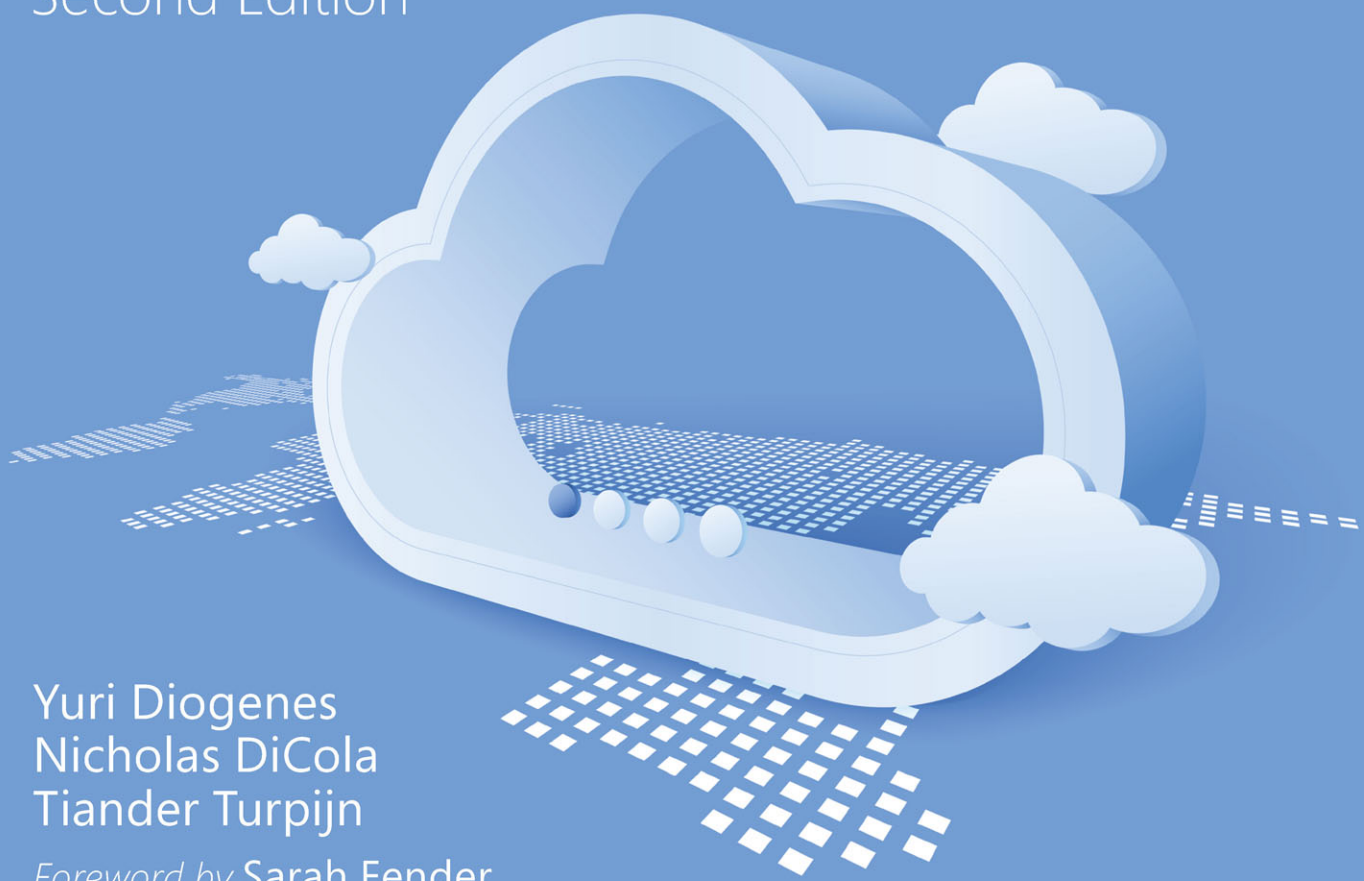
Microsoft

Microsoft Sentinel

Planning and implementing Microsoft's cloud-native SIEM solution

Second Edition



Yuri Diogenes
Nicholas DiCola
Tiander Turpijn

Foreword by Sarah Fender
Partner Director of Product Management – Microsoft Sentinel

Microsoft

Microsoft Sentinel

Planning and implementing Microsoft's cloud-native SIEM solution

Second Edition

Yuri Diogenes
Nicholas DiCola
Tiander Turpijn

Microsoft Sentinel: Planning and implementing Microsoft's cloud-native SIEM solution, Second Edition

Published with the authorization of Microsoft Corporation by:
Pearson Education, Inc.

ISBN-13: 978-0-13-790093-0
ISBN-10: 0-13-790093-7

Library of Congress Control Number: 2022942055

ScoutAutomatedPrintCode

EDITOR-IN-CHIEF
Brett Bartow

EXECUTIVE EDITOR
Loretta Yates

SPONSORING EDITOR
Charvi Arora

DEVELOPMENT EDITOR
Rick Kughen

MANAGING EDITOR
Sandra Schroeder

SENIOR PROJECT EDITOR
Tracey Croom

COPY EDITOR
Rick Kughen

INDEXER
Valerie Haynes Perry

PRODUCTION EDITOR
Dan Foster

PROOFREADER
Dan Foster

TECHNICAL EDITOR
Javier Soriano

EDITORIAL ASSISTANT
Cindy Teeters

COVER DESIGNER
Twist Creative, Seattle

COMPOSITOR
Danielle Foster

GRAPHICS
Vived Graphics

Pearson's Commitment to Diversity, Equity, and Inclusion

Pearson is dedicated to creating bias-free content that reflects the diversity of all learners. We embrace the many dimensions of diversity, including but not limited to race, ethnicity, gender, socioeconomic status, ability, age, sexual orientation, and religious or political beliefs.

Education is a powerful force for equity and change in our world. It has the potential to deliver opportunities that improve lives and enable economic mobility. As we work with authors to create content for every product and service, we acknowledge our responsibility to demonstrate inclusivity and incorporate diverse scholarship so that everyone can achieve their potential through learning. As the world's leading learning company, we have a duty to help drive change and live up to our purpose to help more people create a better life for themselves and to create a better world.

Our ambition is to purposefully contribute to a world where:

- Everyone has an equitable and lifelong opportunity to succeed through learning.
- Our educational products and services are inclusive and represent the rich diversity of learners.
- Our educational content accurately reflects the histories and experiences of the learners we serve.
- Our educational content prompts deeper discussions with learners and motivates them to expand their own learning (and worldview).

While we work hard to present unbiased content, we want to hear from you about any concerns or needs with this Pearson product so that we can investigate and address them.

Please contact us with concerns about any potential bias at *https://www.pearson.com/report-bias.html*.

Figure Credits

FIG01-01A: mei yanotai/Shutterstock

FIG01-01B, FIG01-01C, FIG01-01E, FIG08-11J: boreala/Shutterstock

FIG01-01D: Kar/Shutterstock

FIG01-01F: Vladislav Markin/123RF

FIG01-01G, FIG01-01J, FIG08-11E: edel/Shutterstock

FIG01-01H: mei yanotai/Shutterstock

FIG01-01I: Oxford Designers & Illustrators Ltd./Pearson Education Ltd

FIG01-01K: Dolvalol/Shutterstock

FIG06-22 - FIG06-24: Virustotal

FIG06-25 - FIG06-26: geoiplookup.net

FIG08-11A, FIG08-11C, FIG08-11D, FIG08-11G, FIG08-11H, FIG08-11I, FIG08-11K, FIG08-11L: VectorForever/Shutterstock

FIG08-11B: bspsupanut/123RF

FIG08-11F: popcic/Shutterstock

Contents at a Glance

Contents

Foreword

Microsoft Sentinel, formerly Azure Sentinel, was introduced in 2019 to help organizations modernize security operations in the cloud. At that time, security operations teams—who were under increasing pressure to extend coverage across a growing digital estate, combat escalating threats, and improve efficiency—were beginning to look to the cloud for alternatives to expensive and underperforming on-premises systems. Since then, tens of thousands of customers have adopted a cloud-first approach to power their data and compute-intensive security operations workloads, with Microsoft Sentinel becoming the solution of choice because of its cloud-native architecture and industry leading intelligence and analytics capabilities. Today, some of the world's largest Security Operations Centers (SOCs) run on Microsoft Sentinel, including Microsoft's own SOC. As the hub for security operations, Microsoft Sentinel brings together data, analytics, and workflows to unify and accelerate threat detection and response across the customer's entire digital estate. Microsoft Sentinel provides an extensible solution to power all facets of security operations (threat intelligence and hunting, detection and correlation, incident management, investigation, and remediation) and operate across all data sources.

In this second edition of *Microsoft Sentinel: Planning and implementing Microsoft's cloud-native SIEM solution*, you will have the opportunity to learn from an expert team of cybersecurity experts and engineers who have helped countless customers and partners successfully transform their security operations. They will lay out the foundational aspects of architecting, implementing, and operationalizing Microsoft Sentinel for customers, large and small. Topics include data collection and archiving, threat hunting and detection, incident response and automation, threat intelligence, and more, with practical advice gained from real-world experience.

With the dynamic nature of the security landscape and rapid pace of innovation, this book provides the latest insights you need to realize the full potential of Microsoft Sentinel to help your SOC team achieve more.

Sarah Fender
Partner Director of Product Management
Microsoft Sentinel

Acknowledgments

The authors would like to thank Loretta Yates and the entire Microsoft Press/Pearson team for their support in this project, Sarah Fender for writing the foreword, and also the Microsoft Sentinel Engineering Team. We would also like to thank Javier Soriano for reviewing this book and writing Appendix B, and Mike Kassis for writing Appendix A.

Yuri would also like to thank: my wife and daughters for their endless support; my great God for giving me strength and guiding my path, each step of the way; my co-authors and friends Nicholas DiCola and Tiander Turpijn for such great partnership throughout this project. Thanks to my parents for working hard to give me an education, which is the foundation I use every day to keep moving forward in my career.

Nicholas would also like to thank: my wife and three children for supporting me while working on this book, and my co-authors and friends Yuri and Tiander for their hard work on this book. I would also like to thank our Microsoft Sentinel Engineering team technical reviewers for their support of the book.

Tiander would also like to thank: my love Miriam Smit for believing in me, your endless support for all I do, regardless of whether it makes sense. My dear friends Jimmy Eekhout, Ronald Beijnon, and Arthur de Meij for being there when I need them. I'd like to thank my dear mother, Sonja, who gave everything so that I could study and get a proper education. I miss you every day. Thank you to my co-authors and friends, Yuri Diogenes (thank you for the writing opportunity) and Nicholas DiCola, for your great partnership while writing this book and beyond.

About the authors

Yuri Diogenes, MsC

Yuri holds a Master of Science in cybersecurity intelligence and forensics investigation from UTICA College and is currently working on his Ph.D. in cybersecurity leadership from Capitol Technology University. Yuri has been working at Microsoft since 2006 and currently is a principal program manager for the CxE Microsoft Defender for Cloud Team. Yuri has published a total of 26 books, mostly about information security and Microsoft technologies. Yuri is also a professor at EC-Council University, where he teaches in the Bachelor of Cybersecurity Program. Yuri is an MBA and holds many IT/Security industry certifications, such as CISSP, MITRE ATT&CK® Cyber Threat Intelligence Certified, E|CND, E|CEH, E|CSA, E|CHFI, CompTIA Security+, CySA+, Network+, CASP, and CyberSec First Responder. You can follow Yuri on Twitter at @yuridiogenes.

