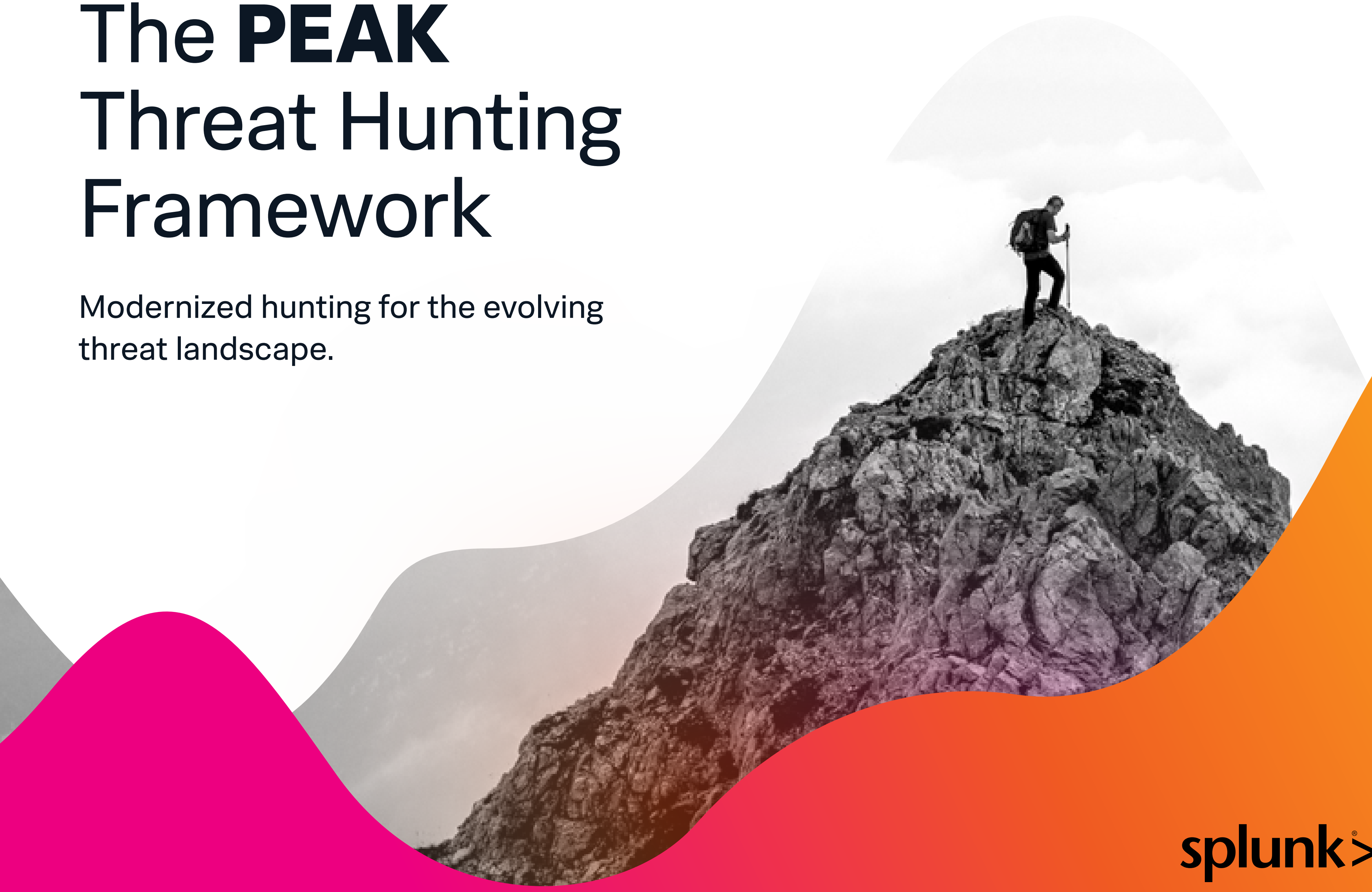
The **PEAK** Threat Hunting Framework

Modernized hunting for the evolving threat landscape.



splunk>

Table of contents

Today’s digital threat landscape is quite intricate, with the number and complexity of threats increasing daily. Organizations use threat hunting to identify malicious activity on their networks, increasing security and helping to manage risk. As hunting programs continue to mature, teams need a framework that not only provides a blueprint to the threat hunting process, but also incorporates the additional experience and lessons learned by top hunters over the last several years.

This is where the PEAK Threat Hunting Framework comes in handy. It’s a practical, customizable approach designed to help organizations create or refine their hunting programs to gain the most value from threat hunting across their entire security organization. It also helps individual hunters by providing expert guidance on how to plan for, conduct and document their hunts.

PEAK does all this by first defining three different types of hunts: hypothesis-driven, baseline, and model-assisted. Each one has unique advantages and can provide a new perspective on your data. Furthermore, PEAK helps transform the information you discover into automated detections so your hunters can focus on their next hunt and not waste their efforts repeatedly hunting the same things.

PEAK also includes a variety of metrics to help you evaluate the effectiveness of your hunting efforts. It has a maturity model to help you determine where your overall hunt program stands and actions you can take to improve.

If you’re ready to take your hunting program to the next level, consider the PEAK framework. It’s a fantastic method to learn about threat hunting and make your organization more secure.

What is threat hunting?

The first question people ask about threat hunting is, “What exactly is it?” For our purposes, we’ll use the most popular definition: Threat hunting is any manual or machine-assisted process for finding security incidents that your automated detection systems missed.

The key here is that even though we often use computers, automation and machine learning techniques to help us identify and filter events of interest, hunting is always human-driven. Our curiosity, imagination and ability to deduce patterns of malicious activity, even when we have never encountered them before, are simply beyond the capabilities of today’s technology.

Why hunt?

Given the definition, one might assume that the purpose of threat hunting is to find more security incidents. This is how some organizations approach threat hunting, but it’s not the best way. The creation of new security incidents during the hunting process is actually a secondary benefit — it’s more of a by-product of the hunt than its primary purpose.

Because threat hunting requires human involvement, it comes with a relatively high cost. With the volume and velocity of security data coming into most organizations, comprehensive human review isn’t just costly, but entirely out of the question. Don’t think of hunting as a way for expensive humans to identify more security incidents.

When a hunter figures out a new way to detect malicious behavior or discovers a visibility gap, they can shore up defenses. This is often more valuable in the long run than simply discovering new incidents. Threat hunting shines when it drives improvements to an organization’s security posture.

Threat hunting frameworks

A hunting framework is a system of repeatable processes designed to make your hunting expeditions more reliable, effective and efficient. Threat hunting frameworks help you understand:

- Which types of hunts exist
- Which type might be most appropriate for your specific hunt
- How to perform each type of hunt
- What the outputs could or should be
- How to measure success

A good framework provides a clear set of guidelines that can be tailored to your specific program or even an individual hunt. It gives you repeatable processes and improves both the efficiency of your operations and the quality of your outputs.

Existing frameworks

Threat hunting frameworks have been around since at least 2015. Two of the most well known are the Sqrrl Threat Hunting Reference Model and TaHiTI. Their influence has shaped how we've hunted threats for years.

The Sqrrl threat hunting reference model (2015)

Published in three parts, Sqrrl's framework was not only the first, but remains one of the most influential threat hunting frameworks. It defines the hypothesis-driven hunting process as a loop with four stages:

1. Create a hypothesis
2. Investigate via tools and techniques
3. Uncover new patterns and TTPs
4. Inform and enrich automated analytics

TaHiTI: Targeted hunting integrating threat intelligence (2018)

The TaHiTI framework, created by a consortium of financial institutions known as the Dutch Payments Association, is another popular threat hunting framework. TaHiTi is best known for:

- Incorporating and building on pieces of the Sqrrl framework, such as hypotheses and the Hunting Maturity Model
- Adding a new type of hunt: the unstructured or data-driven hunt
- Integrating Cyber Threat Intelligence (CTI) into the hunting process

The PEAK threat hunting framework

PEAK, an acronym for "Prepare, Execute, and Act with Knowledge," brings a fresh perspective to threat hunting, incorporating the experiences and lessons learned from the security community over the past several years. It features:

- Three different types of hunting methodologies, detailed processes for each type, and guidance on selecting the right type for the job
- Explanations of the common types of hunting documentation and deliverables
- A new way to categorize and prioritize the different types of automated detections created as a result of hunting
- Key metrics any hunting program should track to show the effect of hunting on their organization's security posture

Hunters and their leaders can adopt the framework as-is or supplement it with their existing practices to create a program tailored to their unique requirements.

PEAK hunt types and structure

PEAK defines three distinct types of threat hunts:

- Hypothesis-Based Threat Hunts
- Baseline Threat Hunts
- Model-Assisted Threat Hunts (M-ATH)

Each type of hunt consists of the same three stages: *Prepare*, *Execute* and *Act*.

The *Prepare* phase is where you do everything necessary to maximize your chances of a successful hunt. You select topics, conduct research and generally plan out your hunt.

The *Execute* phase is where you implement that hunt plan. Although some would consider this where the “real hunting” happens, it’s important to understand that a hunt cannot be successful and impactful without all three phases.
It’s all hunting.

