Adversary tradecraft 101: Hunting for persistence using Elastic Security (Part 1)

By David French, Brent Murphy, Elastic Security Intelligence & Analytics Team

Published: 2022-06-01 · Archived: 2026-04-05 15:25:52 UTC

Last month, we hosted a webinar, Hunting for persistence using Elastic Security, where we examined some techniques that attackers use in the wild to maintain presence in their victim's environment. For each technique covered, we explained:

- How the offensive technique works
- Why the technique is often successful for attackers
- How defenders can hunt for and detect the malicious behavior effectively using Elastic Security

In this two-part blog series, we'll share the details of what was covered during our webinar with the goal of helping security practitioners improve their visibility of these offensive persistence techniques and help to undermine the efficacy of these attacks against their organization.

Part 1 will explain what persistence is and why attackers need it. We'll introduce the Event Query Language (EQL) before showing its practical use cases for threat hunting. We will examine a popular technique used by adversaries to maintain persistence, Windows Management Instrumentation (WMI) Event Subscription (T1084). We'll also share how Elastic Security users can hunt for and detect this technique being used in their environment.

In part 2, we'll explore two additional persistence techniques that are being used by attackers in the wild: BITS Jobs (T1197) and Scheduled Tasks (T1053). This follow-up post will walk through real world examples of these techniques being used and how we can hunt for, detect, and prevent them using Elastic Security.

The Protections team at Elastic Security is responsible for researching, understanding, and developing detections and preventions for attacker behavior in order to stop attacks before damage or loss occur. For organizations that do not have a full-time security operations team, Elastic Security includes out-of-the-box protections against adversary tradecraft, malware, and attacks like ransomware and phishing.

What is persistence and why do attackers need it?

When we consider the common components of an intrusion, such as those depicted in Figure 1, the attacker may have spent a considerable amount of effort carrying out reconnaissance, obtaining initial access to, and establishing a foothold in their target environment. Generally speaking, an adversary will often want to maintain a presence in order to survive disruptions to their access like system restarts or user password changes. An effective persistence mechanism will execute the attacker's malicious code on a regular basis or when a specific condition is met such as a user logon or application launch event.

In many cases, gaining access to an organization's network is harder than maintaining persistence. It is for this reason that attackers continue to use the persistence techniques covered in this post — with largely successful results. When the security industry refers to "dwell time," that's the period of time that begins when an adversary gains access to a system and ends when you detect them. Persistence facilitates longer dwell times, during which the adversary can work to achieve their objectives.

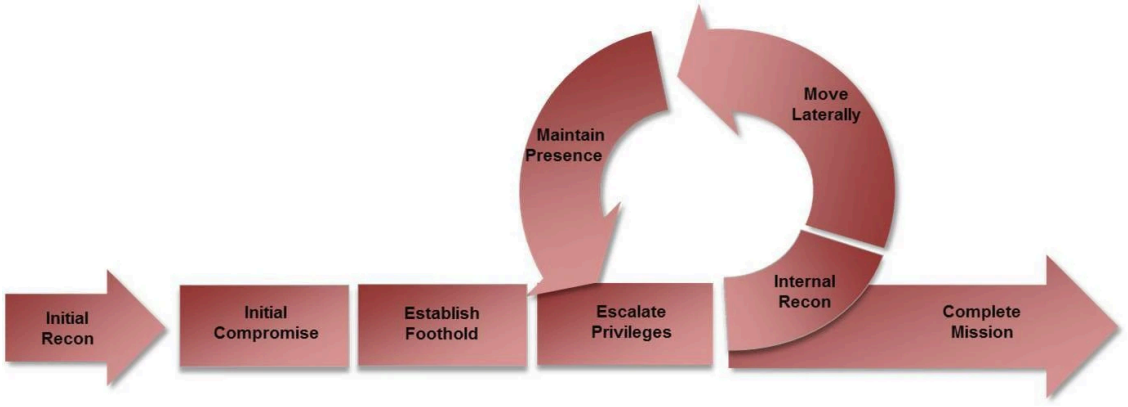

Figure 1 - FireEye Mandiant's Cyber Attack Lifecycle

There are currently 63 persistence techniques documented in the MITRE ATT&CK® knowledge base of adversary behavior. This number can seem daunting at first, but an adversary's need for persistence can be their Achilles' heel, providing defenders with valuable opportunities to detect and remove an attacker from their environment.

