

```
index=__your_sysmon_index__ EventCode=1 Image="C:\\Windows\\*\\powershell.exe" ParentImage!="C:\\Windows\\explorer.exe"|stats values(Corr
```

(Sample of MITRE's CAR describing PowerShell technique and ways to Hunt)

## Hunting Begins

Ultimately, the goal of any security team is prevention of malicious activity through the deployment of security controls. This is not always possible, due either to lacking controls or the fact that not all malicious activity is distinguishable from normal user activity. This is where threat hunting comes in.

Threat hunters detect malicious activity and coverage gaps overlooked by security controls or auditing tools, but detection should not be their only role. They should drive threat and risk mitigation. It is not enough to discover a threat actor on a network or a coverage gap. They must reach out to the appropriate teams in an organization with suggestions on how to remove threats and close gaps.

## Relationship Overview

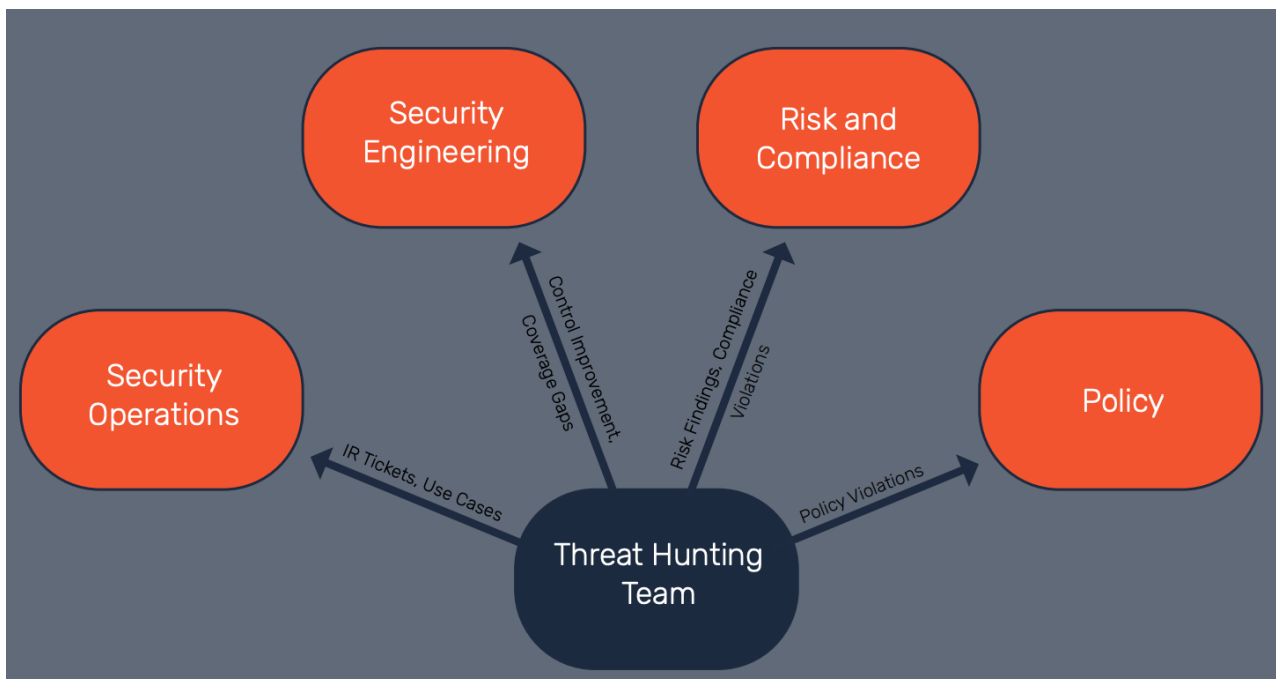
Security teams at organizations are often split into the following:

### Team

Security Operations  
Security Engineering  
Risk and Compliance  
Policy

### Responsible for:

Developing alerts and Incident Response  
Security tools deployed to systems (EDR, AV, firewalls)  
Tracking systemic risks and information assurance  
Creating and updating policies for organization, including acceptable use



## Reporting Success - Measuring What Matters

Threat hunting carries an unusual value proposition. Unlike a development team or a product team, a Threat Hunt team's outcome is less tangible. The Threat Hunt team delivers findings, not products, divided into two sets; threats and risks. Threats include malicious or suspicious activity on the network. Risks include coverage gaps, missing logs or missing alerts. Both sets are tailored to an organization's requirements.

Not all findings will have equal value. For example, discovering an advanced threat actor on a network will have much more value than finding adware on an endpoint. It is important to define values for findings and to keep track of these values in order to demonstrate the return on investment that the Threat Hunt team brings to an organization.