Nicholas DiCola

Nicholas is the Vice President of Customers at Zero Networks, where he leads the customer engineering team that helps customers with pilots and deployments of Zero Networks products. He has a Master of Business Administration with a concentration in information systems. He holds various industry certifications, such as CISSP and CEH. You can follow Nicholas on Twitter at @mastersecjedi.

Tiander Turpijn

Tiander is a principal program manager for Microsoft Sentinel. He joined Microsoft in 1998 and fulfilled multiple roles, from senior escalation support engineer, senior management & security consultant, and architect to a datacenter architecture role. Tiander has a computer science degree and various industry certifications, such as CISSP and CEH. You can follow Tiander on Twitter at @tianderturpijn.

Introduction

Welcome to *Microsoft Sentinel*. This book was developed with the Microsoft Sentinel product group to provide in-depth information about Microsoft's new cloud-based security information and event management (SIEM) system, Microsoft Sentinel, and to demonstrate best practices based on real-life experience with the product in different environments.

The purpose of this book is to introduce the wide array of capabilities available in Microsoft Sentinel. After being introduced to the primary-use case scenarios, you will learn how to deploy and operationalize Microsoft Sentinel for data collection, analytics, incident management, threat detection, and response.

Who is this book for?

Microsoft Sentinel is for anyone interested in security operations in general: cybersecurity analysts, security administrators, threat hunters, support professionals, and engineers.

Microsoft Sentinel is designed to be useful for Azure and non-Azure users. You can have no security experience, some experience, or be a security expert, and you will get value from Microsoft Sentinel. This book provides introductory, intermediate, and advanced coverage of a large swath of security issues that Microsoft Sentinel addresses.

The approach is a unique mix of didactic, narrative, and experiential instruction. The didactic approach covers the core introductions to the services. The narrative instruction leverages what you already understand. We bridge your current understanding with new concepts introduced in the book. Finally, the experiential component is presented in two ways. First, we share our experiences with Microsoft Sentinel, and second, we show you how to get the most out of Sentinel by explaining it in a stepwise, guided fashion. We show you how to configure Microsoft Sentinel to gain all the benefits it has to offer.

In this book, you will learn:

- How to connect different data sources to Microsoft Sentinel
- How to create security analytics
- How to investigate a security incident in Microsoft Sentinel

System requirements

Anyone with access to a Microsoft Azure subscription can use the information in this book.

Errata, updates & book support

We've made every effort to ensure the accuracy of this book and its companion content. You can access updates to this book—in the form of a list of submitted errata and their related corrections—at:

MicrosoftPressStore.com/MicrosoftSentinel/errata

If you discover an error that is not already listed, please submit it to us at the same page.

For additional book support and information, please visit *MicrosoftPressStore.com/Support*.

Please note that product support for Microsoft software and hardware is not offered through the previous addresses. For help with Microsoft software or hardware, go to *http://support.microsoft.com*.

Stay in touch

Let's keep the conversation going! We're on Twitter: *http://twitter.com/MicrosoftPress*.

Chapter 1

Security challenges for SecOps

Microsoft Sentinel is a cloud-native Security Incident and Event Management (SIEM) solution built to provide security analysts with a powerful tool to detect and respond to cyberattacks. Before diving into the purpose and details of the solution, it is important to understand the key challenges facing Chief Information Security Officers (CISOs) and their teams. Today's security teams face myriad challenges, including the speed and sophistication of current threats, exponential growth in the number of digital assets and associated logs, and the lack of available and skilled staff.

In this chapter, we discuss the current challenges facing cyberdefenders, starting with a review of the current threat landscape. One concerning trend is that attackers are now targeting key software supply chains to circumvent traditional security controls. The speed of attacks is always increasing, making traditional and manual response procedures ineffective.

Also, in this chapter, we review the importance and use of threat intelligence in a modern Security Operations Center (SOC). Threat intelligence provides defenders with the details of an attacker's motivations, potential targets, and tactics, techniques, and procedures (TTPs). TTPs can be used by security analysts to build custom detections to alert on attacker activities as they occur, or TTPs can be leveraged to hunt through data for previous indicators of an attack. Finally, we conclude the chapter by providing a high-level overview of Microsoft Sentinel capabilities.

Current threat landscape

The current state of cybercrime shows that amateur threat actors with low technical level skills are investing in Ransomware as a Service (RaaS) for their campaigns because the ransomware kits provided by these professional cybercriminals are very sophisticated and easy to use. According to Microsoft Digital Defense Report 2021, the payment for these ransomware kits can be based on a percentage of the profit, such as 30 percent of the ransom. This model encourages amateur threat actors to take the risk because there will be zero upfront investment.

In 2021, ransomware gained a lot of visibility, mainly after the Colonial Pipeline—one of the largest oil pipelines in the United States—was attacked with ransomware. While the news emphasized the ransomware attack, it's important to understand that the threat actor first had to establish a foothold in the network, which was done by exploiting a legacy VPN. In 2021, threat actors often targeted the VPN infrastructure by exploiting known vulnerabilities in Pulse Secure VPN appliances.

However, it is not only about RaaS; professional cybercriminals also have different online offerings, such as counter-antivirus (CAV) services, which scan antivirus engines to ensure new malware can be successfully deployed without detection. Professional cybercriminals also can take advantage of bulletproof hosting services for online criminal activity. (They're called "bulletproof" because the owners of these servers do not cooperate with law enforcement investigations.) There are even escrow services that act as a third party in online transactions between technical criminals and their criminal clients.

TIP **To download the Microsoft Digital Defense Report 2021, see *www.microsoft.com/digitaldefensereport*.**

In 2021, we also saw Acer getting hit by REvil ransomware (a Russian-based RaaS), where the threat actors demanded $50 million, which is the largest known ransom to date. JBS Foods was also attacked by REvil, causing the temporary closure of operations in Canada, Australia, and the United States. JBS Foods ended up paying $11 million in ransom (see *https://www.cbsnews.com/news/jbs-ransom-11-million*), which is one of the biggest ransomware payments of all time.

TIP **You can see a list of all known techniques used by REvil at *https://attack.mitre.org/software/S0496*.**

Cybercriminals may also use advanced code injection methods, such as fileless techniques. This attack usually leverages tools that are already in the target system, such as PowerShell. By leveraging a tool that is already on the computer, cybercriminals don't need to write to the hard drive. Instead, they only need to take over the target process, run a piece of code in its memory space, and then use that code to call the tool that will be used to perform the attack.

One year earlier, a nation-state cyberattack impacted high-value targets across both the government and private sectors. This supply chain attack was known as Solorigate or Sunburs. SolarWinds Orion Platform DLL was the main component that led to this sophisticated attack. The subtle malicious codes inserted into the DLL were able to call a backdoor composed of almost 4,000 lines of code. This allowed the threat actor behind the attack to operate unrestricted in the target networks. Figure 1-1 summarizes how this attack worked.

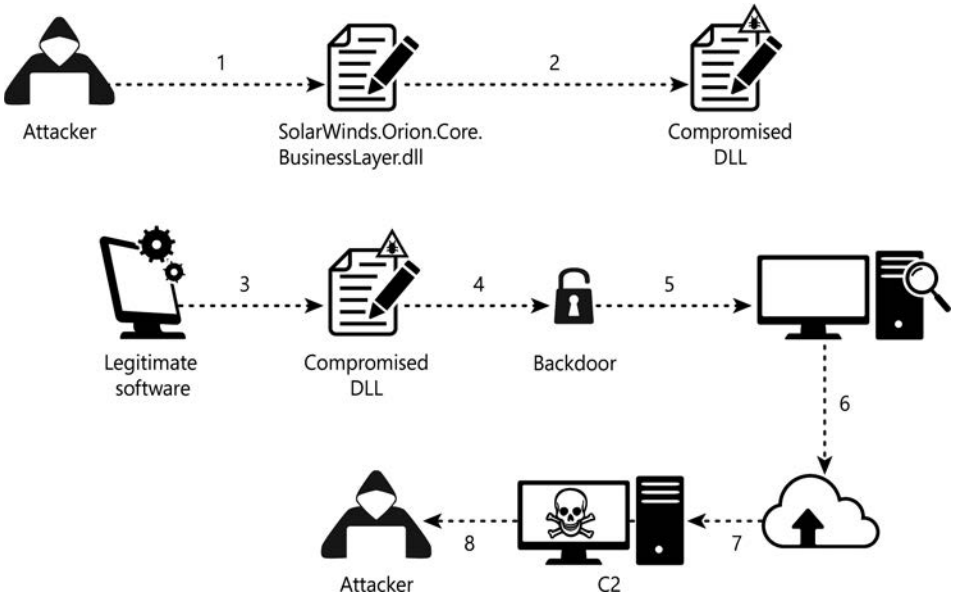


FIGURE 1-1 Stages of the Solorigate attack

The steps described in Figure 1-1 are shown below:

1. The attacker inserts malicious code into the DLL.
2. The compromised DLL is distributed to organizations that use this software.
3. When the legitimate software starts, the compromised DLL is loaded.
4. The malicious code that is part of this compromised DLL calls the function that has the backdoor capability.
5. The backdoor has a laundry list of things that it needs to check to ensure that it is operating in the compromised environment, so it performs many activities as part of this validation.
6. After finishing inspecting the environment, the backdoor will gather the necessary information.
7. After gathering information, the backdoor will make a call to the command-and-control (C2). During this call, the backdoor might receive a list of other C2s to connect with.
8. The backdoor sends the information to the attacker, which at this point is more of a hands-on-keyboard type of attack. The backdoor runs commands received from the attacker to perform additional activities, such as compromising credentials, privilege escalation, and lateral movement.

One advantage of looking at these different phases of the attack is to understand where each phase maps to the MITRE ATT&CK knowledge base (*attack.mitre.org*). This is important for the SecOps team because by understanding the different phases, you can better strengthen your security controls to catch the attack early on and prevent the completion of the attack. Microsoft Sentinel utilizes MITRE ATT&CK throughout many areas of the product to help the

SOC Team prioritize and triage incidents. Microsoft Sentinel has also included an entire blade dedicated to MITRE ATT&CK, as shown in Figure 1-2.

FIGURE 1-2 MITRE dashboard in Microsoft Sentinel

If we go back to step 6, when the information is being gathered, we can map it to the Discovery Tactic of the MITRE ATT&CK (*https://attack.mitre.org/tactics/TA0007*), which is when the adversary is trying to figure out the environment. By understanding the tactics and techniques used by an adversary, you can improve your defenses. For example, Sunburst used Base64 encoding during step 7 to communicate with C2. This subtechnique is mapped to MITRE ATT&CK Data Encoding: Standard Encoding (*https://attack.mitre.org/techniques/T1132/001*). There, you can find proper mitigations to prevent this attack and suggestions for detection, which you can use to incorporate into Microsoft Sentinel.

It is also important to emphasize that when dealing with an attack of this nature, you need to understand the TTPs that are used in this attack. If you know that your environment has SolarWinds Orion, you should start building an inventory of the machines with this component installed. You could use Microsoft Sentinel to run a simple query to gather similar details. In a fully functional environment where Microsoft Sentinel is actively collecting data from multiple locations, the SOC Team can create a simple query to pull the hosts with the SolarWinds process running in the last 30 days based on the process execution.