By learning these offensive tactics, techniques, and procedures (TTPs) and baselining the endpoints and network activity in your environment, you have an opportunity to detect attackers early on in an intrusion before any damage or loss occurs. Security operations teams work tirelessly to detect the techniques captured in the ATT&CK matrix — hunting and writing alert logic to provide the greatest visibility and coverage of their enterprise. To help reduce the barrier to entry and enable those teams, we created EQL and released it to the community.

What is Event Query Language (EQL)?

The queries in this blog post, which can be used for threat hunting and detection, are written in EQL — a powerful and intuitive query language that makes it easy for practitioners to search for complex adversary behavior.

To learn more about EQL, you can read the Getting started with EQL blog post or review the EQL Analytics Library, which contains 200+ free security analytics mapped to the techniques documented in the MITRE ATT&CK matrix. EQL is a core component of Elastic Endpoint Security and is being added to Elasticsearch for use in Elastic SIEM.

Let's now analyze the WMI Event Subscription technique used by attackers in the wild by demonstrating a variety of methods to hunt for and detect this technique. We decided to cover this technique given its popularity among attackers and the low detection rates by traditional security controls.

Persistence via Windows Management Instrumentation (WMI) Event Subscriptions (T1084)

WMI is the Microsoft implementation of Web-Based Enterprise Management (WBEM), a collection of technologies used to manage information and systems in enterprise environments. WMI is a built-in feature of Windows endpoints that allows both administrators and attackers to interact and manage many different functions of the operating system. Components such as network interfaces, power settings, service status, and installed software or patches can all be queried and managed via WMI.

Attackers have been abusing WMI since as early as 2008 to accomplish different objectives such as moving laterally between endpoints, enumerating what processes are running on an endpoint, and to maintain persistence. An attacker can “live off the land” by abusing built-in features of the operating system, which often results in a lower risk of detection than if they introduced malware into a victim environment.

It's no secret that adversaries prefer WMI-based persistence because:

- Many organizations are incapable of monitoring or investigating WMI
- Out-of-the-box WMI visibility is limited, and a third-party utility may be required to effectively convey how WMI is being used
- WMI is an integral component of Windows, and the volume of weak signals from this data source can overwhelm analysts who aren't familiar with these attack types

Like most offensive techniques, until organizations can reliably detect WMI abuse, the adversary is unlikely to change their behavior.

Understanding WMI Event Subscriptions and how they can be abused

Simply put, a WMI Event Subscription can trigger an action when a certain condition is met. A WMI Event Subscription consists of three components.

- EventFilter - specifies a condition that you test for, i.e. a user successfully authenticates, a particular time and day of the week occurs, etc.
- EventConsumer - specifies an action to execute when the EventFilter condition is met, i.e. execute a Windows command or script, delete all Volume Shadow Copies, restart a service, etc.
- FilterToConsumerBinding - this links an EventFilter to an EventConsumer instance

Figure 2 below shows the output from Sysinternals Autoruns, a free tool from Microsoft that can be used to examine various persistence locations on Windows endpoints, also referred to as Autostart Execution Points (ASEPs). Autoruns reveals a persistent WMI Event Subscription, named checkforupdates. It is important to note that there are sometimes benign WMI Event Subscriptions configured, but there shouldn’t be many. Alerting on the creation of new WMI Event Subscriptions and hunting for new ones periodically can be a low cost, high reward exercise.