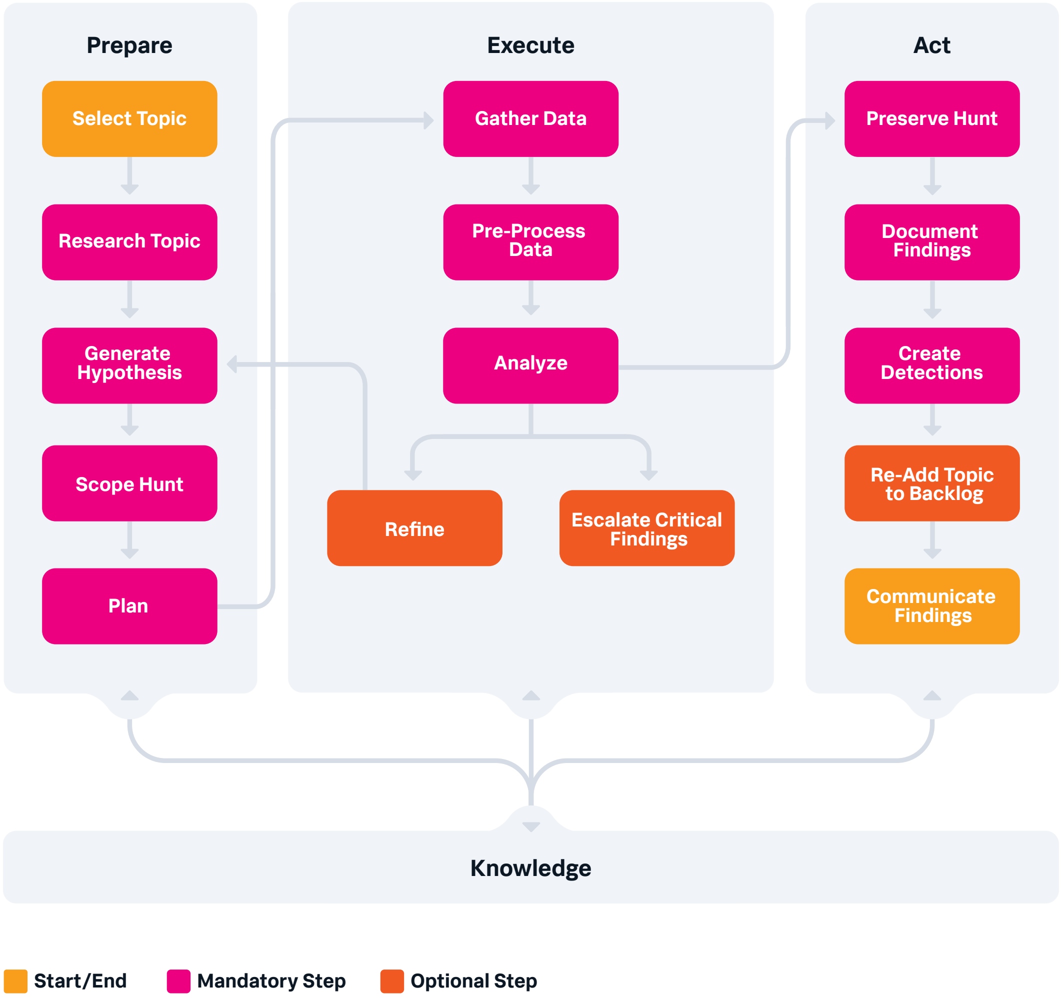
All the detailed planning and expert execution won’t matter if you can’t capture and act on the knowledge gained from your hunt. The *Act* phase focuses on documentation, automation and communication.

Crucially, each phase integrates Knowledge, which could be in the form of organizational or business expertise, threat intelligence, prior experience of the hunter(s) or even the findings from the current or previous hunts. This is a two-way integration — each phase not only uses existing knowledge, but can also create new knowledge to inform the other phases and contribute to the organization’s understanding of its own security environment.

Prepare + Execute + Act + Knowledge = PEAK

Hypothesis-driven threat hunting

This is the classic approach where hunters form a supposition about potential threats and their activities that may be present on the organization's network. Then, they use data and analysis to confirm or deny their suspicions.



Prepare

Select topic:

The first step is to choose the type of activity to hunt for. It's not a full hypothesis yet, but will be used to develop one. For example, "data exfiltration" is sufficient without yet specifying exactly how the exfiltration is accomplished.

Research topic:

Gather all of the information you can to become a subject matter expert in the hunt topic and threat actor tactics related to it. Research might include:

- Learning about typical techniques for implementing threat actor tactics
- Determining how your organization already detects a certain behavior and doing a gap analysis
- Finding example hunts that others have done that target the topic or attack tactic
- Finding out preferred tactics, techniques, and procedures for topics related to specific threat actors (your CTI team or intel vendor would be a good resource here)

Generate hypothesis:

Based on your research, craft a testable statement about the potential threats and their activities in your network. See the "Creating hunting hypotheses" section for more on creating a good hypothesis.

Scope hunt:

Define the boundaries of your investigation by identifying the systems, data and timeframes to examine. You may also want to consider setting a maximum hunt duration (e.g., "I'll hunt this for three days. If I don't find malicious activity, it's probably not occurring.") See the "Are you ABLE to hunt your hypotheses?" section for more information on using the ABLE methodology to scope your hunt.

Plan:

Combine your hypothesis, research findings and scope to outline the tools, techniques and resources you'll need to validate your hypothesis.

- What data do you need, and how exactly will you gather it?
- Which analytic techniques will you use to find the activity you're hunting for?
- If you're hunting as a team, who's doing which part(s) of the hunt?

A good plan leads to a smooth execution, so it's worth investing the time to plan ahead.

Execute

Gather data:

Collect the evidence and bring it all back into one place for analysis. In some cases, this may have already happened (for example, if you're already ingesting the logs you need into your SIEM). In other cases, you might have to identify the specific server(s) and locations on disk from which to collect the data, then copy them to the analysis system manually.

Pre-process data:

Sadly, the data isn't always quite ready for analysis, especially if you have to collect it manually. You might need to:

- Convert it to a different format (e.g., JSON to CSV)
- Convert timestamps to UTC
- Normalize equivalent logs from two different solutions into a common schema
- Throw out records with missing or nonsensical values

Taking the time to make your data clean and consistent will make the analysis much easier.

Analyze:

This is when you dive into the data to look for patterns, anomalies or other evidence that supports or refutes your hypothesis. This is where your intuition and analytical skills truly shine. There are many options when it comes to analytic techniques, including:

- Least/most frequency of occurrence
- Clustering
- Visualization

Most threat hunters pick up new analytic techniques the way mechanics accumulate wrenches. The more hunting you do, the bigger your toolbox will grow and the better you'll be at picking the right technique for the job.

Refine hypothesis:

When your analysis reveals new insights or fails to find what you were looking for, don't hesitate to revise. This is a normal and expected part of threat hunting. We don't always hit the mark the first time, so you will often need to refine your hypothesis once or twice.

Escalate critical findings:

If you find likely or confirmed malicious activity during your hunt, escalate it immediately to the incident response team for swift action. After all, time is of the essence.

Act

Preserve hunt:

Don't let your hard work go to waste. Archive your hunt, including the data, tools and techniques used for future reference or to share with other hunters. It is quite common for hunters to refer to past hunts when confronted with similar hunts later on.

Many teams use wiki pages to write up each hunt, including links to the data, descriptions of the analysis process and summaries of key findings or metrics. Some use ticket tracking systems, document repositories or other systems. The important thing is to make sure that the hunt is not lost and forgotten.

Document findings:

Write up a detailed report on your findings. Make note of whether you validated or refuted your hypothesis, data or detection gaps you found, misconfigurations you identified and incidents you escalated. These findings and the actions your security team takes to address them are key drivers for continuous improvement of your organization's security posture.

Create detections:

Convert your findings into automated detections to catch similar activity in the future. In some organizations, this step may be performed by the hunt team, while in others, it may be the responsibility of the detection engineers. Either way, using hunts to improve automated detection is the other key driver behind continuous security posture improvement.

Re-add topic to backlog:

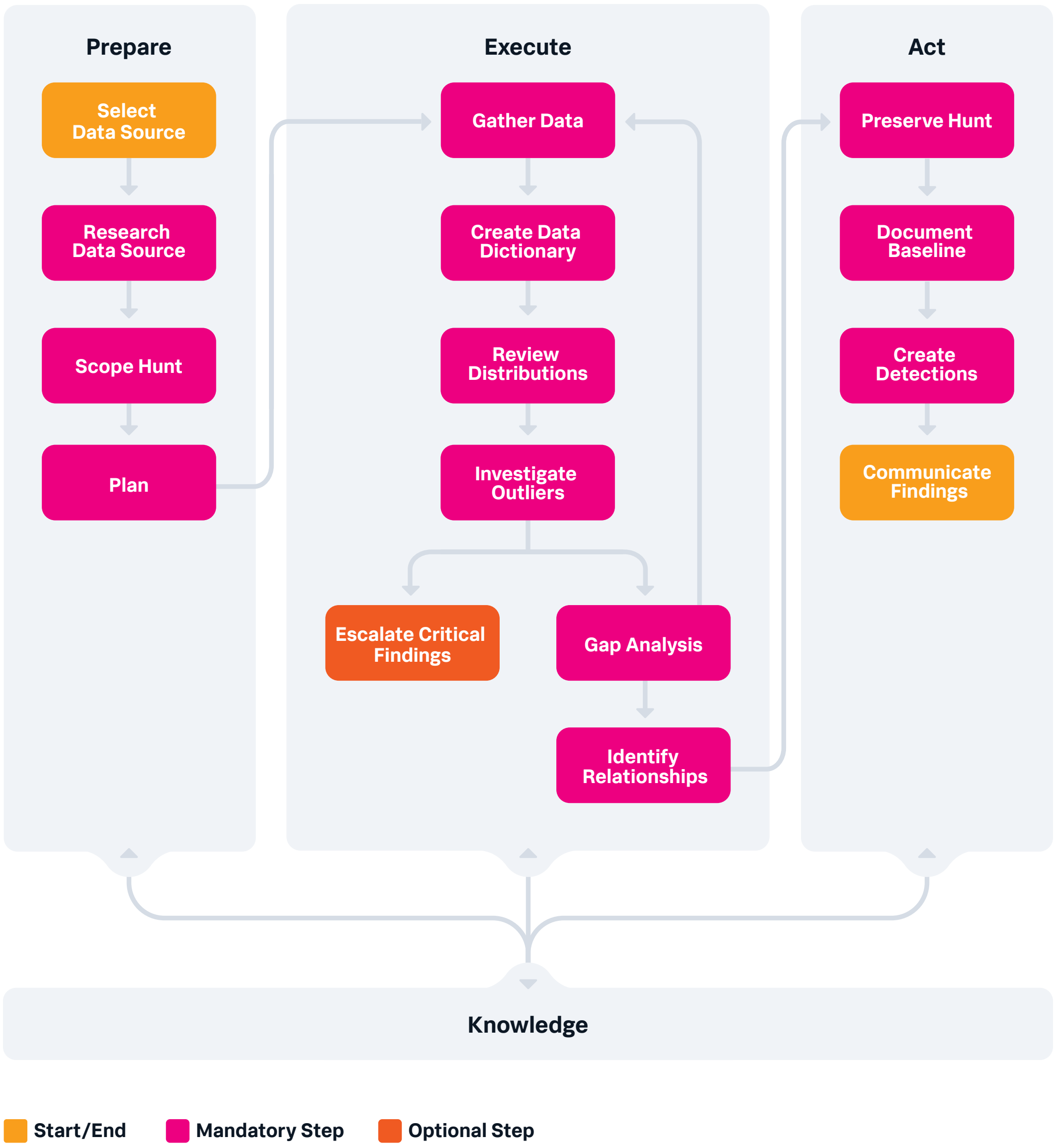
Hunters often uncover new avenues for exploration while in the middle of a hunt. Stay focused, but note those potential new ideas because they can become new topics or hypotheses for future hunting. If your team keeps a slush pile or backlog of potential hunts (and they should), add them there so you can hunt them later.

Communicate findings:

Share your discoveries with relevant stakeholders to improve overall security posture. Maybe the findings for each hunt are emailed to the SOC leadership and the owners of the systems/data involved. Perhaps you hold a hunt briefing for the security team once a month. Find the communication format that works best for both your team and your stakeholders — knowledge is most powerful when shared.

Baseline threat hunting

In this type of hunt, hunters establish a baseline of normal behavior and then search for deviations that could signal malicious activity.



Prepare

Select data source:

The first step is to decide which data source you'd like to baseline. If you're starting from square one to hunt in your organization, you should try to baseline all of your critical data sources. Start with the ones your hunt team relies on most, or maybe with the most security-relevant sources. If you're unsure about where to start, prioritize data sources according to their significance to your organization's business and detection goals.

Research data source:

Once you've determined which data source to focus on, become as familiar with it as possible. If this is a common log source that many organizations deal with, such as a Windows event log or events from a common security product, a good starting point might be to find out what the vendor has to say about what's in the data. At a minimum, do the following:

- Identify the key fields and how to interpret their values
- Research specific situations to look out for based on others' experiences with that type of data

While you're doing your research, don't forget to include any existing monitoring or detection measures implemented for that data. Also, consider consulting with the teams responsible for the systems or applications creating the data. The former can help focus future hunts, while the latter will be useful if you have questions about the logs or how to interpret them.

Scope hunt:

When conducting a hunt, it's important to narrow your focus, especially in larger environments where analyzing all of the data at once may not be feasible. Different systems may exhibit different behaviors, so it's helpful to group them based on similarities (such as "finance desktops" or "internet-facing application servers") and baseline each group individually. This approach is easier and more likely to yield better results.

Another important decision is the timeframe for data collection. Baselines are created by analyzing normal activity over a period of time, so it's essential to use enough historical data to establish what's normal. However, it's also crucial to keep the window size reasonable to avoid being overwhelmed with too much data. For most sources, between 30 and 90 days of data will be enough, though certain types of activity (e.g., "low-and-slow login failures") may benefit from longer windows.

Plan:

Using what you learned from your research, outline the tools, techniques and resources you'll need to create a baseline of your data source(s).

- What data do you need, and how exactly will you gather it?
- Which analytic techniques will you use to find the activity you're hunting for?
- If you're hunting as a team, who's doing which part(s) of the hunt?

Again, a good plan is essential for a smooth execution, so it's worth investing the time.

Execute

Gather data:

Collect the evidence and bring it all back into one place for analysis. In some cases, this may have already happened (for example, if you're already ingesting the logs you need into your SIEM). In other cases, you might have to identify the specific server(s) and locations on disk from which to collect the data, then copy them to the analysis system manually.

As part of the data-gathering process, you may also need to filter your dataset according to the system groups and/or timeline you established while scoping the hunt. Large networks may generate terabytes of data every day. Sifting through this mountain of information manually, or even with automated systems, can be daunting and time-consuming. Filter your dataset to make your analysis more efficient and manageable.

Create data dictionary:

A data dictionary is a structured description of the fields found within a data source. It provides information about their characteristics, relationships and usage. Your data dictionary should contain the following, at a minimum:

- **Field names:** The names or identifiers of the fields
- **Description:** A brief definition of each field and what they are used for
- **Data types:** The type of data each field contains (see below)
- **Field values:** How to interpret the values in each field. In other words, what each field means

When it comes to specifying the types of data for each field, here are some of the most common:

- **Numeric:** Made up of both continuous and discrete data.
  - Continuous data refers to measurements or observations that can take any value within a range, such as latency measured in milliseconds, which you might use to detect DoS attacks against a network.
  - Discrete data represents separate and distinct values, like the count of failed login attempts on a system, which can indicate a brute-force attack.
- **Categoric:** Either nominal or ordinal.
  - Nominal data refers to the names of things, such as different types of identified threats in an incident report (e.g., 'Malware,' 'Phishing,' or 'Exploit Attempt').
  - Ordinal data has an implied order, like risk ratings for the incidents (e.g., 'Critical,' 'High,' 'Medium,' or 'Low').
- **Text:** Free-form text or strings. An example could be the log messages generated by an Intrusion Detection System (IDS), which may contain valuable information for threat hunting. Most syslog messages also fall into this category.
- **Date/time:** There are a wide variety of different timestamp formats, but the UNIX epoch time or ISO 8601 (e.g., "2023-06-14T12:34:56Z") are both common. Be sure you understand what time zone these timestamps represent.
- **Boolean:** A simple binary data type. It could specify whether a security policy is active ('true' or 'false'), whether a particular port on a firewall is available ('open' or 'closed'), or whether certain security functionality is enabled ('on' or 'off').

Because data sources can often contain many different fields, you don't necessarily have to document each and every field to have a workable data dictionary. Often, it's sufficient to choose the fields that seem to be most relevant for security. For example, in a file transfer log, fields like account names, file names, transfer commands and statuses might be more useful than file sizes or average transfer rates.

Review distributions:

Use descriptive statistics to summarize the values found in each field in your data dictionary. For example, you might compute:

- The average and/or median of numeric values
- The top most common categorical values
- The number of unique values found in that field (also called the cardinality)

These statistical descriptions are the beginnings of a baseline for normal activity.

Investigate outliers:

Now that you know what “normal” looks like in your data, you can begin to identify anomalies or outliers that might indicate suspicious activity. There are many techniques for this, but here are a few of the most common:

- **Stack counting:** Also known as *stacking* or *Least Frequency of Occurrence (LFO)* analysis, this method involves counting the number of occurrences of each unique value and sorting them in ascending order. The values with the lowest counts are considered outliers. In some cases, this can be reversed, with the values with the highest counts being considered outliers, though this is relatively rare.
- **Z-scores:** When dealing with numeric values, a statistical test like z-score can be used. This test looks for values that are ± a certain threshold from the standard deviation. Typically, this threshold is either two or three standard deviations, depending on how sensitive you want to be about what is considered an outlier.
- **Machine learning:** For those who want to be fancier, machine learning techniques like isolation forests or density functions can be used. See the “Model-assisted threat hunting” section for more details.

When you find an outlier, determine whether it represents a security issue or is just a benign oddity. Also, look for multiple outliers that exhibit the same or similar characteristics — these may be related somehow, and it may be beneficial to review them as a set in addition to their individual investigations.

Escalate critical findings:

Should you find likely or confirmed malicious activity during your hunt, escalate it immediately to the incident response team for swift action. After all, time is of the essence.

Gap analysis:

As with most projects involving data, especially data you’ve never looked at before, things rarely go entirely smoothly. Gap analysis is where you identify the challenges that you ran into while hunting and take action to either resolve or work around them.

Usually, these challenges will be with the data, though in some cases, tools or analysis techniques might also present issues. For example, you may find that your initial collection somehow missed data from certain systems. If you can do without those systems, you may elect to just carry on as normal, but if those systems are critical to your hunt, you may need to revisit the “Gather Data’’ phase and gather the additional data.