> ***NOTE*** **The Microsoft Sentinel Team maintains a GitHub repository with sample queries that you can use for this purpose. See *http://aka.ms/MSSentinelQueries*.**

In November 2021, the world was surprised by the CVE-2021-44228 vulnerability, which was related to Log4Shell. This vulnerability affected Log4j, which is an open-source logging framework in Java that is widely used by developers in cloud and enterprise applications. At that point, threat actors started working to exploit this vulnerability. The exploitation could be done by creating a specially crafted Java Naming and Directory Interface (JNDI) command. That command is then sent to a vulnerable server (hosting a vulnerable version of log4j), using a protocol such as LDAP, RMI, NDS, or DNS that can run code remotely. Organizations using Microsoft Sentinel created a detection query to look for devices with applications with this vulnerability.

NOTE **The sample query can be obtained at *http://aka.ms/MSSentinelL4J*.**

The history of a supply-chain attack

A supply-chain attack is not something new; attackers have been targeting software supply chains to gain access to the systems and data they are after for a long time. Malicious software inserted into legitimate applications will run with the same permissions and trust as the valid code. In May 2017, Microsoft security researchers identified Operation WilySupply, in which attackers compromised a text editor's software updater and installed a backdoor on targeted organizations.

Figure 1-3 shows an example of the timeline and process-tree views from Microsoft Defender for Endpoint (formerly known as Microsoft Defender Advanced Threat Protection) that was used to pinpoint the execution chain and lead researchers back to the compromised updater.

NOTE **You can read more about the investigation at *http://aka.ms/wilysupplycyberattackms*.**

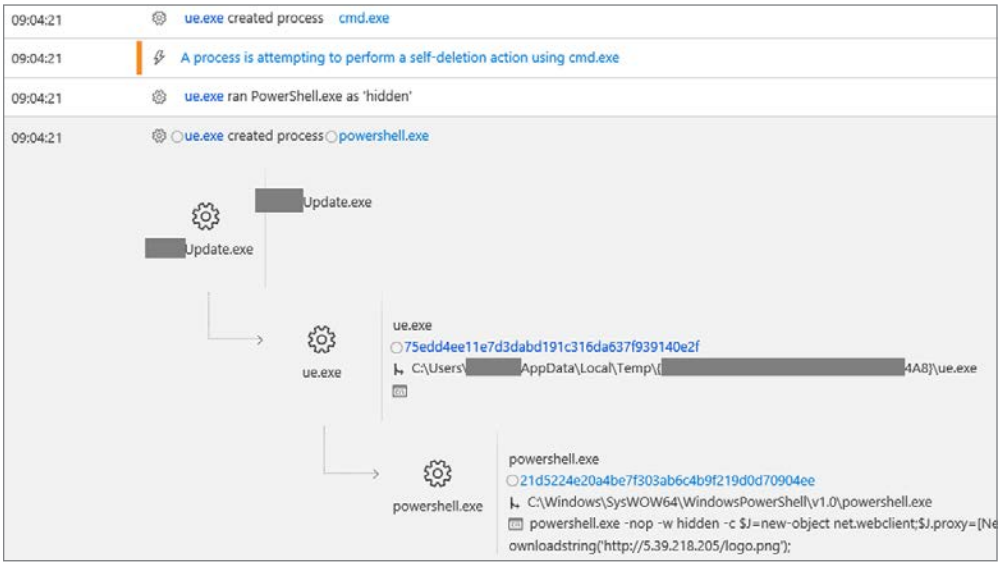


FIGURE 1-3 Detection of Operation WilySupply from Microsoft Defender for Endpoint

In March 2018, the first major software supply chain attack occurred when attackers compromised the update process for a peer-to-peer application. The poisoned updater then installed coin-mining malware.

TIP **You can read more about the attack details at *https://aka.ms/poisonedp2pcyberattack/*.**

The other major finding reported by Microsoft was that phishing emails continue to be the preferred method for attackers looking to gain a foothold within a company's network. As defenses have gotten better, attackers have evolved their phishing methods to evade detection. One common and highly effective method is using legitimate hosted and public cloud infrastructure as part of the attack. Additionally, attackers try to hide within the noise of commonly leveraged document sharing and collaboration sites and services.

In one specific case investigated by the Microsoft Detection and Response Team (DART), a large manufacturing organization was compromised via a targeted phishing attack. A phishing email was delivered to several company employees. The email body included a link that, if clicked, redirected them to a spoofed webpage. Once on the webpage, the employees were asked to authenticate with their domain credentials to gain access to a sensitive document. Once the attacker got access to several legitimate Office 365 accounts, they began sending additional emails to high-value individuals within the company. In this case, DART resolved the situation in just three hours and used Microsoft Sentinel to do it!

Security Challenges for SecOps

Security Operations, or SecOps, is a subdiscipline within the information security industry focused on running the day-to-day tasks of a security operations center (SOC). Before diving into specific challenges facing SecOps, it is important to understand the basic functions and operations required to conduct effective security operations. For most organizations, the SOC is the central hub responsible for identifying and responding to cybersecurity threats. MITRE (*www.mitre.org*) defines a SOC as "a team primarily composed of security analysts organized to detect, analyze, respond to, report on, and prevent cybersecurity incidents."

Although there are different ways to structure a SOC, analysts are typically divided into tiers based on their experience level and associated responsibilities. A commonly found pattern would include:

- **Tier 1: High-speed remediation—**Tier 1 analysts are typically new security professionals and the most junior staff members in the SOC. Their job is to perform the initial triage of an alert or reported incident and resolve the alert based on established operating procedures for common alert scenarios. Tier 1 is a high-volume, low-touch operation, and the analyst should spend no more than a few minutes on an alert before escalating to tier 2 for deeper investigation. Tier 1 analysts handle the majority of the SOC's workload.
- **Tier 2: Advanced Analysis, Investigation, and Remediation—**Tier 2 analysts are higher seniority security analysts and take escalations from the tier 1 analysts.

Resolution for this tier will take hours or days to complete depending on the situation. This could include the need to capture and analyze media images or potential malware samples for deeper review.

- **Tier 3: Proactive Hunting and Advanced Forensics—**Tier 3 analysts have specialized skills in attacker techniques, tactics, and procedures; malware analysis; threat intelligence; and threat hunting. These analysts leverage many tools and data sources to proactively look for malicious actors who have evaded traditional detection techniques. Also, these specialists evaluate trends and use advanced analytics and correlation techniques to find malicious activities.
- **Support Engineers—**An SOC will also have support engineers who are responsible for maintaining the infrastructure needed to run an effective cyberdefense program. This will include the installation, maintenance, and tuning of the SIEM and other specialized tools.

Microsoft has adopted a fusion center model for cyberdefense operations that bring SOC teams together from across the company into a shared facility known as the Cyber Defense Operations Center, or CDOC. This model allows Microsoft to maintain deep specialization while sharing situational awareness and subject matter expertise across teams. As you can see in Figure 1-4, Microsoft CDOC has also adopted a tiered response model that begins with automation, or Tier 0.

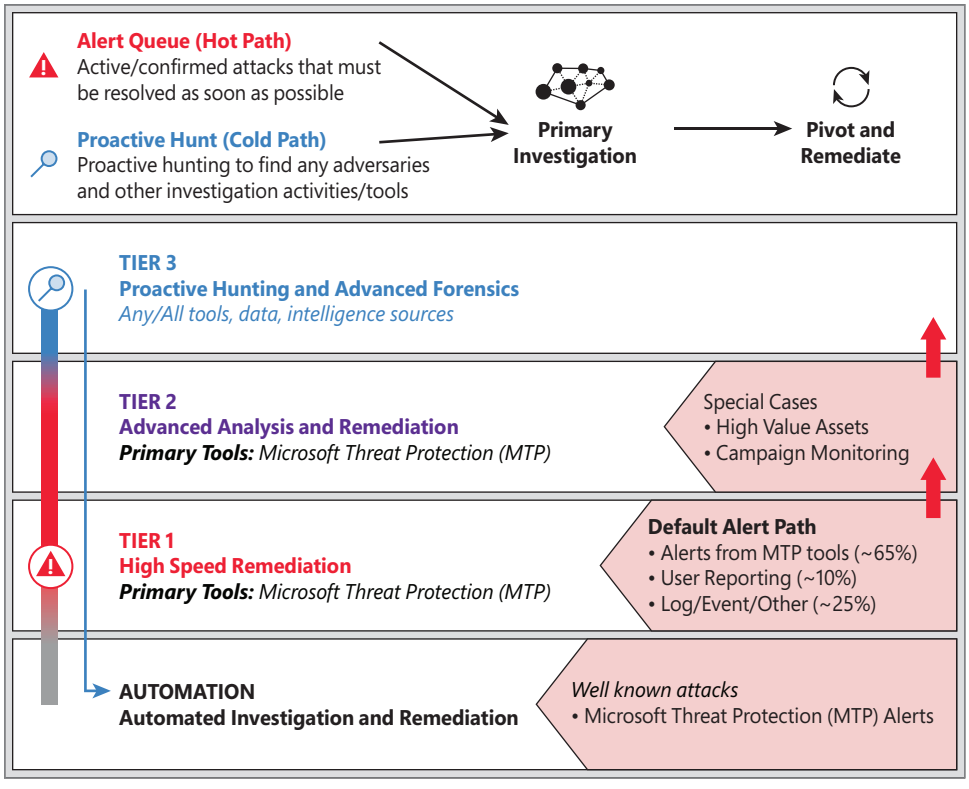


FIGURE 1-4 Microsoft CDOC tiered SOC model

Tier 0 requires no human intervention and is used to triage and respond to commonly occurring, extremely high-fidelity alerts (+95% true positive). Tier 1 analysts focus on high-speed, low-touch remediation efforts and escalate more advanced cases to Tier 2 analysts. Tier 3 analysts work on proactively threat hunting, advanced correlation, trend analysis, and first-party threat intelligence production and dissemination.

NOTE **You can learn more about the CDOC at *http://aka.ms/minutesmatter*.**

Resource challenges

(ISC)2 is an international, nonprofit for information security practitioners conducted a cybersecurity Workforce Study in 2020 that concluded that there are 879,000 cybersecurity professionals in the workforce and an unfilled need for another 359,000 workers. Also, according to a study conducted by Cybersecurity Ventures, the number of unfilled cybersecurity jobs grew from one million positions in 2013 to 3.5 million in 2021. In 2021, Microsoft launched a national campaign with US community colleges to help increase skills and recruit into the cybersecurity workforce 250,000 people by 2025.

Staffing shortages have hit SOCs especially hard for a few reasons. First, SOCs run operations 24x7x365 and therefore require a heavy investment in personnel. Not only must all shifts be covered, but enough staffing must be added to account for analyst time off (vacation and sick leave). Also, tier 1 analysts, who make up the bulk of a SOC's personnel, are difficult to retain. Entry-level analysts are required to work less desirable days and shifts (weekends, holidays, and nights). Entry-level analysts are also prone to burnout as they sit in front of a computer monitor triaging an unending number of alerts. Analysts are also under pressure to move quickly, while knowing that misdiagnosing one alert could result in a major breach. Finally, security analysts require a unique set of knowledge and skills that are difficult to find in today's competitive employment environment. Not only should an analyst understand common attacker techniques, but they should also have strong intuition, a desire to dig into the details and volumes of alerts and logs, and be driven to continuously learn.

With these staffing challenges, CISOs and their SOC leaders are looking for solutions that make their analysts more efficient, reduce the volume of mundane, manual tasks, and provide robust automation and orchestration capabilities.

Finding the proverbial needle in the haystack

Corporate security teams are drowning in the volumes of data being generated by the digital assets they are paid to protect. Data volumes are increasing every day as more operations are being digitized and with the deployment of smart sensors and Industrial Internet of Things (IIoT) devices within corporate networks. Security has truly become a big data problem. For example, Microsoft's CDOC receives more than 15 billion individual events per month.