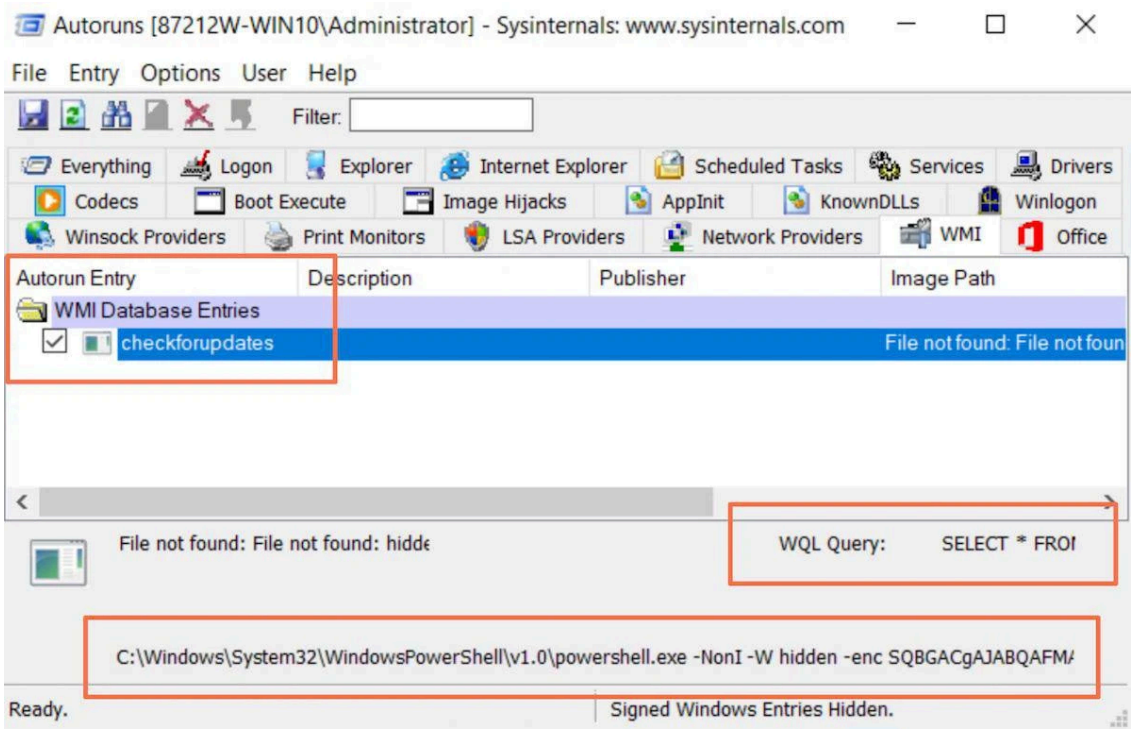

Figure 2 - Autoruns showing a WMI Event Subscription registered by Empire

The full WMI Query Language (WQL) query from the Autoruns results is shown below. Autoruns conveniently enumerated the WMI __EventFilter class for us in the root\Subscription namespace where this malicious entry was created. Interpreting this WQL query, the EventFilter condition will be met when the system's uptime is between 240 -325 seconds. The EventConsumer is an Empire PowerShell script and will execute when this EventFilter condition is met.

To summarize, a WMI Event Subscription has been configured to execute a malicious PowerShell script shortly after the endpoint boots up. This enables the attacker to maintain persistence in the victim's environment and survive system restarts.

```
 SELECT * FROM __InstanceModificationEvent WITHIN 60 WHERE TargetInstance ISA 'Win32_PerfFormattedData_PerfOS_Sy
TargetInstance.SystemUpTime >= 240 AND
TargetInstance.SystemUpTime < 325
```

Figure 3 - WMI Query Language (WQL) query showing WMI EventFilter condition

Hunting for and detecting malicious WMI Event Subscriptions

The EQL query in Figure 4 shows how we can search for a sequence of three WMI events, which were generated by the same unique process ID (PID). This query demonstrates one of the strengths of EQL by using the join function. This allows us to match sequences of events in any order. Searching for these three WMI events without binding them together by unique PID might not return meaningful results. This query matches when a WMI EventFilter, EventConsumer, and FilterToConsumer binding are created by the same process in succession, which typically occurs when malware is executed and creates a new WMI Event Subscription for persistence. This query can be used to monitor for and detect abuse of WMI.

```
join by unique_pid
 [wmi where properties.Operation == "*IWbemServices::PutInstance*EventFilter*"]
 [wmi where properties.Operation == "*IWbemServices::PutInstance*EventConsumer*"]
 [wmi where properties.Operation == "*IWbemServices::PutInstance*FilterToConsumerBinding*"]
```

Figure 4 - EQL query to search for the creation of a WMI EventFilter, EventConsumer, and FilterToConsumerBinding in succession

The above EQL query can be saved as a custom rule in Elastic Endpoint Security so that analysts can be alerted every time a new WMI Event Subscription is created by a process (Figure 5).

ALERT TYPE	EVENT TYPE	ASSIGNEE	OS	IP ADDRESS
• Persistence Detection	WMI FilterToConsumer Binding Creation	Unassigned	Windows 10 (v1809)	172.16.66.9
Execution Detection	PowerShell with Unusual Arguments	Unassigned	Windows 10 (v1809)	172.16.66.9

Figure 5 - Alert created by custom rule in Elastic Endpoint Security

Resolver is the primary interface for interacting with alerts in Elastic Endpoint Security and enables users to visualize process ancestry along with relevant events such as DNS, file, network, WMI, and PowerShell, to name a few. It also provides one-click response actions to resolve the alert.