Identify relationships:

Any non-trivial dataset is also likely to exhibit relationships between the values in different fields. These relationships can hold critical insights, often providing much more context about the event than you can get just by examining individual data points. A classic example is the count of user logins and how they relate to the time of day, with an increase expected during the start of the typical work shift.

Act

Preserve hunt:

Don't let your hard work go to waste. Archive your hunt, including the data, tools and techniques used for future reference or to share with other hunters. It is quite common for hunters to refer to past hunts when confronted with similar hunts later on.

Many teams use wiki pages to write up each hunt, including links to the data, descriptions of the analysis process, and summaries of key findings or metrics. Some use ticket tracking systems, document repositories, or other systems. The important thing is to make sure the hunt is not lost and forgotten when you need it later.

Document baseline:

Your baseline consists of the data dictionary, statistical descriptions, and field relationships. Document these in a way that is clear and easy for others to understand later.

Almost any large dataset will have suspicious-looking but benign anomalies. Don't forget to include a list of these known benign outliers. Recording those you already identified and investigated during the "Investigate Outliers" phase will save time during future hunts and incident investigations.

Create detections:

Distill your baseline and known outliers into automated detection. Examine each of the key fields or the common relationships you identified between fields to see if there are certain values or thresholds that would indicate malicious behavior. If so, consider creating rules to automatically generate alerts for these situations.

This may not always be feasible, so don't worry if you aren't able to identify good alerting candidates. Baselines are all about identifying abnormal activity. However, just because something is abnormal doesn't mean it's malicious. Simply alerting on any abnormal behavior will likely cause a flood of low-quality alerts. The trick with alerting is to identify outliers that are more likely than not to signal malicious behavior.

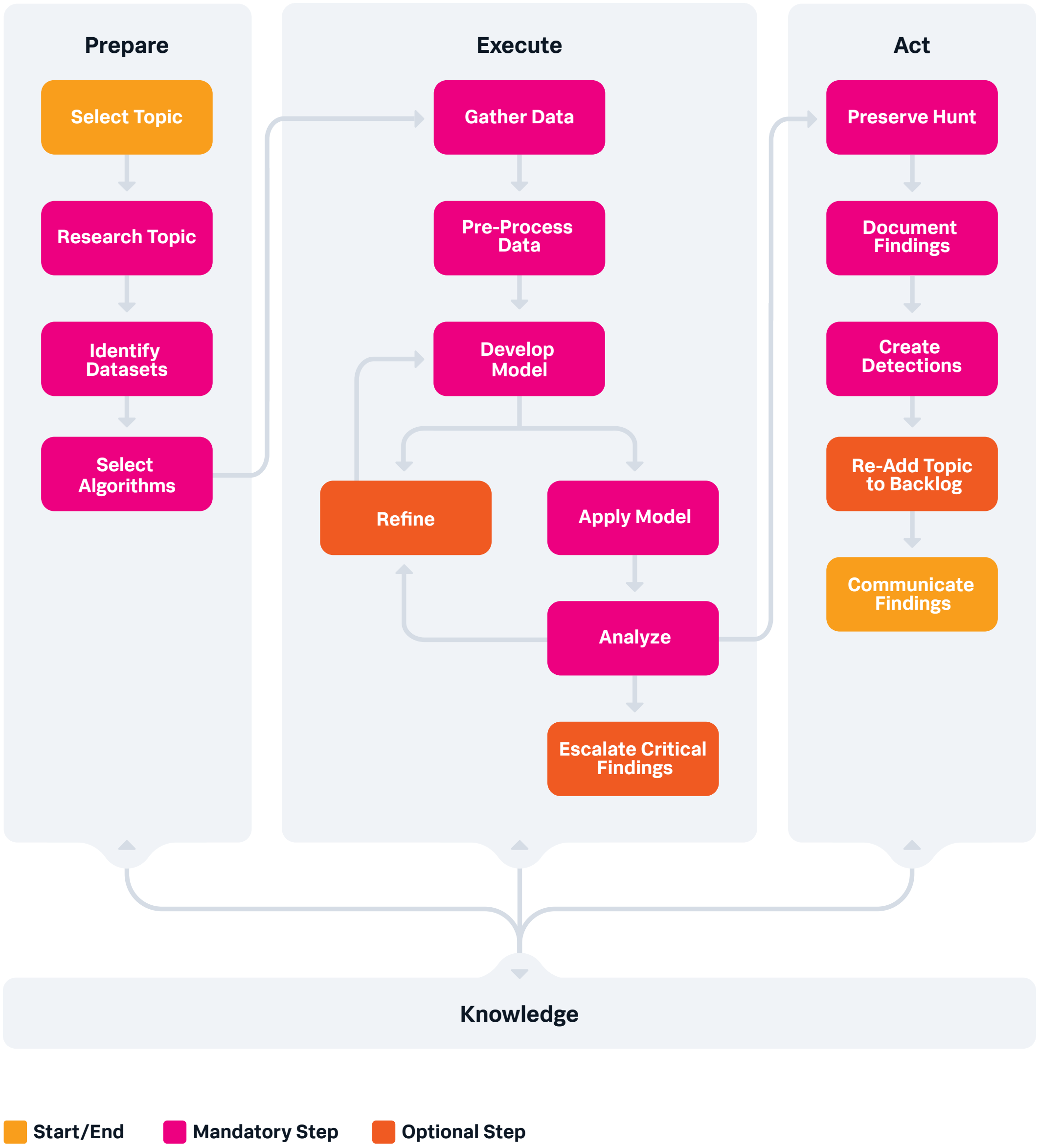
See the "Turning hunts into automated detection" section for more information on the different types of detections you might create.

Communicate findings:

As with all types of hunts, baselines are most impactful only when you share them with relevant stakeholders to improve your overall security posture. In addition to sharing with the owners of the system you baselined, ensure that your SOC analysts, incident responders and detection engineers are aware that the baselines exist and can access them easily. Consider linking to the baselines from the playbooks that your SOC analysts use to triage alerts.

Model-assisted threat hunting (M-ATH)

M-ATH can be described as a data science process wrapped in a threat hunting process, and focuses on applying algorithms to help find leads for threat hunting. For example, hunters can use machine learning (ML) techniques to train models to recognize malicious or identify suspicious behavior. Think of this as a hybrid of the hypothesis-driven and baseline hunt types but with substantial automation from the ML.



Prepare

Select topic:

Your topic can be an exploratory question or hypothesis about your data that requires a more advanced analysis method. It should also include some consideration of your modeling approach. For example, hunting a targeted adversary behavior may be a fit for supervised classification. Exploring differences in user groups may be a hunt you can perform with unsupervised clustering. If you're not sure which model-assisted methods apply, you can double check, or refine your ideas during the next step.

Research topic:

Gather all of the information you can to become a subject matter expert in the hunt topic and related threat actor tactics. Research might include:

- Diving into the literature, comparing existing detection methods and their accuracy
- Checking for existing open source information, models, code and datasets
- Determining how your organization already detects a certain behavior and conduct a gap analysis

Identify datasets:

Understanding the methods and resources available to hunt for threats will help you understand the level of effort that may be required, and the possible approaches. For supervised learning approaches, your data should be properly labeled (i.e., classification categories or some other source of "ground truth" is present). In many cases, the ideal dataset will not exist and you may have to create it. In any case, available datasets can be useful for prototyping a model before testing against your own data, or helping to validate your approach after development.

Select algorithms:

Algorithm selection includes a broad category of choices. Some examples of the most popular families of algorithms include:

- **Classification:** Classifier algorithms predict the value of a categorical field (e.g., "malicious" or "benign" status, malware family names, etc.). Classification is a supervised learning method.
- **Clustering:** Clustering is the grouping of similar events and can be especially useful when looking for outliers in large, multi-dimensional datasets (i.e., sets with many data points). Clustering is an unsupervised learning method.
- **Time series analysis:** Time series analysis involves analyzing a sequence of events and their associated timestamps to forecast future values. In threat hunting contexts, time series analysis is most likely to be used as an outlier detection mechanism by looking for large deviations between predicted and observed values. Time series analysis is an unsupervised learning method.
- **Anomaly detection:** Anomaly detection algorithms use statistical approaches to find outliers in numerical or categorical fields. While simple statistical tests like the z-score can be used for numeric data, other model-assisted methods are available, like Autoencoders, clustering methods or single-class Support Vector Machines.

Even within the same family, different algorithms may yield significantly different results. It is common to compare the performance and accuracy of several algorithms during the "Develop Model" step to select whichever one(s) work best.

Execute

Gather data:

Collect the evidence and bring it all back into one place for analysis. In some cases, this may have already happened – such as if you're already ingesting the logs you need into your SIEM. In other cases, you might have to identify the specific server(s) and locations from which to collect the data, then copy them to the analysis system manually.

Pre-process data:

Sadly, the data isn't always quite ready for analysis, especially if you had to collect it manually. You might need to:

- Convert it to a different format (e.g., JSON to CSV)
- Convert timestamps to UTC
- Normalize equivalent logs from two different solutions into a common schema
- Throw out records with missing or nonsensical values
- Encode categorical data into numeric fields
- Label data

Taking the time to make your data clean and consistent will make the analysis much easier.

Develop model:

Developing your model typically includes running, comparing and tuning multiple model options for your target problem. This is a very open-ended process — you may need to evaluate several different approaches or algorithms before you settle on a workable model.