For the past decade, SOC leaders have leveraged SIEM technologies in an attempt to establish a "single pane of glass" for their analysts. Unfortunately, challenges with early SIEM technologies made this difficult because of the constant need to buy and install more and more hardware to handle increasing data volumes. Often, security teams were required to forgo connecting data sources because of the costs associated with scaling out their SIEMs. In addition, early search and correlation engines could not handle the volume of data, and analyst queries would time out before completing their task. In addition, static correlation rules often missed anomalies that, when combined with other contextual data, indicated that an attacker had successfully infiltrated a system. Typically, early SIEMs were not built with machine-learning models to help identify such anomalies. In addition, as mentioned earlier in this chapter, most corporate security teams cannot afford to hire data scientists to build, test, and deploy their own models. Finally, many SIEM deployments were done with a "deploy and forget" mentality. This resulted in analysts working on many false positives that strained personnel and made it difficult to identify the true, high-value events. To be effective, SIEMs and their associated log providers require constant attention and fine-tuning.

Threat intelligence

Knowledge of your adversaries is essential. Cyber threat intelligence, or CTI, is the collection, analysis and synthesis, and dissemination of information related to cyberattackers' tactics, techniques, and procedures (TTPs). CTI also includes an evaluation of a threat actor's intent, motivations, and overall capabilities. Studying threat actors makes it often possible to make proactive strategic and tactical decisions to prevent and detect attacks. Following are some examples:

- Strategic CTI is primarily intended for senior decision-makers and executives and focuses on developing an overall picture of threat actors' capabilities and maintaining overall situational awareness of emerging threats. Strategic CTI is often performed by national computer emergency response and information sharing centers to provide timely warnings to their constituencies.
- Operational CTI assesses specific incidents to identify and report on attacker campaigns and commonly used malware and/or tools by identified and named threat actors (such as Advanced Persistent Threat [APT 34]).
- Tactical CTI assesses real-time events and activities and provides actionable information to SOC operators. Key tactical CTI products include threat detection signatures like Yara (see *https://virustotal.github.io/yara*) rules for malware and indicators of compromise (IOC).

As shown in Figure 1-5, CTI informs each of the SOC functions by providing context and actionable alerts to leaders, analysts, and hunt teams.

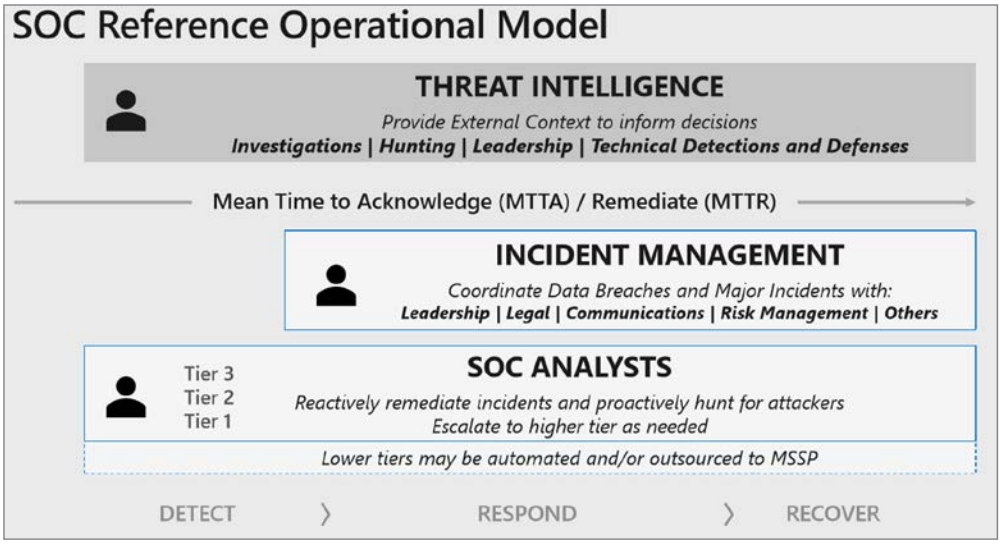


FIGURE 1-5 Cyberthreat intelligence's place in the SOC reference operational model

Structured Threat Information Expression (STIX) makes it easier to share CTI across organizations. STIX is open source and free for anyone to use. STIX information is stored as JSON, which makes it easy to integrate with existing security tools. Listing 1-1 shows an example of a STIX indicator object representing a malicious URL from the project's documentation page.

```
Listing 1-1  STIX indicator object representing a malicious URL{
 "type": "bundle",
 "id": "bundle--44af6c39-c09b-49c5-9de2-394224b04982",
 "spec_version": "2.0",
 "objects": [
  {
   "type": "indicator",
   "id": "indicator--d81f86b9-975b-4c0b-875e-810c5ad45a4f",
   "created": "2014-06-29T13:49:37.079Z",
   "modified": "2014-06-29T13:49:37.079Z",
   "labels": [
    "malicious-activity"
   ],
   "name": "Malicious site hosting downloader",
   "pattern": "[url:value = 'http://x4z9arb.cn/4712/']",
   "valid_from": "2014-06-29T13:49:37.079000Z"
  },
  {
```

```
   "type": "malware",
   "id": "malware--162d917e-766f-4611-b5d6-652791454fca",
   "created": "2014-06-30T09:15:17.182Z",
   "modified": "2014-06-30T09:15:17.182Z",
   "name": "x4z9arb backdoor",
   "labels": [
    "backdoor",
    "remote-access-trojan"
   ],
   "description": "This malware attempts to download remote files after establishing a
foothold as a backdoor.",
   "kill_chain_phases": [
    {
     "kill_chain_name": "mandiant-attack-lifecycle-model",
     "phase_name": "establish-foothold"
    }
   ]
  },
  {
   "type": "relationship",
   "id": "relationship--6ce78886-1027-4800-9301-40c274fd472f",
   "created": "2014-06-30T09:15:17.182Z",
   "modified": "2014-06-30T09:15:17.182Z",
   "relationship_type": "indicates",
   "source_ref": "indicator--d81f86b9-975b-4c0b-875e-810c5ad45a4f",
   "target_ref": "malware--162d917e-766f-4611-b5d6-652791454fca"
  }
 ]
}
```

People who perform threat hunting are among the most common consumers of threat intelligence. In the above example, a hunt team would take the STIX object and hunt within Azure Sentinel, looking for indicators that a corporate computer attempted to access the malicious domain. This hunting query would search all associated logs to determine if any user and/or computer communicated with the domain `'http://x4z9arb.cn/4712/'`. If communication with `'http://x4z9arb.cn/4712/` occurred, further queries would be written to determine the scope of the attack, such as compromised credentials, lateral movement, and so on.

Trusted Automated Exchange of Intelligence Information (TAXII) is a companion to STIX and acts as a transport-sharing mechanism for sharing CTI written in STIX format. TAXII is not an application itself; instead, it is a set of specifications for exchanging CTI.

> ***NOTE*** **You can find more details about STIX and TAXII at *https://oasis-open.github.io/cti-documentation/*.**

> ***IMPORTANT*** **You will learn more about how to leverage CTI in Microsoft Sentinel in Chapter 5, "Hunting."**

Introducing Microsoft Sentinel

Microsoft Sentinel is Microsoft's cloud-native SIEM solution. It is the first SIEM solution built into a major public cloud platform. Microsoft Sentinel also contains a security orchestration and automated response (SOAR) capability. Microsoft Sentinel's SOAR capability is fully customizable and allows security teams to write Playbooks that can, if desired, automate the entire response to a security event. For example, once Microsoft Sentinel identifies a malicious domain, a Playbook can be triggered that would automatically add a block rule for that domain to the company's firewalls.

Gartner defines a SIEM as technology that supports "threat detection and security incident response through the real-time collection and historical analysis of security events from a wide variety of event and contextual data sources." Most traditional SIEMs started as on-premises solutions composed of hardware and software that supported log ingestion and storage and provided a user interface and search engine to correlate system events and security alerts. As log ingestion and storage requirements increased, customers needed to buy larger hardware or distribute the workload across multiple servers.

Over the last several years, more and more vendors have retooled their SIEMs to make them available in a Software as a Service, or SaaS, model. However, these SIEMS are typically built on top of a public cloud provider's infrastructure and don't offer the same automatic scaling and storage benefits found in Microsoft Sentinel. With Microsoft Sentinel, there are no requirements for the customer to open support tickets to scale out their services like other SaaS-based SIEMs. All of this is handled automatically by Microsoft, and the customer can focus on the main task at hand—identifying and responding to cyberthreats.

Core capabilities

While the purpose of this chapter is not to go into depth in any particular area, it is important that you understand the core capabilities of Microsoft Sentinel. Microsoft Sentinel provides security teams with unprecedented visibility into their digital estate. The core capabilities of the solution include the following:

- Data collection and storage across all users, devices, applications, and infrastructure, whether on-premises or in the cloud
- Threat detection leveraging Microsoft's analytics and threat intelligence
- Investigation of threats by hunting for suspicious activities at scale
- Rapid response to incidents leveraging built-in orchestration and automation of common tasks

Now that you have an idea of Microsoft Sentinel's core capabilities as a cloud-native SIEM, let's get into the details in the next chapter.

Chapter 2

Introduction to Microsoft Sentinel

Given the threat landscape presented in Chapter 1, there is a clear need for a system that can collect data from different sources, perform data correlation, and present this data in a single dashboard.

Microsoft Sentinel delivers intelligent security analytics and threat intelligence across the enterprise, providing a single solution for alert detection, threat visibility, proactive hunting, and threat response. Microsoft Sentinel natively incorporates proven foundation services from Azure, such as Log Analytics and Logic Apps. Also, Microsoft Sentinel enriches your investigation and detection with Artificial Intelligence (AI) in conjunction with Microsoft's threat intelligence stream.

In this chapter, you will learn more about the architecture, design considerations, and initial configuration of Microsoft Sentinel.

Architecture

Because Microsoft Sentinel is part of Azure, the first prerequisite to deployment is to have an active Azure subscription. As with any other security information and event management (SIEM), Microsoft Sentinel needs to store the data that it will collect from the different data sources that you configure. Microsoft Sentinel will store this data in your preferred Log Analytics workspace. Depending on your business needs and technology requirements, you can create a new workspace or use an existing one.

To help you to better understand Microsoft Sentinel's architecture, you must first understand the different components of the solution. Figure 2-1 shows a diagram of the major Microsoft Sentinel components.