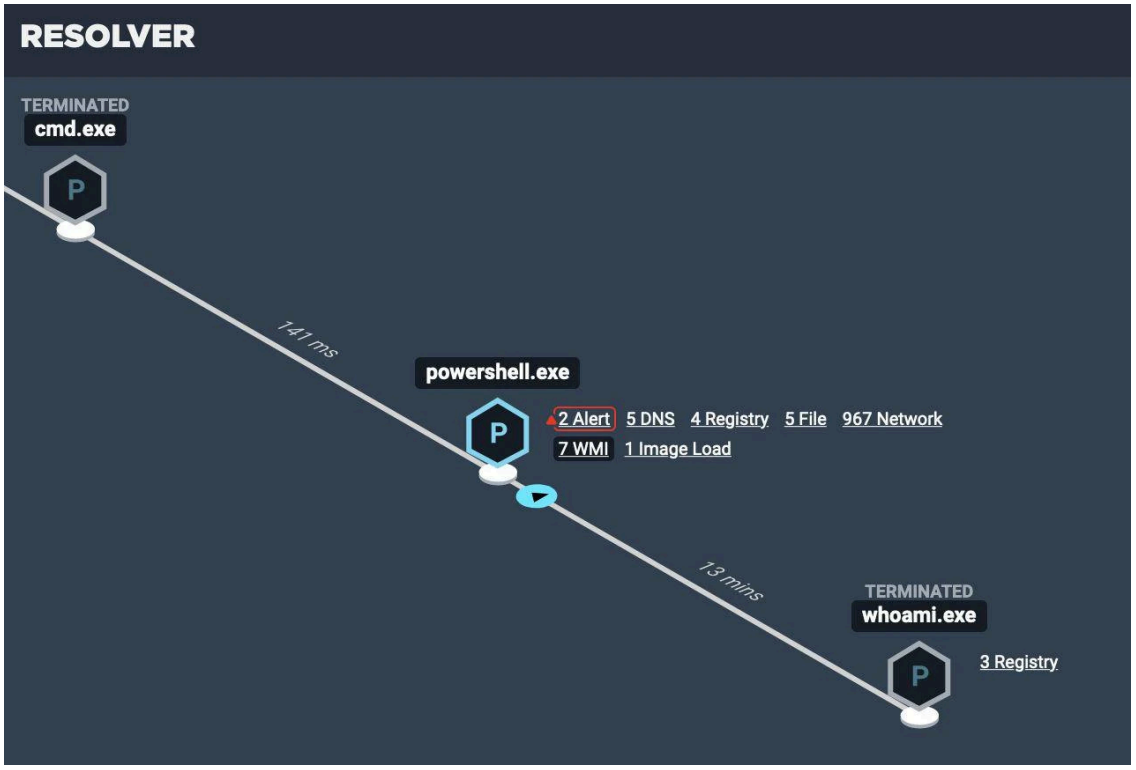

Figure 6 - Resolver showing process ancestry and events created by powershell.exe

Clicking the WMI event type next to powershell.exe in Resolver enables users to review the WMI events that were generated by the process. Figure 7 shows the WMI event that was logged when powershell.exe was used to create the new WMI FilterToConsumerBinding, checkforupdates.