Refine:

When your analysis reveals new insights or fails to find what you were looking for, don't hesitate to revise until the model meets both your performance and your accuracy requirements. Revisions may include adjusting algorithms, incorporating new or different fields from the dataset (the data's features), or tuning hyperparameters. Ultimately a successful threshold for model performance is up to your team. For example, you may need to consider trade-offs in model accuracy for completeness, depending on the sensitivity of the results and the scale of results your team can manage.

Apply:

When your model has reached an acceptable level of performance and accuracy on test data, apply it against your full hunting data set to create a list of suspicious events or activities.

Analyze:

Investigate the suspicious activity to determine what's likely benign and what could genuinely be malicious. Analysis may include more traditional hunting methods, such as filtering, sorting, stacking or clustering. It can also involve enriching data with external sources to provide additional context. If there are many false positive findings, consider labeling them, adding them to your training data and then revisiting the "refine" step.

Escalate critical findings:

If you find likely or confirmed malicious activity during your hunt, escalate it immediately to the incident response team for swift action. Time is of the essence.

Act

Preserve hunt:

Don't let your hard work go to waste. Archive your hunt, including the data, notebooks, trained models, tools and techniques used for future reference or to share with other hunters. It is quite common for hunters to refer to past hunts when confronted with similar hunts later on.

Many teams use wiki pages to write up each hunt, including links to the data, descriptions of the analysis process, and summaries of key findings or metrics. Some use ticket tracking systems, document repositories or other systems. For M-ATH hunts in particular, source code repositories such as GitHub may be useful. The important thing is to make sure that the hunt is not lost and forgotten.

Document findings:

Write up a detailed report on your findings. Include whether you validated or refuted your hypothesis, the data or detection gaps you found, misconfigurations you identified, and incidents you escalated. These findings and the actions your security team takes to address them are key drivers for continuous improvement of your organization's security posture.

Create detections:

The best models may result in worthwhile detections. This is a best-case scenario, and you may decide these should run on a regular cadence to produce detection alerts. In other more common situations, your model may get you 80% of the way there. If the results aren't reliable enough for alerts, enrich the data and/or put an analyst in the loop via a repeatable report, a dashboard or a M-ATH playbook to close that remaining 20%.

See the "Turning hunts into automated detection" section for more information on creating automated detection out of M-ATH hunts.

Re-add topic to backlog:

Hunters often uncover new avenues for exploration while in the middle of a hunt. Stay focused, but note those potential new ideas because they can become new topics or hypotheses for future hunting. If your team keeps a slush pile or backlog of potential hunts — and they should — add them there so you can repeat them later.

Communicate findings:

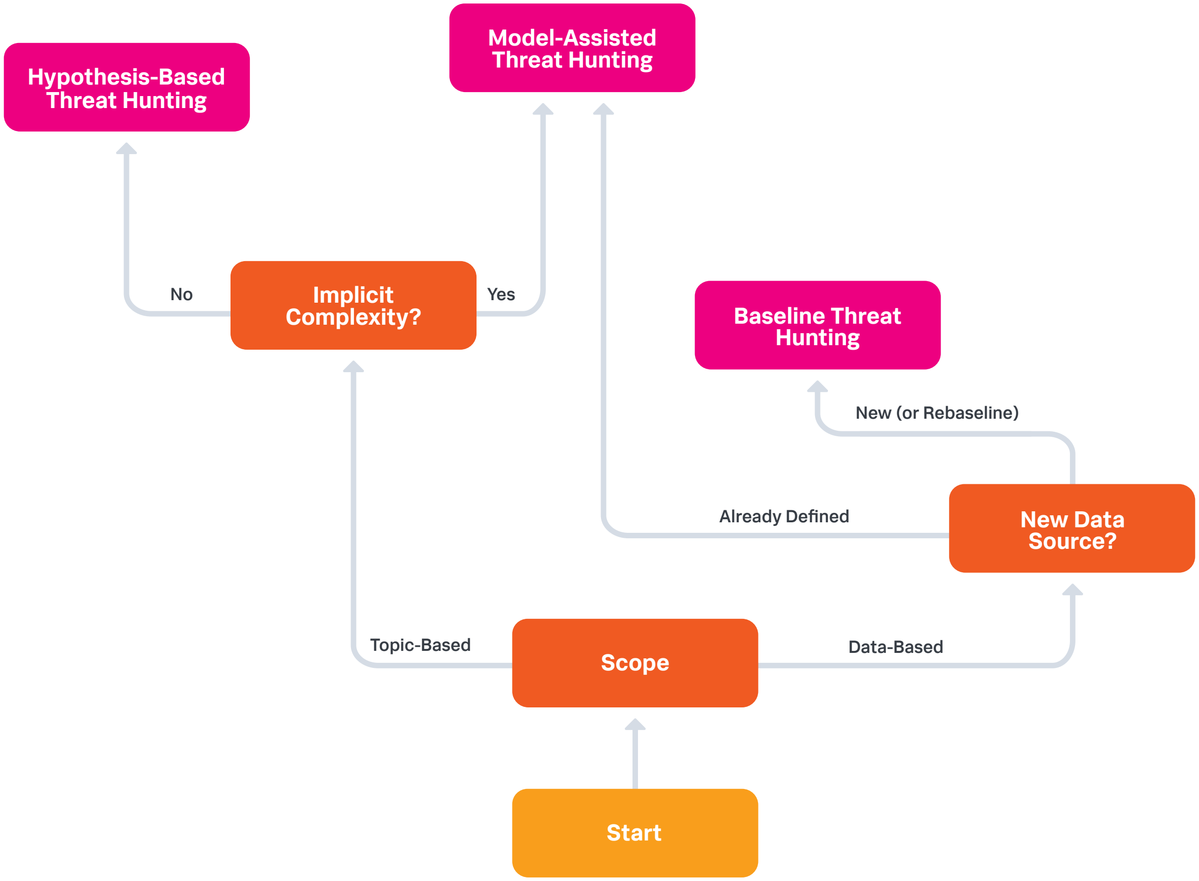
Share your discoveries with relevant stakeholders to improve your organization's overall security posture. Maybe the findings for each hunt are emailed to the SOC leadership and the owners of the systems/data involved. Perhaps you hold a hunt briefing for the security team once a month. Find the communication format that works best for both your team and your stakeholders — knowledge is most powerful when shared.

Choosing the best path

Choosing the best type for a specific hunt may sound tricky at first. Each has its own strengths and weaknesses, and it's important to select the right type for the task at hand. Ask and answer some key questions first to make the choice much more straightforward.

Some points to consider:

- "Implicit Complexity" exists when the data has many variables with complicated relationships, or when programming an explicit algorithm might be challenging or impractical. Most problems with implicit complexity are good candidates for M-ATH hunts.
- Periodically re-baselining data sources is useful to ensure there is no drift or changes that have impacted the validity or completeness of the data source or the baseline.
- Forecasting or time-series analysis can be used to perform a M-ATH hunt using an existing baseline.



Creating hunting hypotheses

Hypothesis-based hunts rely on a central hunch or educated guess that guides your investigation. These hypotheses are based on a combination of a hunter's intuition, experience and research. Crafting a solid hunting hypothesis requires a delicate blend of creativity and analytical thinking.

There are three steps to creating a good hypothesis:

1. **Select a topic:** Identify a specific behavior of concern. Draw on your understanding of the threat landscape, recent incidents and emerging trends to pinpoint potential risks inside your network. Research some of the priority threat actors targeting your organization or your industry to identify their typical behaviors. However you go about it, the first step is to figure out what type of activity you want to look for.
2. **Make it testable:** Write the topic as a statement or assertion that can be either proved or disproved. A hypothesis that can be disproved is said to be falsifiable. If the hypothesis is not falsifiable, it is not a valid hypothesis.
3. **Refine as necessary:** Restate and rescope your hypothesis until it is falsifiable and you are certain that you can hunt it, given your timeframe and the resources available to you. Don't be surprised if you need to refine your hypothesis even during the middle of the hunt.

Hypothesis generation can be tricky at first. If you get stuck, this paper on crafting hypotheses or this one on identifying priority targets can give you some ideas on where to start.

Example: Data exfiltration hypothesis

Here is an example of using the above process to create a good hunting hypothesis.

1. **Select a topic:** Assume you are concerned about the theft of sensitive information from your network. Your topic would be "data exfiltration."
2. **Make it testable:** It probably isn't feasible to look for all possible types of exfiltration, so rewrite a hypothesis for a specific type of exfiltration. For example, "A threat actor may be exfiltrating sensitive data using DNS tunneling."
3. **Refine as necessary:** Large organizations might still consider this hypothesis to be somewhat untestable because there are so many different types of data to steal across the enterprise. You could further refine this to: "A threat actor may be exfiltrating sensitive financial data using DNS tunneling." This hypothesis may be more practical because you've narrowed down the type of data you suspect might be leaving your network.

Are you ABLE to hunt your hypothesis?

Even when you have a clear and testable hypothesis, you still need to know a few things before you can start hunting. You need to know possible indicators of the activity, data source(s) you need to examine, and which parts of the network the activity is happening in.

PEAK incorporates the ABLE method to help you capture the critical pieces of your hunting hypothesis, which include:

Actor:

The threat actor (or sometimes the general type of threat actor) that you are looking for. Many behaviors are not tied to a specific actor, so you won't always need to specify this part. But if you do, it can supply valuable context to help with the rest of your hunt.