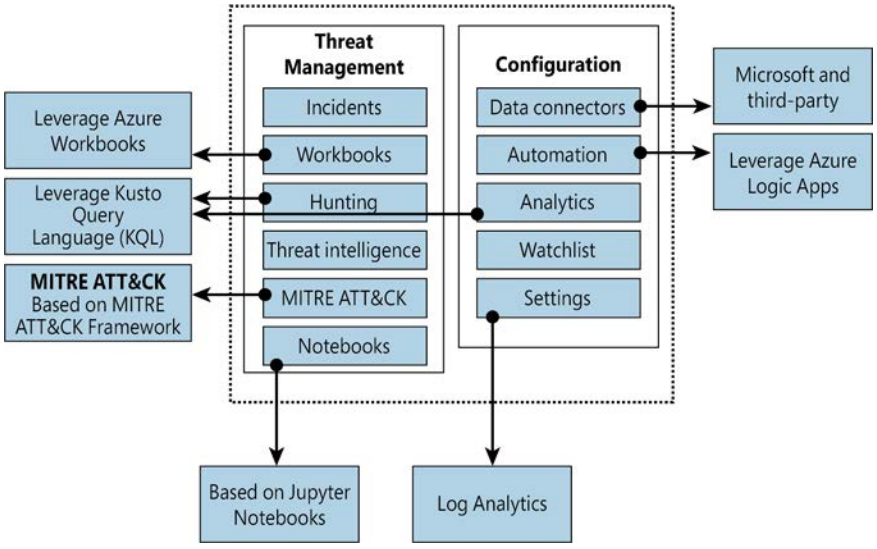


FIGURE 2-1 Major components of Azure Sentinel

The components shown in Figure 2-1 are presented in more detail below:

- **Incidents** This is a centralized place to manage your security incidents. An incident will have relevant data that you can use to understand its impact. You will learn more about incidents in Chapter 4, "Incident management."
- **Workbooks** Built-in dashboards based on Azure Workbook that provide data visualization for your connected data sources. These Workbooks enable you to deep dive into the events generated by those services. You will learn more about Workbooks in Chapter 8, "Data visualization."
- **Hunting** This is a powerful tool for investigators and security analysts who need to proactively look for security threats. The searching capability is powered by Kusto Query Language (KQL). You will learn more about hunting in Chapter 5, "Hunting."
- **Threat Intelligence** Cyber threat intelligence (CTI) is an important capability leveraged by defenders to better understand the behavior of threat actors. This option allows Tier 2 and Tier 3 SOC analysts to curate their CTI within Microsoft Sentinel by tagging existing data. You will learn more about threat intelligence in Chapter 5, "Hunting."
- **MITRE ATT&CK** This page contains active scheduled queries and near-real-time (NRT) rules coverage according to the MITRE ATT&CK framework. You will learn more about MITRE ATT&CK in Chapter 5, "Hunting."
- **Notebooks** By integrating with Jupyter notebooks, Microsoft Sentinel extends the scope of what you can do with the collected data. The notebooks feature combines full programmability with a collection of libraries for machine learning, visualization, and data analysis. You will learn more about notebooks in Chapter 6, "Notebooks."

- **Data Connectors** Built-in connectors are available to facilitate data ingestion from Microsoft and partner solutions. You will learn more about data connectors later in this chapter.
- **Automations** A collection of procedures that can be automatically executed when an alert is triggered by Microsoft Sentinel by leveraging Azure Logic Apps. This will help you to automate and orchestrate tasks/workflows. You will learn more about Playbooks in Chapter 7, "Automating response with Playbooks."
- **Analytics** Analytics enable you to create custom alerts using Kusto Query Language (KQL). You will learn more about analytics in Chapter 3, "Analytics."
- **Watchlist** This list allows you to correlate data from a data source you provide with the events in your Microsoft Sentinel environment. You will learn more about analytics in Chapter 3, "Analytics."
- **Settings** This section has a variety of configuration options, including the Log Analytics workspace. Microsoft Sentinel uses this workspace to store the data you collect from the different data sources. You will learn more about workspace configuration later in this chapter.

Roles and permissions

Microsoft Sentinel uses Azure role-based access control (Azure RBAC), which provides a set of pre-defined privileges to perform certain actions in the environment. These built-in roles can be assigned to users, groups, and services in Azure.

Microsoft Sentinel also adds its own roles to Azure that were designed to perform specific actions based on a scenario. All Microsoft Sentinel built-in roles grant read access to the data in your Microsoft Sentinel workspace. The Microsoft Sentinel built-in roles are:

- **Microsoft Sentinel Reader** View data, incidents, Workbooks, and other Microsoft Sentinel resources.
- **Microsoft Sentinel Responder** In addition to the actions enabled by Microsoft Sentinel Reader, it can also manage incidents (assign, dismiss, and so on).
- **Microsoft Sentinel Contributor** In addition to the actions enabled by Microsoft Sentinel Responder, it can also create and edit Workbooks, analytics rules, and other Microsoft Sentinel resources.
- **Microsoft Sentinel Automation Contributor** Allows adding Playbooks to automation rules. This is not a role that is meant to be used by user accounts.

Because many other scenarios are enabled by Microsoft Sentinel, you may need to use additional roles according to a given need. Use Table 2-1 as a reference for these scenarios:

TABLE 2-1 Microsoft Sentinel scenarios

SCENARIO	CONSIDERATIONS
Automate responses to threats by leveraging Playbook capability	The Playbook capability in Microsoft Sentinel uses Logic Apps; in this case, you might need to add members of the team who are responsible for creating automated responses to the *Logic App Contributor* role.
Connecting Microsoft Sentinel to an external data source	Regardless of the data connector source, the users responsible for creating connectors will need to have write permissions on the Microsoft Sentinel workspace.
Temporary employee or guest assigning incidents	There might be scenarios where you will need to have temporary personnel in charge of triaging incidents and assigning them to the right team. In this case, in addition to assigning the user to the *Microsoft Sentinel Responder* role, you will also need to assign them to the Directory Reader role.
Manipulating Workbooks	You might have a user who works with data visualization and oversees creating and deleting Workbooks in Microsoft Sentinel. In this case, you can either add this user to the *Microsoft Sentinel Contributor* role or the Microsoft Sentinel role with less privilege and add them to the Azure Monitor *Workbook Contributor role*.

Be mindful of role aggregation scenarios in which a user has been assigned to a higher Azure role and a stricter Microsoft Sentinel role, which means the user will still be able to perform stricter operations. If you want to harden your user's permissions to reflect their operation only in Microsoft Sentinel, you should carefully remove this user's prior permissions, making sure you do not break any needed access to another resource. Figure 2-2 shows an example of how to organize your users according to their roles and Azure resources.

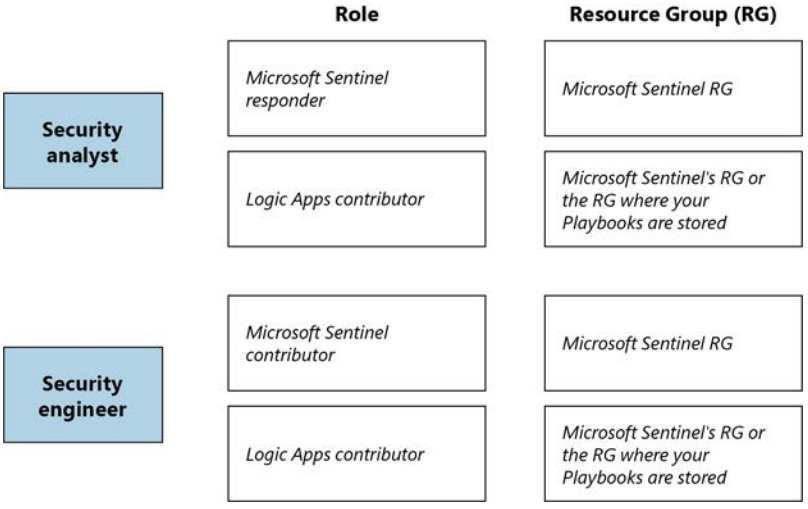


FIGURE 2-2 Role-based access control for different users

Workspace design considerations

Log Analytics workspace is the core foundation of Microsoft Sentinel. Later in this chapter, you will see that the first step to enabling Microsoft Sentinel is to select the workspace. A Log Analytics workspace provides a geographic location for data storage and data isolation by granting access rights to different users if necessary, as well as a set of configuration options.

One best practice is to always reduce the number of workspaces in use, but there are some specific scenarios that will lead you to have more than one workspace. The most common ones are listed below:

- Data sovereignty requirements and regulatory compliance standards that the company needs to abide by
- Data ownership requirements created by different company boundaries, such as subsidiaries and headquarters
- Companies that have multiple Azure Active Directory tenants
- Companies may need to use chargeback and have a more granular control over the Azure bill
- Companies that need more granular access control
- Companies that have different retention policies per subsidiary
- A company is a Managed Security Service Provider (MSSP)

Keep in mind that you will first need to deploy Azure Lighthouse to provide visibility across tenants for a multi-tenant scenario. If your design process leads you to conclude that you must have multiple workspaces, Microsoft Sentinel allows you to see incidents on multiple workspaces, which facilitates central incident monitoring and management across multiple workspaces. The advantage of this centralized view is that you can manage incidents directly and see incident details seamlessly in the context of the originating workspace.

Microsoft Sentinel also supports querying multiple workspaces. This is done in a centralized view and a single query, which allows you to search and correlate data from multiple workspaces in a single place and in one single query. The list below provides the other Microsoft Sentinel features that support this cross-workspace ability:

- Analytics rules
- Workbooks
- Hunting

IMPORTANT **You can have up to 30 cross-workspace analytics rules, while you can view up to 100 cross-workspace incidents (in preview). Keep in mind that querying multiple workspaces in the same query might affect performance.**

The data connector is what will send data to the workspace and is another important consideration when designing your workspace architecture. Connectors that are based on diagnostics settings (such as Azure Firewall, Azure Storage, Azure Activity, or Azure Active

Directory) cannot be connected to workspaces that are not located in the same tenant as the source workspace. For additional considerations related to multiple workspaces, check the decision tree in this article, *http://aka.ms/SentinelLAWDecisionTree*, and the sample templates at *http://aka.ms/SentinelLAWTemplates*.

While Microsoft Sentinel can be utilized for multiple regions, your design requirements might require you to adopt one workspace per region. This can happen for numerous reasons, including regulations and data separation by team. For this type of scenario, you need to consider that egress costs apply when the Log Analytics or Azure Monitor agent is required to collect logs (for example, a VM). You also need to consider the bandwidth costs, which vary depending on the following factors: source and destination region and collection method.

While there are scenarios that will require you to have multiple workspaces, you may also find scenarios where the same workspace will be shared by Microsoft Sentinel and Microsoft Defender for Cloud. For best practices on how to design your solution for this requirement, see *http://aka.ms/LAWBPD4CSentinel*.

Hardening considerations

To improve the security hygiene of your Microsoft Sentinel and its Azure environment, it is recommended that you use the security baseline provided by Azure Security Benchmark. This benchmark provides security recommendations for the following areas:

- Network security
- Identity management
- Privilege access
- Data protection
- Asset management
- Logging and threat detection
- Posture and vulnerability management
- Backup and recovery

Additional considerations

Before enabling Microsoft Sentinel, you should have a good understanding of which data sources you will use to connect with Microsoft Sentinel. If you are unsure, you can always start with the free data connectors, which are: Azure Activity Logs, Office 365 Audit Logs, including all SharePoint activity, Exchange admin activity, and Teams, Security alerts (including alerts from Microsoft Defender for Cloud, Microsoft 365 Defender, Microsoft Defender for Office 365, Microsoft Defender for Identity, Microsoft Defender for Endpoint), Microsoft Defender for Cloud and Microsoft Defender for Cloud Apps alerts.

Another important consideration during this design exercise is to identify if you will need to have a partner or custom connector. This will require you to configure Syslog and CEF connectors with the highest priority first.

Once you finish defining the use cases, data sources, and data size requirements, start planning your budget, considering cost implications for each planned scenario. Make sure to include in your budget the cost of data ingestion for both Microsoft Sentinel and Azure Log Analytics and any Playbooks that will be deployed.

IMPORTANT **For the latest information on Microsoft Sentinel pricing, visit** ***https://azure.microsoft.com/en-us/pricing/details/microsoft-sentinel.***

Enabling Microsoft Sentinel

Now that you have finished planning your Microsoft Sentinel adoption, it is time to enable the service. Remember that you need an active Azure subscription before enabling Microsoft Sentinel. Follow the steps below to enable Microsoft Sentinel in your subscription:

1. Open the **Azure portal** and sign in with a user who has contributor permissions on the subscription and in the resource group where the workspace resides.
2. In the search bar, type **Sentinel** and click the **Microsoft Sentinel**; the **Microsoft Sentinel** blade appears, as shown in Figure 2-3.