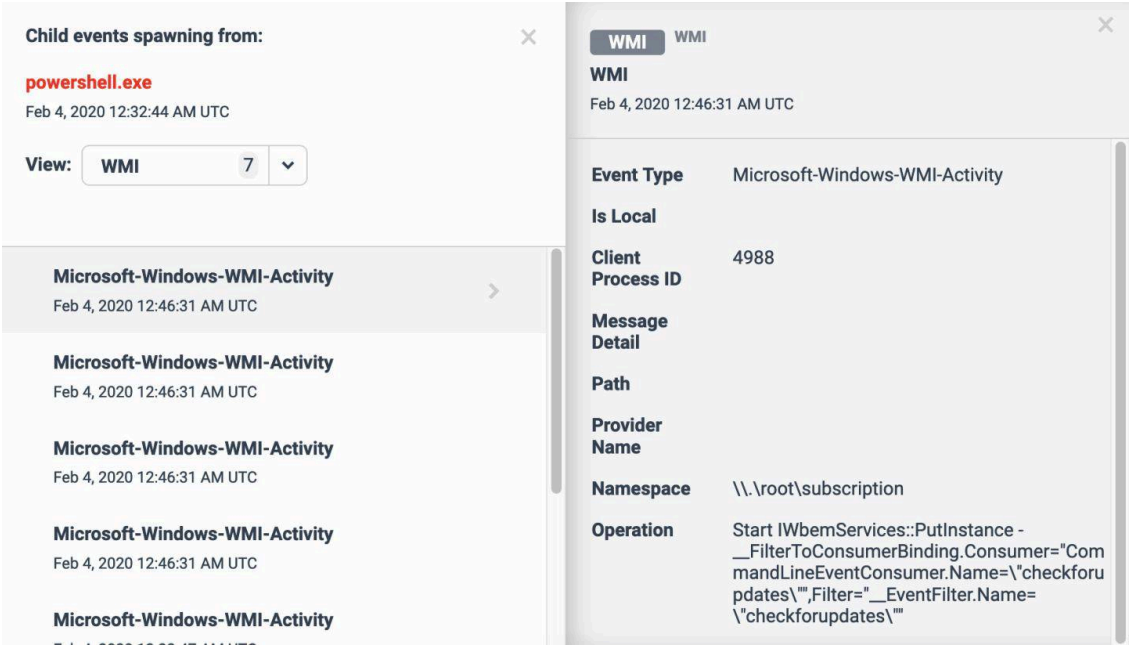

Figure 7 - WMI event created by powershell.exe

Elastic Endpoint Security also includes template-based hunts that lower the barriers to entry for less experienced analysts and allows for the easy collection and analysis of data across an organization's endpoints. There are currently 25 categories of persistence data that can be collected and analyzed using template-based hunts.

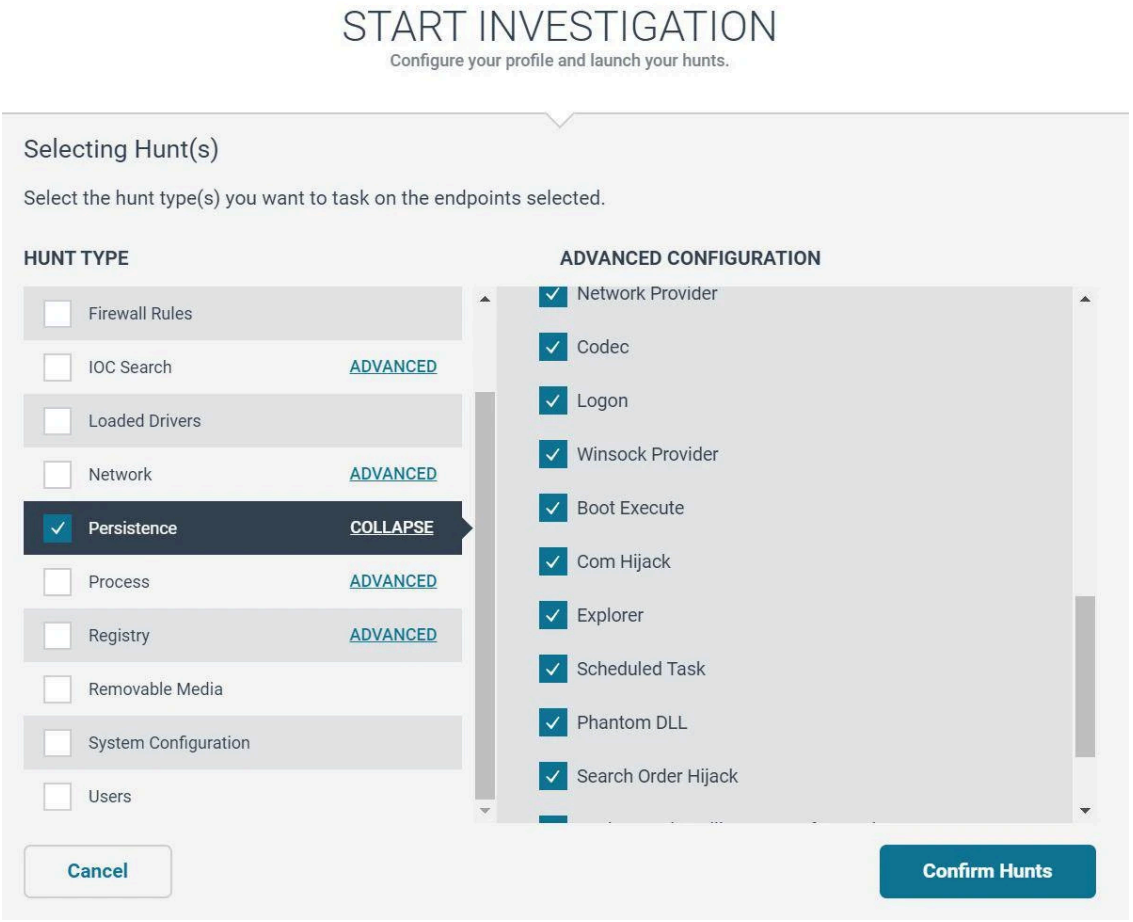

Figure 8 - Hunting for persistence using Elastic Endpoint's template-based hunts

Once a hunt has been executed, there are several ways to sort or aggregate the data — such as by frequency or file path. This helps analysts to quickly baseline their endpoints and identify anomalies to investigate further.