Behavior:

The specific activity you're trying to find — sometimes called TTPs (Tactics, Techniques, and Procedures). Instead of hunting for an entire attack lifecycle's worth of behavior, focus on one or two pieces at a time.

Location:

The part(s) of your organization's network where you would expect to find the behavior (e.g., "end-user desktops," "internet-facing web servers," or even just "internal" versus "perimeter"). A proper location helps narrow the scope of your hunt, making it easier and more efficient.

Evidence:

A combination of which data source(s) you'd need to consult to find the activity and what it would look like if it were present. You'll need to know these when you plan your data collection and create your analysis strategy.

Example: Using ABLE to hunt DNS exfiltration

Consider the following hypothesis: "PIFFLING PANGOLIN may be exfiltrating sensitive financial data using DNS tunneling." Here's how you might break this down using the ABLE framework.

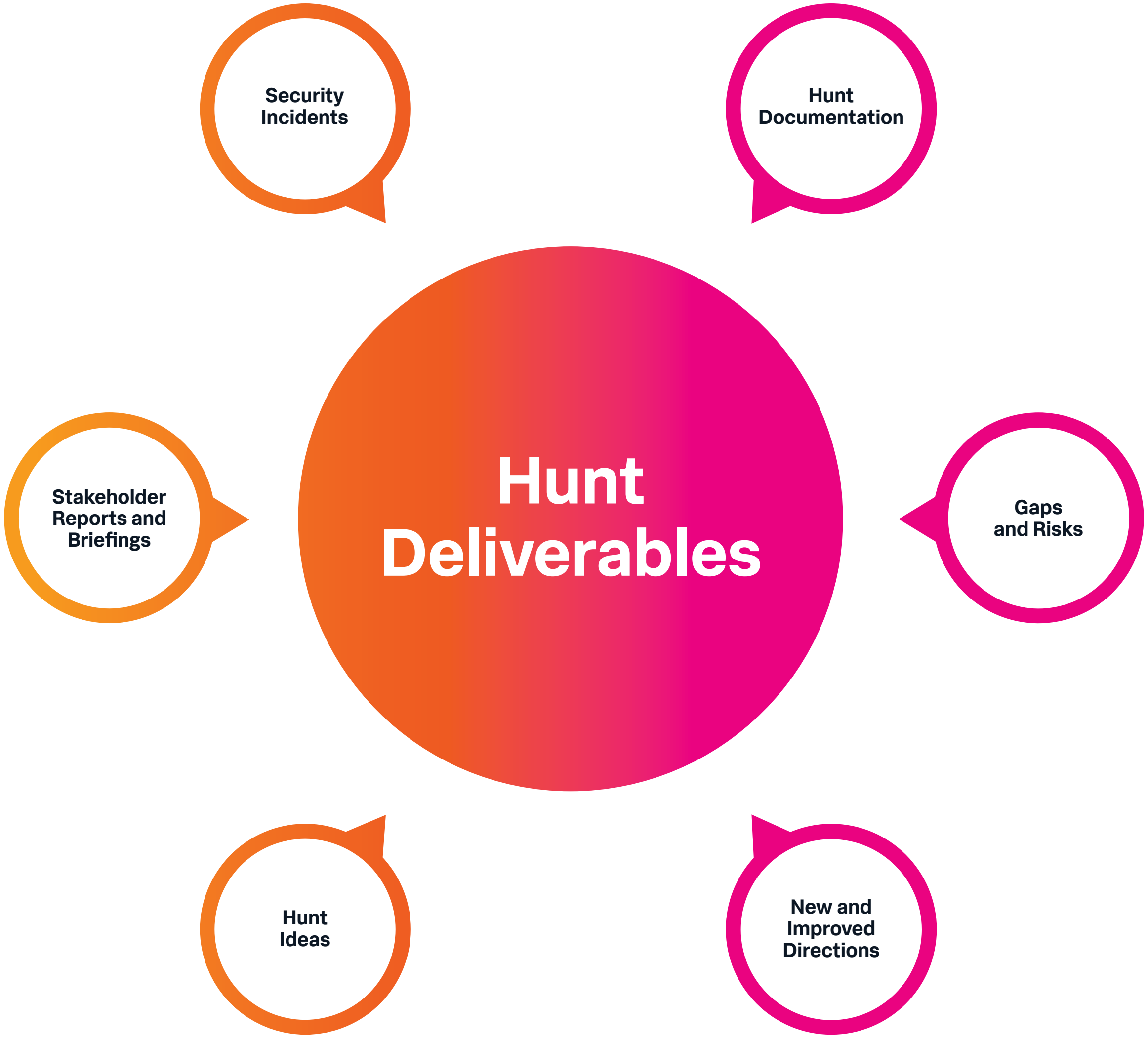
- **Actor:** Although financial data would appeal to many cybercriminals, this hypothesis is about a specific group. Understanding this actor's typical operations may give you clues to aid in your hunt, such as known C2 domains or specific tools they use for DNS tunnels.
- **Behavior:** The behavior you're hunting for is data exfiltration through DNS tunneling. By focusing on this specific tactic, you can narrow down your investigation and concentrate on relevant indicators of compromise.
- **Location:** The hypothesis suggests that the finance department is being targeted. This pinpoints the parts of the network that you need to scrutinize — the finance network for the source of the data and the network perimeter for the internet-based exfiltration.
- **Evidence:** To detect DNS tunneling, examine DNS query logs or full passive DNS logs. Look for unusually large or frequent DNS queries, odd DNS query types or indicators of known DNS tunneling tools.

With the ABLE framework applied to your hunting hypothesis, the outline of an actionable hunt plan becomes clear:

- **Gather DNS logs** for hosts associated with the finance department (user desktops and servers hosting financial applications).
- **Look for unusually large queries or responses** since those are typical indicators of DNS tunneling.
- **Identify the DNS record types associated with "normal" traffic** and investigate any queries involving unusual record types.
- **Research existing DNS tunneling tools** and look for their unique network or host artifacts. See the Pyramid of Pain for more on detecting tools and their artifacts.

Threat hunting deliverables

Each PEAK hunt type has an *Act* phase, in which different types of hunting outputs or deliverables play key roles. This section includes some of the most common types of hunting deliverables. Depending on your organization's requirements, there may be others, but this is the minimal set that most hunting programs should produce.



Security incidents

You will often find suspicious or malicious activity while hunting — this should be escalated immediately. Include the information you have already gathered and explain why you believe this might be a security incident. This will help your SOC get a head start on the investigation and will also foster good relationships between your teams. The information you provide could be as simple as gathering user, host and surrounding activity information. Anything you can do to help support your case will benefit the SOC analysts.

Hunt documentation

As you progress through your hunt, take notes about the following:

- What you're doing
- Why you're doing it
- What the results were
- Your analysis and interpretation of the results

Details matter. Hunts should be repeatable by other hunters. The queries executed, data results and even screenshots are quite helpful when going back through what took place over the course of your hunt. This information can be used for re-hunting or training up new threat hunters.

Add an executive summary of everything you did during your hunt as well. Be sure to include:

- Key findings
- Whether you proved or disproved your hypothesis

Take your work even further and develop knowledge base articles or wiki pages to share with your teams so that everyone has access to them during future hunts or incident responses.

New and improved detections

As you execute a hunt, you're figuring out how to find certain types of exact malicious activity. You may find new methods or improvements to existing detections you already have in place. Detections that are developed based on your specific organization and environment are often more valuable than vendor-provided out-of-the-box detections. The types of detections you create may vary and don't always have to be signatures.

See the "Turning hunts into automated detection" section for more on determining the best types of detections for your hunt.

Gaps and risks

Threat hunting involves examining all kinds of data and activity in your organization. Throughout this process, you may encounter gaps in your people, processes, tools or data. These gaps could be risks that your company is taking, perhaps unknowingly. Here are some examples of these risks:

- A particular data source is missing an event type that is valuable for monitoring.
- Multi-factor authentication (MFA) is not turned on in parts of your organization.

When you find gaps and risks like these, be sure to share them with the appropriate stakeholders for remediation.

Hunt ideas

One of the most difficult challenges with threat hunting is staying in scope. Often you come across intriguing events that lead you down a rabbit hole of an investigation. But watch out for scope creep — any out-of-scope hunting will impact your hunt timeline.

To hunt efficiently, you may need to revise your hunt hypothesis to make the current hunt more feasible. Even if you adjust the hunt scope, you don't want to overlook the out-of-scope work. Add those additional hunt ideas to your topic backlog for future hunting.

Stakeholder reports and briefings

Sharing your hunt findings is an important part of the Act phase. Teach what you have learned as part of your hunt to help maximize the value of your efforts and provide others with information that they can apply to their own work. Schedule technical readouts with all your security stakeholders, but especially blue team groups such as:

- Threat Hunt
- Detection Engineering
- Threat Intelligence
- SOC
- Incident Response

Don't forget to include the owners or administrators of the tools and logging you included in your hunt. For example, if you were hunting around a high-priority web application, include the data owners, application owners, and administrators of the applications and the system(s) it runs on. Talking with other teams could potentially spark new ideas for hunts or detections!

Consider scheduling regular executive readouts with your leadership team. Highlight your hypotheses, high-level analysis processes and of course, any incidents or other important findings. This gives them a closer look at how threat hunting benefits your security posture.