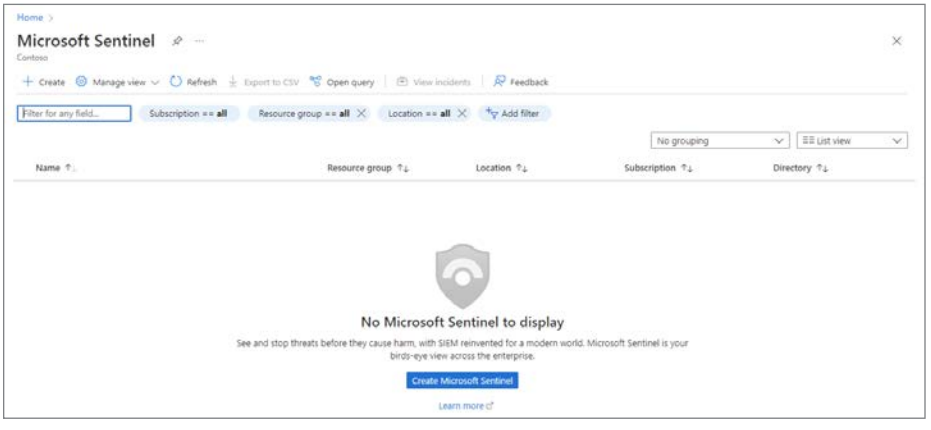


FIGURE 2-3 Microsoft Sentinel initial page

3. Click the **Create Microsoft Sentinel** button. Because there is no workspace selected, a page similar to Figure 2-4 appears.

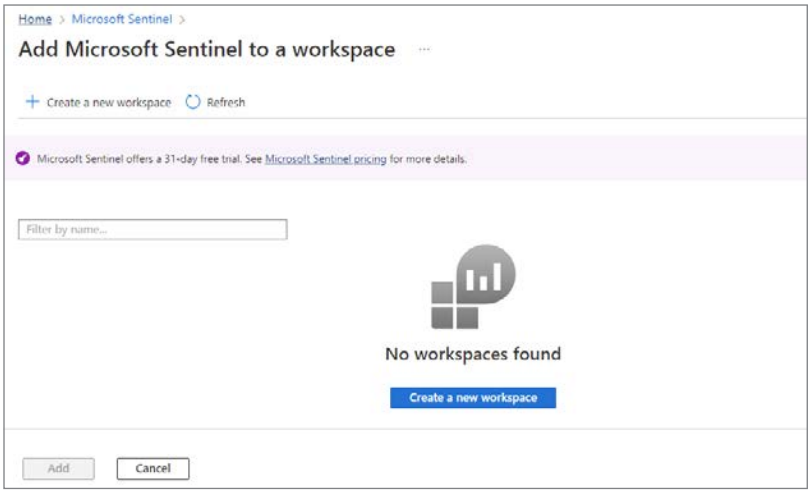


FIGURE 2-4 No workspaces available

4. Click the **Create A New Workspace** button.
5. You will be redirected to the **Create Log Analytics Workspace** page, as shown in Figure 2-5.

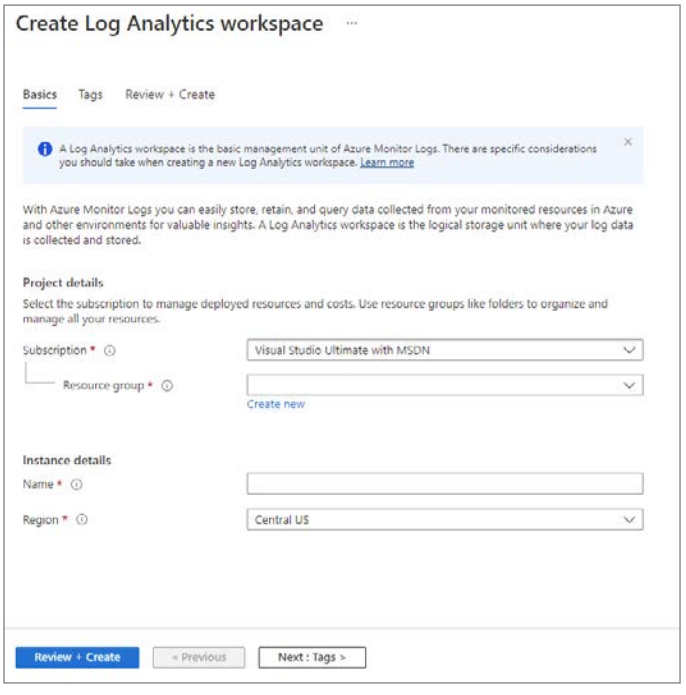


FIGURE 2-5 Create Log Analytics Workspace

6. Follow the steps on the screen to create a new workspace using the default selections. When you finish filling those options, click the **Review + Create** button.
7. Once you see a green check mark indicating that the validation has passed, you can click the **Create** button to conclude.
8. You will be redirected to the **Add Microsoft Sentinel To A Workspace** page. If the screen doesn't refresh, click the **Refresh** button, and you should see the workspace and the **Add** button. Click the **Add** button to continue. Because this is the first time you have used Microsoft Sentinel in this brand-new workspace, you will receive a notification similar to the one shown in Figure 2-6. (Be mindful that the date range will change according to the date you created your workspace.)

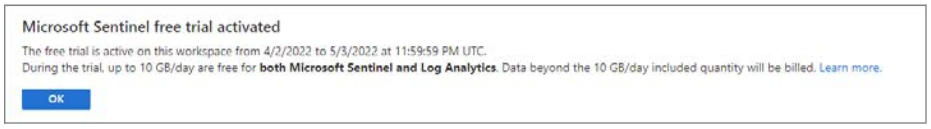


FIGURE 2-6 Trial activation notification

9. Click the **OK** button to continue, and you will see the **Microsoft Sentinel News & Guides** page, as shown in Figure 2-7.

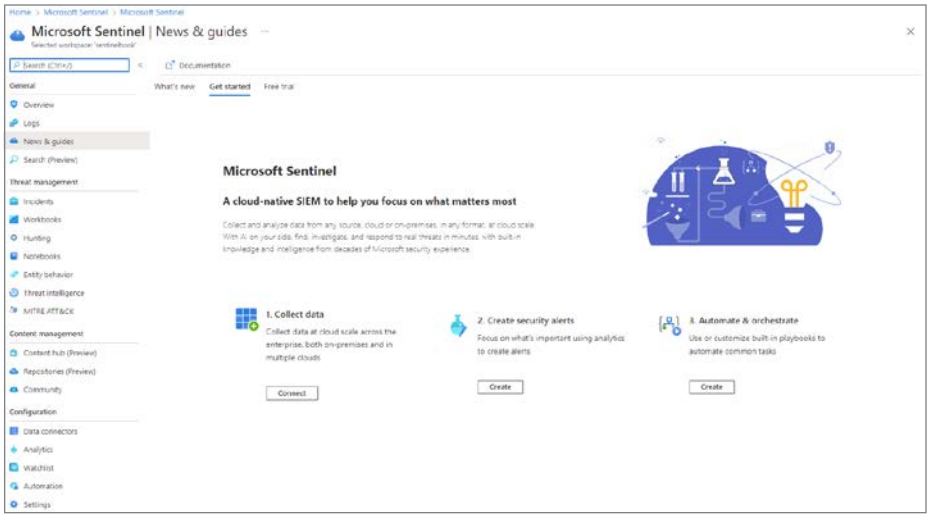


FIGURE 2-7 Microsoft Sentinel News & Guides page

At this point, you have a workspace, and Microsoft Sentinel is enabled on it. Next, you need to start ingesting data, and as mentioned before, you can start by ingesting data from the free connectors.

Ingesting data from Microsoft solutions

One way to quickly start validating Microsoft Sentinel's data ingestion is to start the configuration by using Microsoft built-in connectors. Each data connector will have its own set of prerequisites, including the type of license for the service from which you are trying to ingest data.

Start with the Azure Activity Log to visualize data from the subscription-level events that have occurred in Azure, which includes data ranging from Azure Resource Manager (ARM) operational data to updates on service health events. Follow the steps below to connect with the Azure Activity Log:

1. In the Microsoft Sentinel dashboard, click **Data Connectors** in the left navigation pane under the **Configuration** section. The **Data Connectors** page appears, as shown in Figure 2-8.

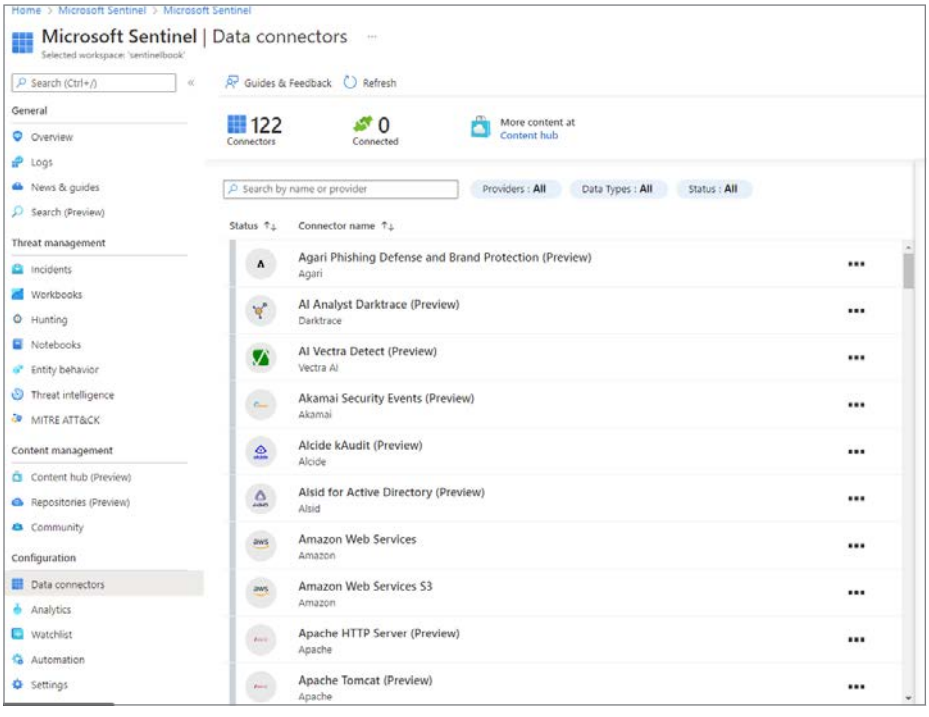


FIGURE 2-8 Data Connectors

2. In the search bar, type **Azure Activity**.
3. Click **Azure Activity**, and the **Azure Activity** blade appears on the right side, as shown in Figure 2-9.