Figure 9 depicts the filtered results of a persistence hunt wherein the same malicious WMI Event Subscription was configured. Elastic Security enumerates persistence locations across your endpoints and enables users to apply filters to identify anomalies or suspicious artifacts for further analysis. Hunt teams will often assume that traditional passive and reactive security controls are fallible, and will proactively hunt for malicious persistence mechanisms in search for attackers operating in their environment.

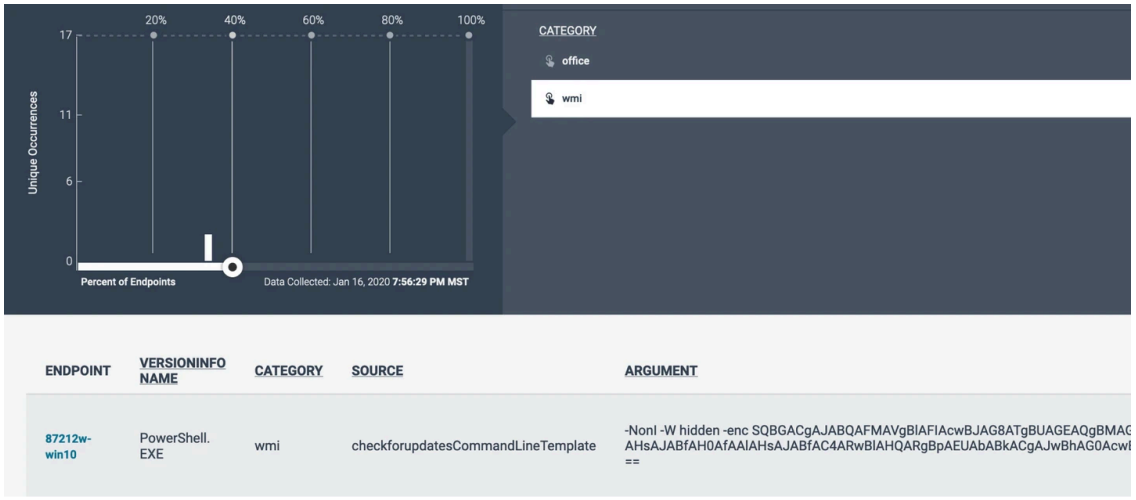

Figure 9 - Results of an Elastic Endpoint hunt showing malicious WMI persistence

Elastic Security comes with out-of-the-box detections for WMI abuse. Figure 10 shows the signal that was generated by Elastic Endpoint and shipped to Elastic SIEM when a malicious WMI Event Subscription was created.