You don't need a separate readout for each hunt, especially if you have an aggressive hunting cadence. Another option is to create a highlight reel of recent hunts, perhaps monthly or even quarterly. You can even experiment with the format, such as an email that shares the most recent wins for the team. This helps shine a light on all your hard work and encourages stakeholders to act on your findings.

Turning hunts into automated detection

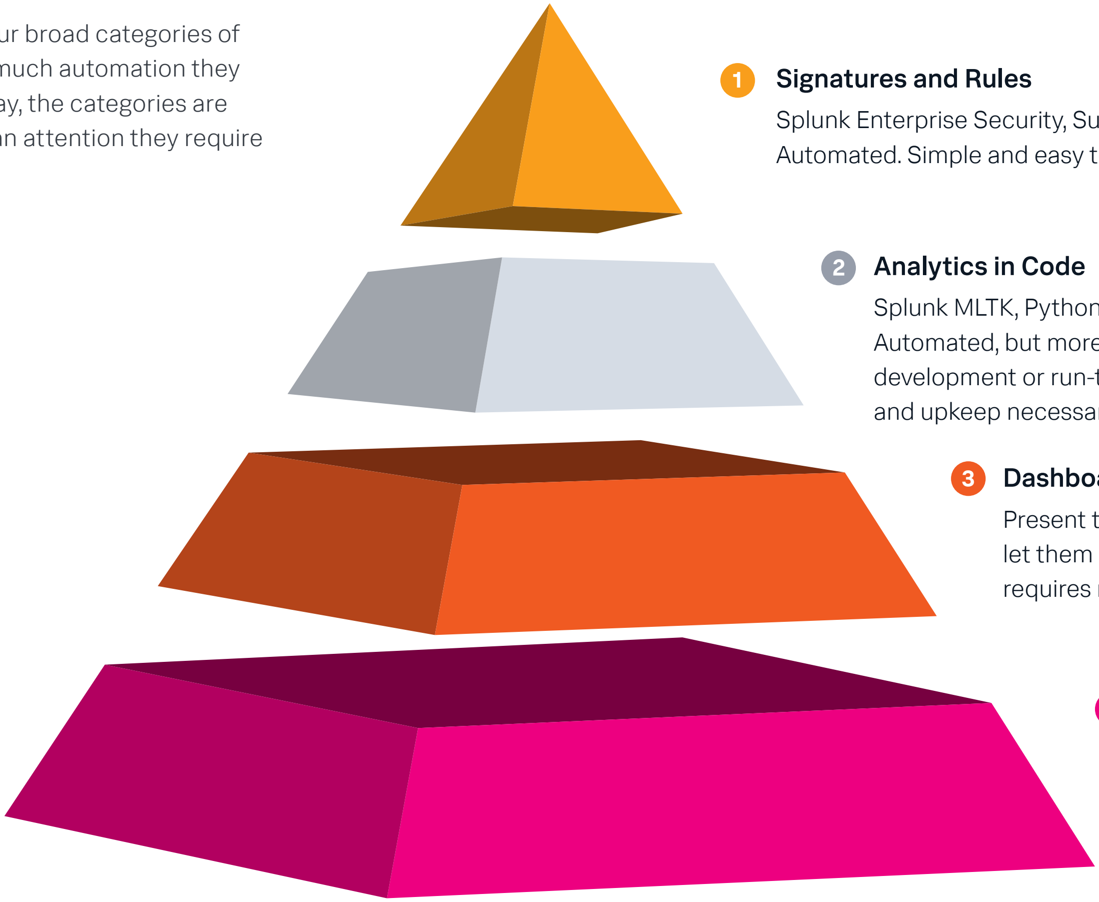
As mentioned in the "Why Hunt?" section earlier, the purpose of threat hunting isn't just to find security incidents. Finding incidents is a helpful side effect of hunting, but the best reason to do it is to drive improvement to your security posture over time.

Improvement can take many forms, but probably the most obvious is enabling the implementation of new methods to detect malicious activity. However, finding these new ways isn't very useful unless you take the next logical step: Turn your hunts into automated detections so you don't have to spend time and energy hunting them repeatedly.

Creating detections can be as easy as writing a new SIEM rule, but what if your new method isn't quite so straightforward? That's where the PEAK Hierarchy of Detection Outputs can help. The hierarchy is a model for understanding the different types of detections you can create from your hunts and when each type might be most appropriate.

The hierarchy of detection outputs

The hierarchy consists of four broad categories of detection, ordered by how much automation they offer. Or, to put it another way, the categories are ordered by how much human attention they require to find the bad things.



1 **Signatures and Rules**

Splunk Enterprise Security, Suricata IDS, etc. Automated. Simple and easy to manage.

2 **Analytics in Code**

Splunk MLTK, Python, or other algorithmic detections. Automated, but more complex. May require significant development or run-time resources. More management and upkeep necessary.

3 **Dashboards and Visualizations**

Present the distilled info to humans and let them decide. Some automation, but still requires manual review on a regular basis.

4 **Reports**

Saved searches with little or no pre-processing. Human must spend significant effort to extract meaning.

The hierarchy consists of the following levels, starting from the bottom:

- **Reports:** Sometimes you know what you want to find, but searching for it will likely return many irrelevant results, such as authorized benign activity or other false positives. Finding relatively rare examples of true malicious intent may require analyst experience and insight. In this case, an automated report might be the best option for detection. Run it on an appropriate schedule (daily, weekly, monthly, etc.) and put it into the analysts' ticket queue or use some other mechanism to ensure that someone actually reviews it.

  Reports are the lowest level of the hierarchy and thus the least desirable because they require the most human attention. They're not ideal, but due to resource constraints and the potential volume of non-malicious results, sometimes they are the best you can do. Hunts that result in report detections might make good candidates for additional detection engineering or even re-hunting in the future. Once you have more experience with the reports, you might find better ways of focusing on the relevant results and climbing up the hierarchy.

- **Dashboards and visualizations:** These present summarized information instead of just raw details, and are much easier for humans to understand. Although dashboards and visualizations still require regular human attention and likely have many benign entries, they are designed to make it easy for analysts to quickly pick out the important things.

  Like reports, you need to ensure that analysts review dashboards on a regular schedule. Also, like reports, dashboard detections could be refined in the future to move up the hierarchy.

- **Analytics in code:** When you know how to find the exact activity you're looking for, but it requires outside computation to make it work, you're dealing with what we call Analytics in Code. That is, you have created a script or program to analyze the data and find the targeted activity. This is especially common with model-assisted hunts but can occur with any hunt. Ideally, the output of this code will be highly accurate and suitable for inserting into your SIEM as an alert.

  Although this level in the hierarchy is perfectly respectable from the standpoint of automation and alerting, you may need to devise a mechanism to ensure that the code runs regularly and reliably.

- **Signatures and rules:** These are at the top of the hierarchy because they require no human input to generate alerts. They are also typically easier to manage than analytics written in code. As a result, they are the preferred choice for automated detection when feasible.

Applying the hierarchy

Applying the hierarchy to your hunts is usually straightforward. After completing each hunt, ask the following questions:

- **Have I figured out how to find this activity?** Don't worry if you didn't find any malicious activity. This is common. You can have a successful hunt even if you didn't find what you were looking for, as long as you are satisfied that you would have found it if it were present.
- **When I have a result, would I be confident enough to put it in front of a SOC analyst as an alert?** If yes, then you'll probably want to use either Analytics in Code or Signatures and Rules. Otherwise, Reports or Dashboards and Visualizations are likely the best choices.

Choose the highest level in the hierarchy that you can feasibly implement. Dashboards are better than reports, and rules are better than code. But don't worry too much about making the "right" choice. Do what seems best — you can always revisit your selection if it doesn't work out. In fact, you should revisit the reports and dashboards occasionally just to see if you can figure out how to move those detections to a higher level.

Also, remember that you may discover more than one potential detection mechanism for any given hunt. Therefore, you may create detections at different levels in the hierarchy.

Key threat hunting metrics

As organizations struggle to keep up with attackers in the cybersecurity arms race, continuous improvement is a strategic priority. An effective threat hunting program is one of the best ways to drive positive change across an organization's entire security posture, but both hunters and their leaders often struggle to define what "effective" means for their programs.

The PEAK Threat Hunting Framework provides a set of key metrics that you can use as a starting point to measure the impact of hunting on your security program. It also incorporates a maturity model, which leaders can use to assess the current state of their hunting program and figure out how to get where they would like to be.

The PEAK metrics philosophy

One of the most common ways to measure a hunt is by the number of new incidents opened during that hunt. Though this is the most obvious measure, it's not a strong indicator of success. Hunters don't control adversary actions or timing, and just because the thing they hunted for wasn't happening when they looked for it doesn't mean that the hunt failed. Hunts that find no evidence of malicious activity can still be successful.

In very broad terms, you have two choices regarding metrics. You can:

- Measure what you've done
- Measure the effect of what you've done

It's the difference between saying "We performed nine hunts this quarter" and "We put twelve new automated detections into production this quarter to find cloud exfiltration activity that we previously missed." The number of hunts that you completed is interesting information, but it's just a measure of how hard your team worked.

The purpose of hunting isn't just to work hard — it's to drive continuous security improvement. Therefore, the number of new and updated detections created is much more relevant. It shows just how much you were able to strengthen your security posture as a result of hunting. PEAK key metrics are designed to help you measure improvements this way across several important facets of your organization's security posture.

PEAK key metrics

There are five measures of security impact that every hunting program can benefit from tracking, though they are by no means the only metrics you should collect. Rather, consider them a good starting point that any organization can benefit from. Organizations can and should develop their own metrics as well.