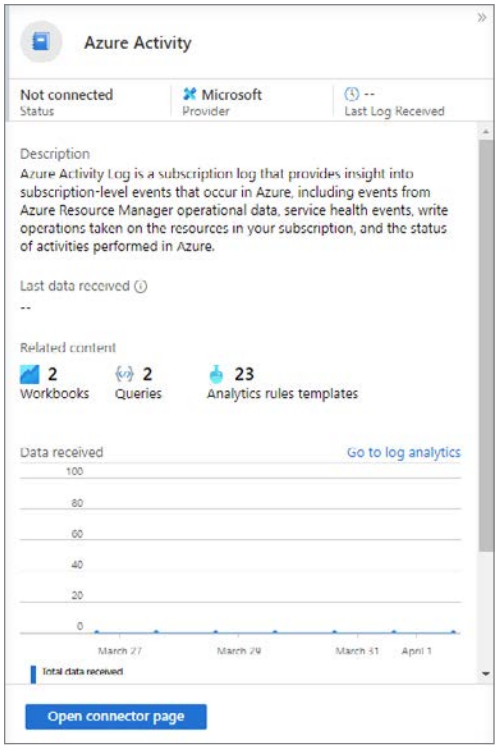


FIGURE 2-9 Azure Activity blade

4. Click the **Open Connector Page** button, and on the **Instructions** tab, click the **Launch Azure Policy Assignment Wizard** button. The **Configure Azure Activity Logs To Stream To Specified Log Analytics Workspace** page appears, as shown in Figure 2-10.

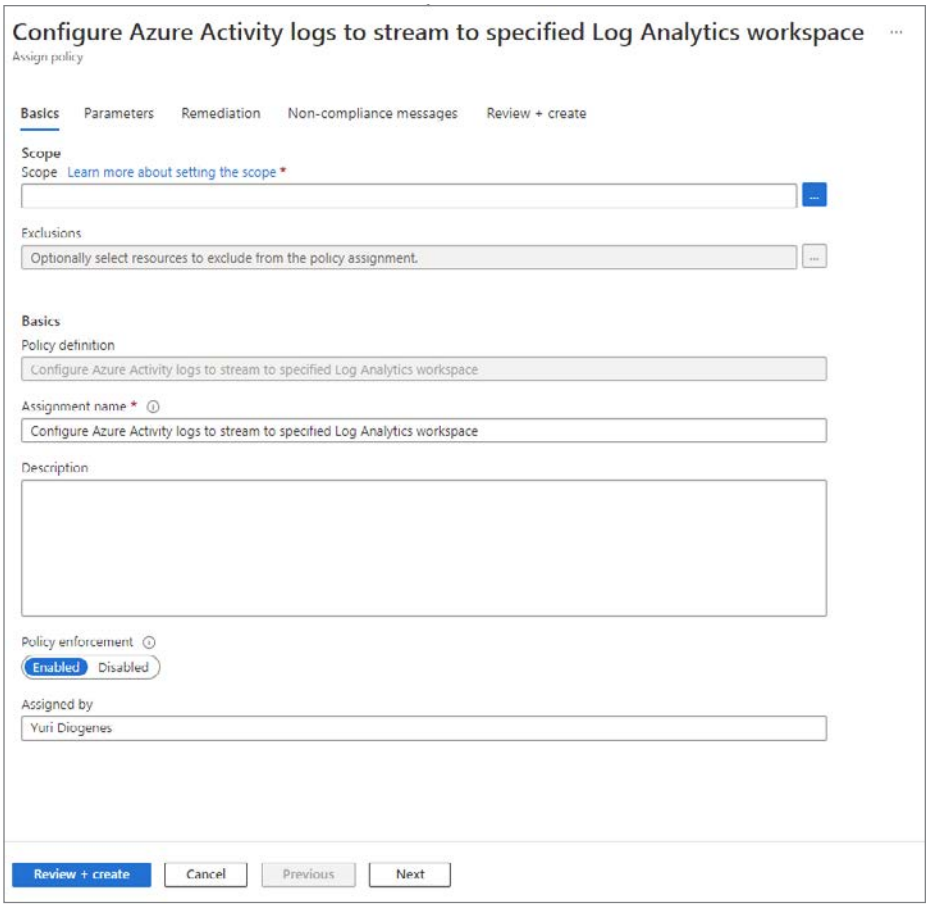


FIGURE 2-10 Enabling Azure Activity logs via Azure Policy

5. Under **Scope**, select the subscription, leave the other settings as is, and select the **Parameters** tab. On this tab, you will select the workspace that you created earlier.
6. Click the **Remediation** tab and select the **Create A Remediation Task** checkbox and leave the other options as is.
7. Click the **Review + Create** button, and then click **Create** to conclude.
8. Because this setting is deployed via Azure Policy, it may take several minutes to update the screen. You can close the connector's page and return to the main Microsoft Sentinel dashboard.

Connecting Microsoft Defender for Cloud

If you have Microsoft Defender for Cloud enabled in your subscription, you can start ingesting the Security Alerts generated by Defender for Cloud, which provides a rich set of threat detections.

Defender for Cloud will generate alerts according to the enabled plans. Follow the steps below to connect to Defender for Cloud and start streaming security alerts to Microsoft Sentinel:

1. In the Microsoft Sentinel dashboard, in the left navigation pane, click **Data Connectors** in the **Configuration** section.
2. In the search bar, type **Defender for Cloud**, click the **Microsoft Defender for Cloud** option, and the **Microsoft Defender for Cloud** blade appears on the right side. In this blade, click the **Open Connector Page** button, and the **Instructions** tab appears selected, as shown in Figure 2-11.

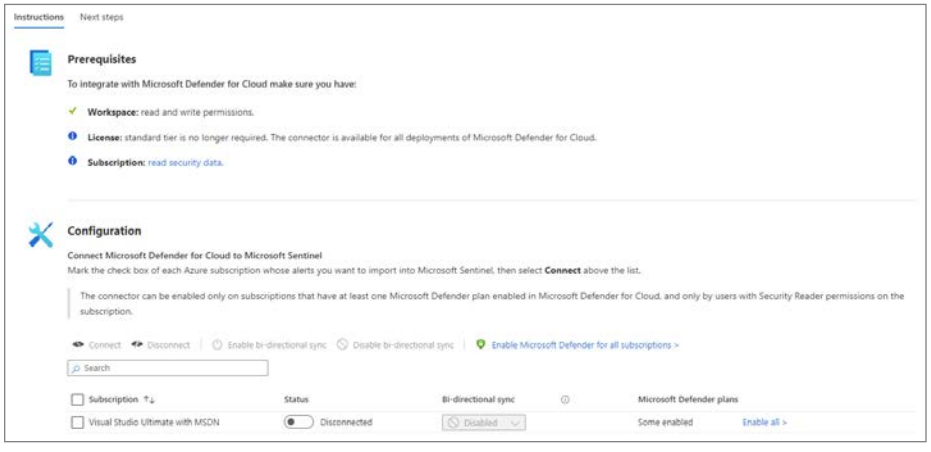


FIGURE 2-11 Enabling the Microsoft Defender for Cloud connector

3. In the **Configuration** section, select the **Subscription**, and in the **Status** column, click the Status toggle to **Connected**.
4. By default, **Bi-Directional Sync** is enabled. Leave this as-is to allow Microsoft Sentinel and Defender for Cloud to be in sync regarding the alert's status.

5. On the same page, scroll down and you will also see the **Enable** button under **Create Incidents–Recommended!** section, as shown in Figure 2-12.

FIGURE 2-12 Option to create incidents automatically

6. The advantage of using this option is that the alerts arriving from Defender for Cloud will be surfaced in Microsoft Sentinel as incidents. Click the **Enable** button to commit this change. A quick validation will be done, and the **Enable** button becomes unavailable.
7. Close the page and return to the main **Data Connectors** page.

After some time, you will be able to refresh the connector, and you will see that the status has changed to **Connected**, as shown in Figure 2-13.

FIGURE 2-13 Defender for Cloud Status changes to Connected

Connecting to Azure Active Directory

Azure Active Directory (Azure AD) is the identity and access-management service in the cloud. Each Azure tenant has a dedicated and trusted Azure AD directory. The Azure AD directory includes the tenant's users, groups, and apps, and it is used to perform identity and access-management functions for tenant resources. If you want to export sign-in data from Active Directory to Azure Sentinel, you must have an Azure AD P1 or P2 license.

The initial steps to configure Azure Activity Directory connector are the same as any other connector. The only difference is the page that opens up once you access the connector's page. For Azure Active Directory, you will see a page similar to Figure 2-14.

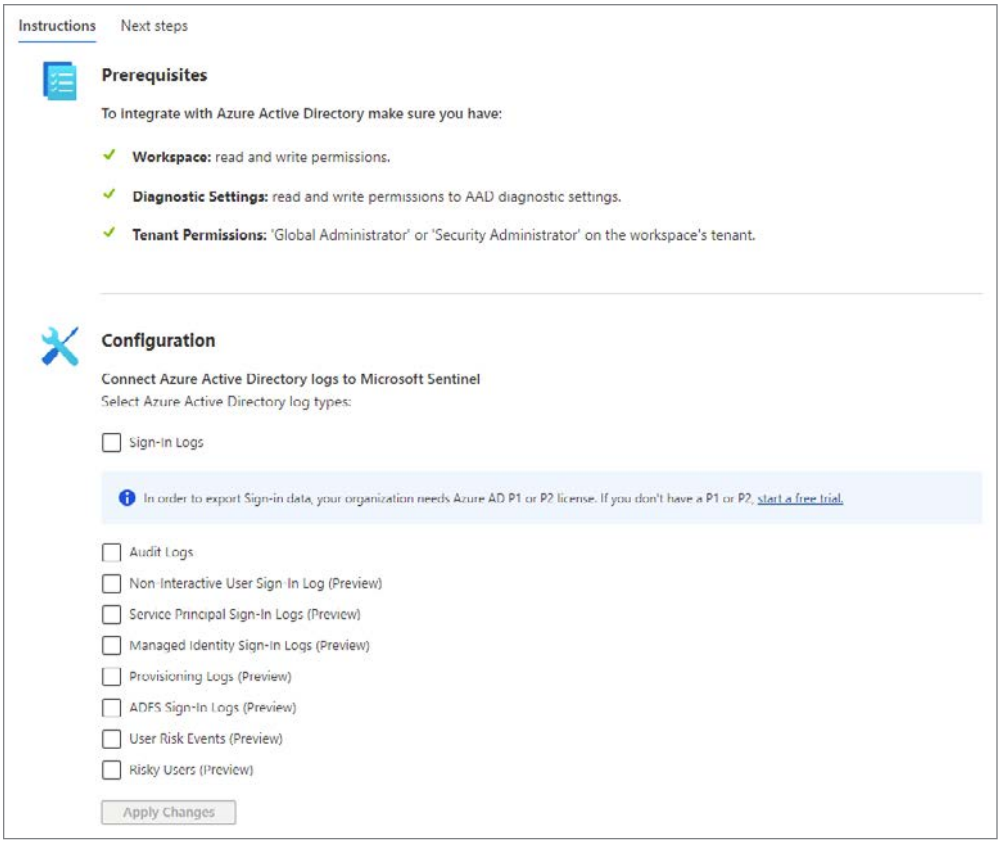


FIGURE 2-14 Azure Active Directory Connector page

This connector allows the stream of sign-in logs containing information about interactive user sign-ins where a user provides an authentication factor. At the time of publication, there were also a series of additional logs in preview, including:

- Non-interactive user sign-in logs
- Service principal sign-in logs
- Managed Identity sign-in logs
- Audit logs
- Provisioning logs

After you finish making your selection, click the **Apply Changes** button to commit these changes.

Accessing ingested data

After connecting with the data sources that you need, you can start validating the connection flow to ensure the data is being saved in the workspace. To perform this validation, you have two options:

- You can open the connector's page and click the **Next Steps** tab. There, you will find some sample queries that you can use.
- You can also access the workspace directly from Microsoft Sentinel and perform some free queries using Kusto Query Language (KQL).

A Kusto query is a read-only request to process data and return results. The request is stated in plain text, using a data-flow model designed to make the syntax easy to read, author, and automate. The query uses schema entities that are organized in a hierarchy similar to SQL's databases, tables, and columns. You will learn more about this language in Appendix A, "Introduction to Kusto Query Language."