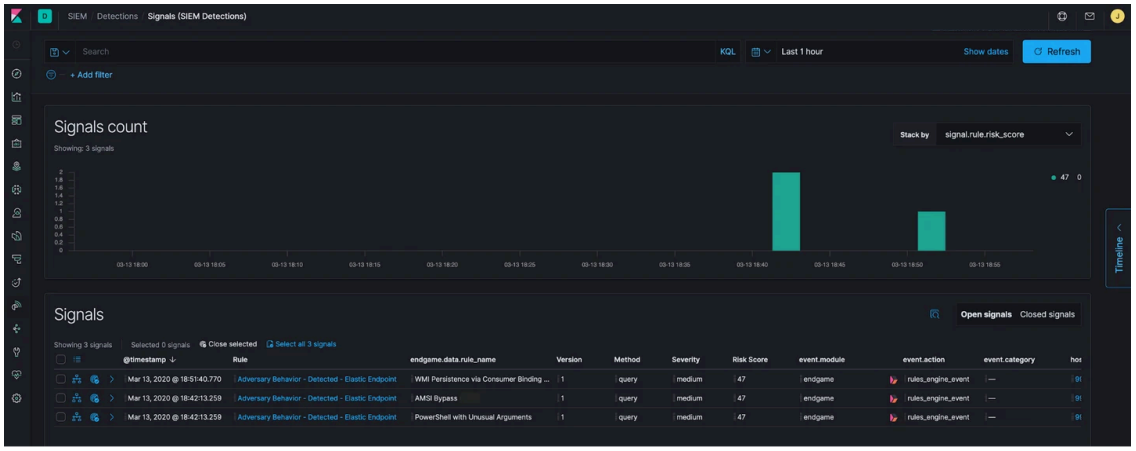

Figure 10 - Elastic Endpoint signal shown in Elastic SIEM

Users have the option to view a signal in the Timeline within Elastic SIEM. Timeline enables analysts to search for similar activity across their data, gather and document evidence, and forward potential incidents to ticketing and SOAR platforms with ease. Part 2 of this series will show the workflow and features of Timeline.

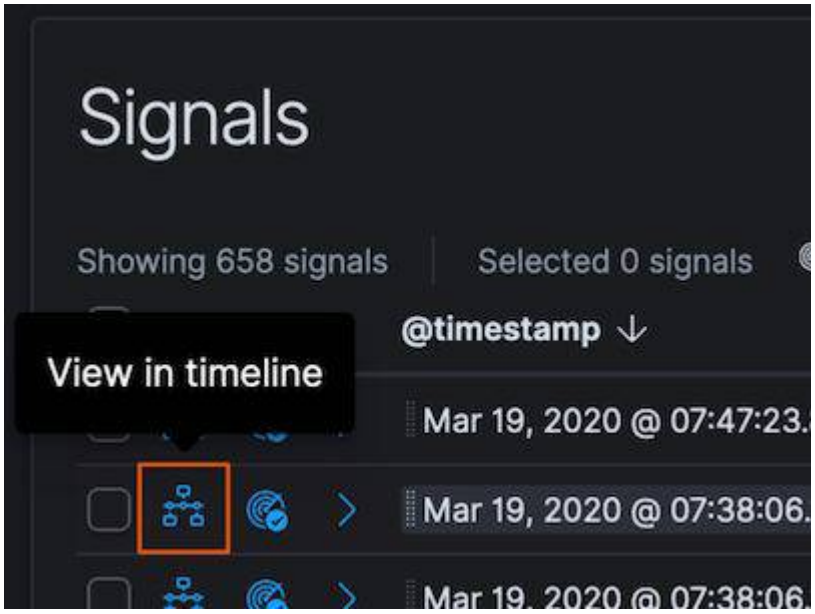

Figure 11 - Option to view signal in Timeline in Elastic SIEM

As of version 7.6 of the Elastic Stack, the SIEM app comes with a detection engine, which enables security teams to create their own custom rules. For example, Windows 10 logs Event ID 5861 when a new WMI EventFilterToConsumer binding is created. Figure 12 shows how a custom rule can be created in Elastic SIEM to search the winlogbeat-* index pattern for winlog.record_id: 5861. We can configure a description, severity, and risk score for the new rule, as well as map the rule to the relevant techniques in the MITRE ATT&CK matrix. This information will help an analyst triage and determine the steps to take when the rule condition occurs.