Below is an example of one way to define and quantify Threats and Risks.

Values of Threats Discovered	
Value	Definition
High	<ol style="list-style-type: none"> <li>1. Threat actors on the network</li> <li>2. Malware that may allow for the remote control of a system and/or further infections</li> <li>3. Unauthorized external access regardless of intent or motive</li> <li>4. Credential leaks which may lead to future compromise, whether leak is public or internal</li> </ol>
Medium	<ol style="list-style-type: none"> <li>1. Policy abuse</li> <li>2. Risky software (e.g., RAT, unapproved AV)</li> <li>3. Risky traffic (e.g., large transfers to drop sites)</li> </ol>
Low	<ol style="list-style-type: none"> <li>1. Potentially unwanted software</li> </ol>

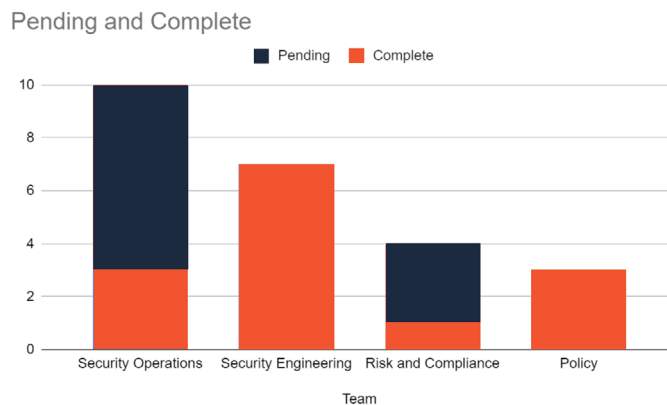
Values of Risks Discovered	
Value	Definition
High	<ol style="list-style-type: none"> <li>1. Requires a large financial investment to address and possible cultural change to address.</li> <li>2. Use Case Development</li> <li>3. Misconfiguration that may pose a risk to the environment</li> </ol>
Medium	<ol style="list-style-type: none"> <li>1. Missing patches that pose a risk to the environment</li> <li>2. Missing logs</li> </ol>
Low	<ol style="list-style-type: none"> <li>1. Gaps that are quick fixes (i.e. missing security product or patch on a single host)</li> </ol>

The Threat Hunt team uses the above metrics to demonstrate their value and organizational impact to management.

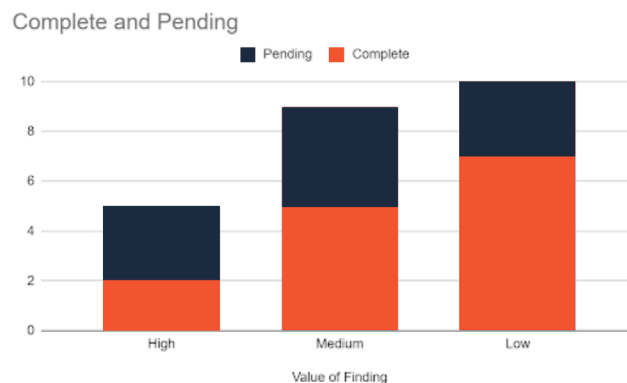


To ensure that findings are addressed the threat team must track closure. A ticketing system such as Jira is a useful tool for tracking closure. Assign each ticket with the responsible security team as well as its value metric. At the end of each quarter, the team can provide these metrics to management. In turn, management can use the metrics as a guide to allocate resources. For example, the Threat Hunt team may have discovered risks and gaps that fell under the onus of the security engineering team. By the end of the quarter, if the security engineering team has only actioned a small percentage, management can consider allocating additional resources to that team.

The chart below is a sample deliverable to management.



The above may suggest that the Security Engineering and Policy teams are capable of addressing the issues discovered by the Threat Hunt team, while the Security Operations or Risk and Compliance Team may require more investment and resources. Of course the Threat Hunt team would want to quantify its findings and present them as a chart to management and demonstrate ROI:



## About this Campaign

One application of an actionable cyber threat intelligence program should inform where a security stack cannot detect. Over the coming month, Nisos will publish a variety of articles that go deeper building an actionable cyber intelligence program that builds on a hypotheses-led threat hunting program with limited resources allowing a program to scale over time. We will dig into more depth around how to effectively use risk findings, penetration testing results, threat intelligence feeds, and incident response reports to systematically report and track hypotheses that deliver actionable reporting and metrics. If captured appropriately, a security program can scale by reducing resources to the security unit teams but reducing time to respond which is the ultimate goal of any SOC.