In no particular order, the metrics are:

The number of detections created or updated

The number of new detections created due to your hunting efforts is one of the most useful ways to track improvements to automated detection. Throw in the number of existing detections you improved (e.g., decreased false positives or added detections for edge cases), and you begin to tell the story of how you're driving improvement to automated detection.

The number of incidents opened during or as a result of the hunt

Simply tracking the number of new incidents opened during your hunts is not a great metric. You may have a great hypothesis or a killer machine learning approach to find the bad guys, but if they were not active in your network recently, you wouldn't open any incidents during your hunt.

A better approach would be to track the number of new incidents opened both during and after the hunt, due to the detections you created or improved. This requires you to track which detections came from which hunts, but it will give you a clear view of how many security incidents you caught, and hopefully mitigated before they became breaches, due directly to your hunting efforts.

The numbers of gaps identified and gaps closed

As you go about your hunting, you will inevitably notice that some things are missing. Gaps can occur in data visibility, access, documentation or even the tooling required to make those things work. Identifying these gaps and either closing them or bringing them to the attention of the responsible parties is an important function of the hunt team. Although the hunt team is probably not the team that's responsible for closing most gaps, tracking both the number reported and the number eventually closed as a result of your reports is another great way to measure your impact.

The number of vulnerabilities and misconfigurations identified and the number closed

The more you start poking around in new areas, the more opportunities for improvement you'll find. As mentioned above, many of these may be gaps of some sort that need to be closed, but you will also frequently encounter misconfigurations, older software versions or other situations that create actual vulnerabilities. Reporting these is critical, but so is tracking your reports and the vulnerabilities eventually remediated as a result of your reporting.

Techniques hunted by ATT&CK, Kill Chain, or Pyramid of Pain

Variety is the spice of both life and hunting programs. You're hunting for different types of activity each time, so be sure to track what you're hunting for. The simple way to do this is to count the number of unique techniques you're looking for (e.g., "We hunted nine different adversary techniques this quarter.") But, also consider matching this up with another mental model such as MITRE ATT&CK, the Lockheed Martin Cyber Kill Chain or even the Pyramid of Pain.

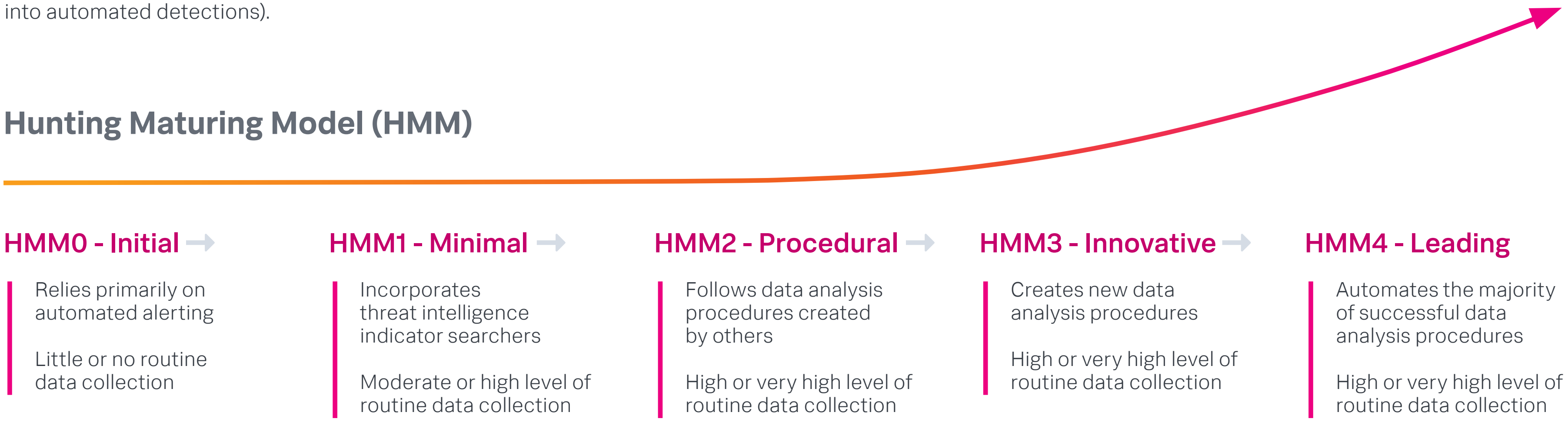
For example, tracking your hunts against the Kill Chain can give you an idea of your potential impact because targeting and detecting activity closer to the beginning of the chain gives you more time to intervene before the adversary can complete their attack. Tracking against ATT&CK can help you identify and prioritize new hunt topics, though you should remember that not everything in ATT&CK is intended to be turned into a detection. These models represent different ways of looking at your hunt topics, so you may even want to measure against more than one.

Measuring program maturity

First published by David Bianco in 2015, the Hunting Maturing Model (HMM) gives CISOs and other hunt leaders a simple way to measure the maturity of their threat hunting programs. It considers the three key measures of a hunting program (data collection, data access and the hunters' analysis skills) and reduces these to a single maturity number, HMM0 - HMM4.

Not only is the HMM useful to gauge a program's current maturity, but it also serves as a roadmap for how to get from where you are today to where you'd like to be. Choose your desired maturity level, note the differences between where you are now and where you'd like to be, and plan for the improvements necessary to achieve the goal. Most hunting organizations should be at least at HMM2 (high level of data collection, generally following hunts designed by others), though many will opt to go for HMM3 (creating novel hunt procedures, perhaps with M-ATH), or, ideally, HMM4 (turning the majority of their hunts into automated detections).

Hunting Maturing Model (HMM)



HMM0 - Initial →

Relies primarily on automated alerting

Little or no routine data collection

HMM1 - Minimal →

Incorporates threat intelligence indicator searchers

Moderate or high level of routine data collection

HMM2 - Procedural →

Follows data analysis procedures created by others

High or very high level of routine data collection

HMM3 - Innovative →

Creates new data analysis procedures

High or very high level of routine data collection

HMM4 - Leading

Automates the majority of successful data analysis procedures

High or very high level of routine data collection

Adopting PEAK in your hunting program

Organizations looking to start new threat hunting programs can use the PEAK framework as a comprehensive guide for building an effective program from scratch. For teams with existing hunting practices, PEAK provides numerous enhancements that can be adopted incrementally to bolster and mature current efforts.

New threat hunting programs

For new teams, PEAK delivers a complete methodology to follow when constructing a program. PEAK defines three primary types of hunts: hypothesis-driven, baseline and model-assisted. New teams can adopt this hunt classification and use PEAK's detailed guidance on performing each type as the foundation for their workflows.

PEAK prescribes a three-phase structure for hunts: *Prepare*, *Execute* and *Act*. Each phase has specific steps and considerations outlined in the framework, which teams can follow to ensure they have robust procedures for performing their hunts. For example, the Act phase highlights the importance of consistent hunt documentation, creating automated detections and communicating findings to stakeholders.

PEAK also provides recommendations on deliverable types that should be produced, including hunt write-ups, identified gaps or risks and ideas for future hunts. Tracking and delivering these tangible outputs helps quantify the impact of hunting and maximize its value.

To measure effectiveness, PEAK defines five key metrics that new programs should incorporate from the outset: detections created/updated, incidents opened, gaps identified/closed, vulnerabilities identified/closed and techniques hunted. Monitoring these metrics provides crucial insight into how well hunting improves security posture.

Finally, new teams should assess their maturity using the Hunting Maturity Model to establish a baseline and set goals for achieving higher maturity levels in both short- and long-term timeframes. This will help drive the continual enhancement of capabilities.

Enhancing existing programs

For established teams, aligning an existing program with PEAK can be easy and painless. View the framework as a blueprint for incremental enhancements to your existing processes and procedures. You do not have to do everything all at once. Pick one or two areas to focus on first, then follow up with additional pieces of the framework when those are complete.

One simple yet highly effective technique is to evaluate your team's current types of hunts and workflows in the context of PEAK's guidance. This comparison will illuminate any gaps where existing practices may deviate from PEAK recommendations. Remember, PEAK is designed to be flexible and customizable, so you don't always have to follow these procedures exactly. But when you deviate from them, do it intentionally so you can still achieve comparable results.

PEAK's metrics provide another avenue to expand the measurement of success beyond any existing tracking. Incorporate them into regular reporting to spotlight the areas where hunting has the most impact and show opportunities to expand that impact into other parts of your overall security program.

As you gradually realign processes with PEAK, periodically reassess your maturity level using the included model. Any gaps between your current and target states will inform priorities for maturing along the PEAK methodology.

For any program

For both new and existing programs, the PEAK framework offers a model that can be adapted and integrated in whole or in part to strengthen threat hunting practices. Focus first on high-value areas and evolve over time for maximum impact. Leveraging PEAK this way will help your team get up to speed quickly, while also giving you some quick but substantial wins when it comes to improving your organization's security posture.

Security at Splunk is a family business. Credit to authors and collaborators: David Bianco, Ryan Fetterman, and Sydney Marrone. Learn more about how the SURGe Security Research Team at Splunk provides research to reinforce your blue team.

Splunk, Splunk> and Turn Data Into Doing are trademarks and registered trademarks of Splunk Inc. in the United States and other countries. All other brand names, product names or trademarks belong to their respective owners. © 2023 Splunk Inc. All rights reserved.