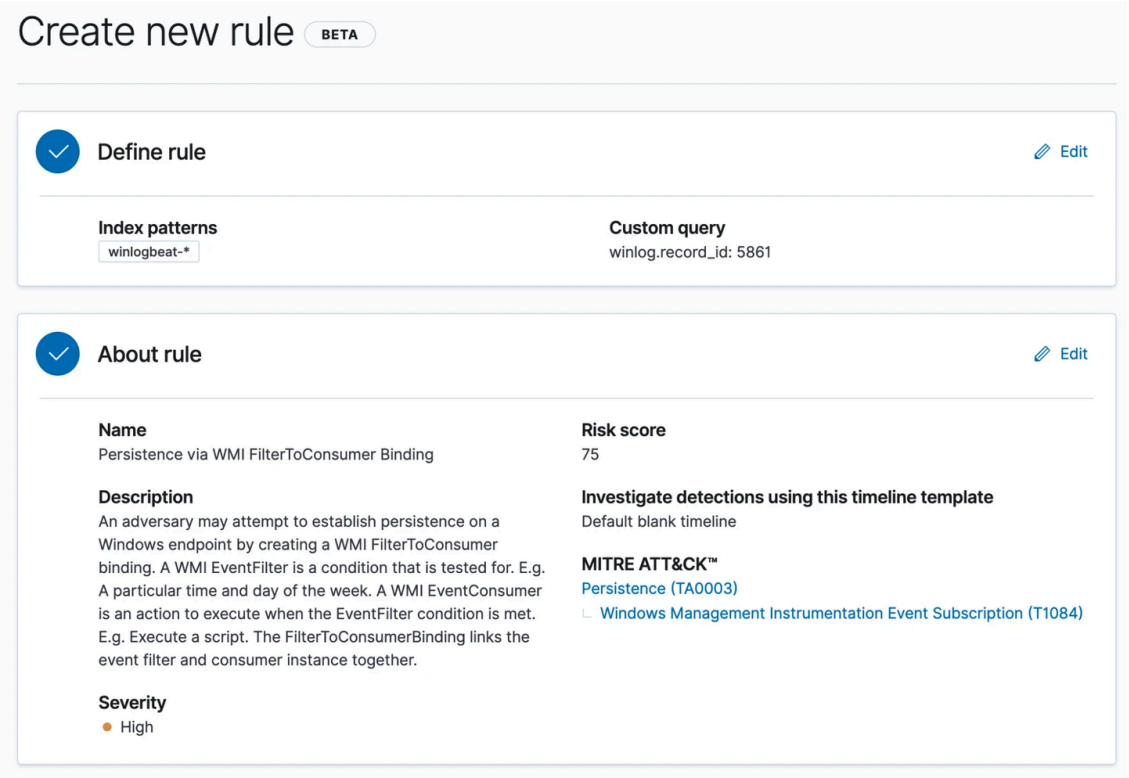

Figure 12 - Example of a new rule being created in Elastic SIEM

Elastic SIEM is part of the Basic subscription and currently includes 92 detections that utilize Windows, Linux, network, and APM logging. We will continue to add new rules in new releases of the Elastic Stack.

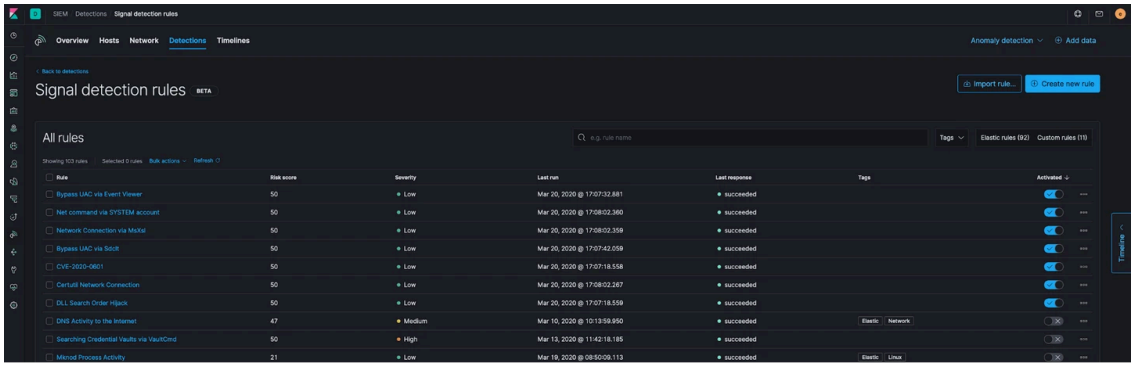

Figure 13 - Detection rules view in Elastic SIEM

In this section, we learned how WMI Event Subscriptions can be abused for persistence and how we can hunt for and detect this technique using the combination of Elastic Endpoint and SIEM. It's important to note that we have only scratched the surface of how WMI works and the many ways that it can be used by attackers.

WMI can be used during every phase of an attack, such as moving laterally between endpoints, conducting local and enterprise reconnaissance, and stealing data. Elastic Security has released detections for many techniques that leverage WMI.

Conclusion

In this blog post, we examined a popular technique that attackers use to maintain a presence in their target environments. The number of techniques in an attacker's arsenal can seem daunting at first, but we demonstrated a formulaic approach to examining, hunting for, and detecting techniques effectively. By building comprehension around adversary tradecraft, you can identify interesting patterns, behaviors, and artifacts that you can use to your advantage.

Elastic Security makes hunting for persistence easy. The features of Elastic Endpoint Security and SIEM (along with the protections provided out of the box) lower the barriers to entry for analysts, provide detailed visibility into endpoint activity, and enable organizations to prevent, detect, and respond to malicious behavior at scale.

To learn more about threat hunting, download a free copy of The Elastic Guide to Threat Hunting.

Plus, EQL support is being added to Elasticsearch!