Follow these steps to access the workspace from Azure Sentinel and perform the validation for Azure Activity Log, which was the first data source that you connected in this chapter:

1. In the Microsoft Sentinel dashboard, click **Logs** in the left navigation pane under the **General** section.
2. If this is the first time you are accessing this option, the **Welcome To Log Analytics** page appears. Close this welcome page and close the sample queries page.
3. On the **Logs** page, type **AzureActivity** and click the **Run** button. You should see all activities that were performed and collected in the last 24 hours (which is the default timeframe). The result should look similar to Figure 2-15.

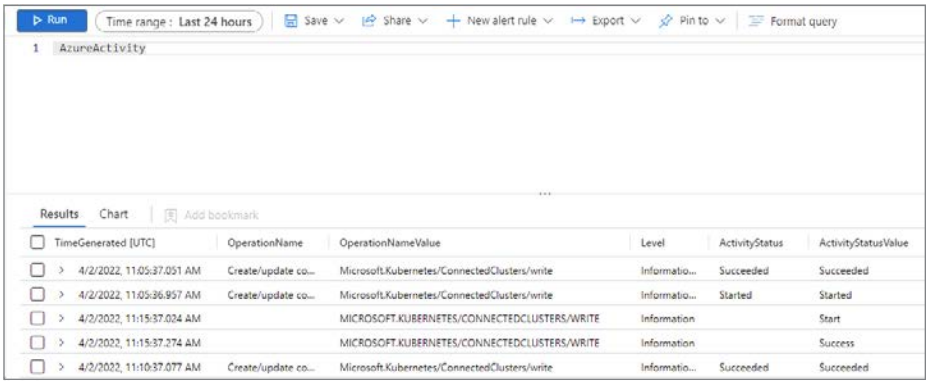


FIGURE 2-15 Last 24 hours of activities

As you can see, the logs are flowing, and you can obtain all results with a single query. However, in a real scenario, you want to narrow the results. An easy way to learn KQL while performing queries is to leverage the context-sensitive IntelliSense capability. To do that, write the query, and IntelliSense will open a dropdown menu showing the available options, as shown in Figure 2-16.

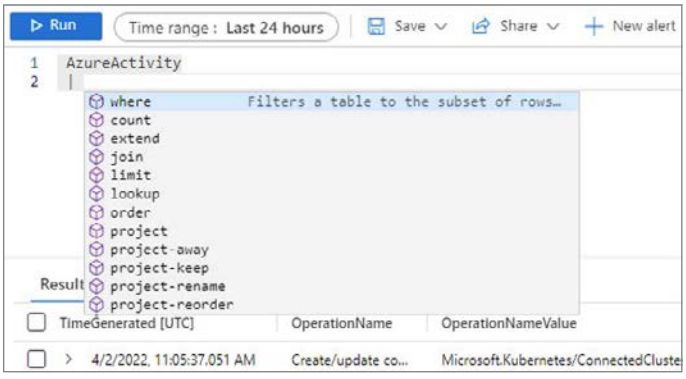


FIGURE 2-16 IntelliSense suggestions

To narrow the search to look only for activities that are related to the operation, you can use `OperationName`. For example, if you are trying to identify who deleted a VM, you can type the query below and click **Run**.

```
AzureActivity
| where OperationName contains "Create or Update Virtual Machine"
```

The results will be narrowed to show only activities that contain the specified string, which will make it easier to investigate. A sample result is shown in Figure 2-17.

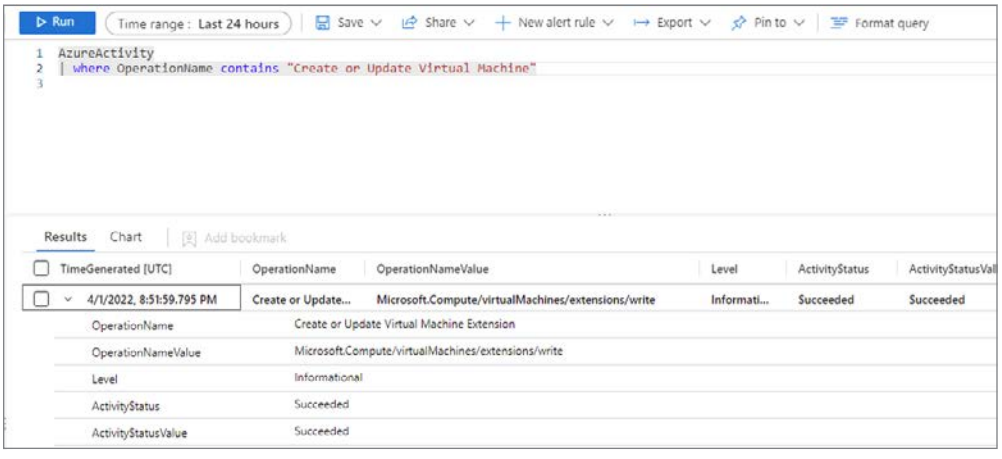


FIGURE 2-17 Customizing your query

To validate the other data sources that were ingested in this chapter, you can use the following sample queries:

- **Azure Active Directory**
  - Query: `SigninLogs`
    - (Use this query to visualize all Azure AD sign-in logs.)
  - Query: `AuditLogs`
    - (Use this query to visualize all Azure AD audit logs.)
- **Microsoft Defender for Cloud**
  - Query: `SecurityAlert | where AlertName contains "suspicious"`
    - (This query will list all alerts generated by Defender for Cloud where the alert name contains the keyword "`suspicious`".)

As you validate each connector, make sure to continue to explore other queries to obtain more precise information. While the goal of this chapter is not to dive into details about queries, you will need to continue expanding your KQL skills to use later in this book.

Chapter 3

Analytics

The power of Microsoft Sentinel comes from the ability to detect, investigate, and remediate threats. To do this, you must ingest data in the form of alerts from security providers, such as Microsoft solutions or third-party solutions. It can also be in the form of raw logs from services and endpoints that you need to monitor and want to turn into alerts.

Analytics in Microsoft Sentinel allow you to define detection rules across ingested data and create incidents for investigation by security analysts. Some of those rules might be simple and create an incident for an alert that comes from a connected solution. Others might be more complex and join data from various sources to determine whether a threat exists. For example, you might look for an unregistered DHCP server using a rule that looks for network traffic sent on UDP port 67 to an IP address that is not in a watchlist that contains DHCP-registered server IP addresses.

As you create analytic rules, it will be important to understand how many incidents each rule will generate in your environment. This will help prevent your analysts from becoming alert fatigued. In this chapter, you will learn about the components that make up an analytic rule, the types of analytics rules, how to create an analytic rule, and how to validate it.

Why use analytics for security?

In December 2021, the world was hit with what would become known as Log4j vulnerability. The vulnerability, which could allow even an unsophisticated attacker to take remote control over millions of endpoints using the Log4j package, was discovered by a researcher in China. Upon investigation, Microsoft released guidance to its MSRC blog, establishing Microsoft Sentinel queries customers could use to detect a potential Log4j attack. Also, Microsoft established a list of IP addresses as indicators of compromise (IOC).

> ***NOTE*** **To learn more about how Microsoft Security Researchers identified the indicators of compromise for Log4J, see *https://www.microsoft.com/security/blog/2021/12/11/guidance-for-preventing-detecting-and-hunting-for-cve-2021-44228-log4j-2-exploitation*.**

The use of analytics can be extremely beneficial for creating custom alerts that will trigger compromise indicators found in the systems. This is a powerful way to identify compromised systems without warning from other security controls (such as antimalware that relies on signatures). While this is considered a reactive work, you can also use analytics to identify whether a system is under attack because the system was already compromised. You can do this by creating alerts that use indicators of attack (IOA). By using analytics to create alerts based on an IOA, you can identify a potential attack in execution. For example, you can identify an attempt to elevate privileges to execute a built-in Windows tool, such as PowerShell, to download a piece of malware from a compromised site.

Also, the use of analytics can be useful to trigger alerts based on techniques used by known-malicious actors. For example, WannaCry used the `attrib` tool to perform file permission modification. You can create alerts based on custom queries that will trigger once the `attrib` technique is used.

TIP **You can use the MITRE ATT&CK website to learn more about the tools and techniques used by different kinds of malware. See *https://attack.mitre.org/software/S0366/.* MITRE ATT&CK is a globally accessible knowledge base of adversary tactics and techniques based on real-world observations. The ATT&CK knowledge base is used as a foundation for the development of specific threat models and methodologies in the private sector, government, and cybersecurity product and service community.**

Understanding analytic rules

In Microsoft Sentinel, the rules users create are called analytic rules. A rule is composed of several parts that define how the rule should trigger and how the incident should be handled. To access the Analytics dashboard, follow these steps:

1. Open the **Azure portal** and sign in as a user who has Microsoft Sentinel contributor privileges.
2. In the search pane, type **Microsoft Sentinel** and click the Microsoft Sentinel icon when it appears.
3. Select the workspace on which Microsoft Sentinel is enabled.

4. In the left navigation pane, click **Analytics**. The **Microsoft Sentinel | Analytics** blade appears, as shown in Figure 3-1.

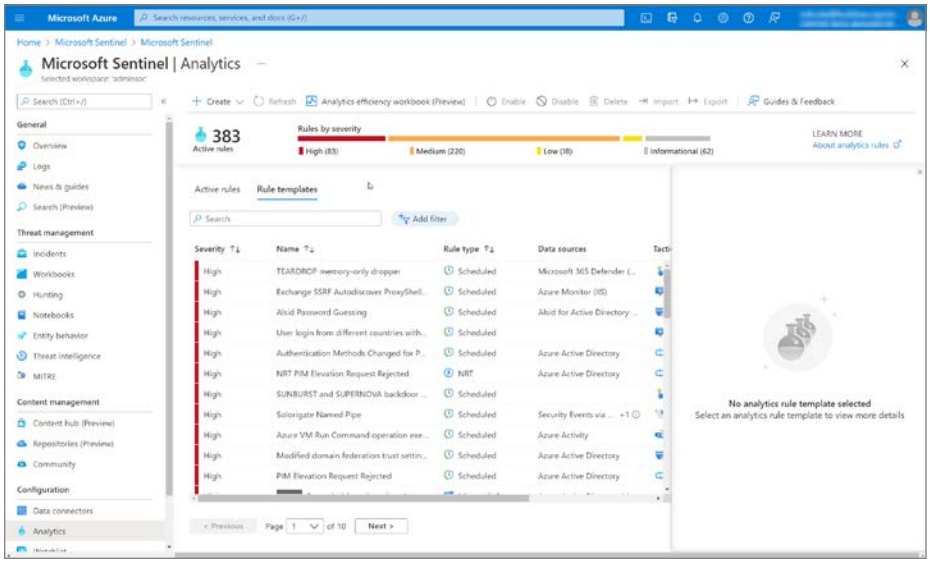


FIGURE 3-1 Microsoft Sentinel Analytics blade

5. There are several components in the **Analytics** blade. In the top pane, click **+Create** to create an analytic (see Figure 3-2).

FIGURE 3-2 Top pane of the Analytics blade

6. The middle pane shows the number of active analytic rules you have created or enabled. It also shows a breakdown of the analytic rules by severity (**High**, **Medium**, **Low**, and **Informational**). See Figure 3-3.

FIGURE 3-3 Middle pane of the Analytics blade

7. The bottom pane shows two tables. As you can see in Figure 3-4, one table shows the **Active Rules,** and the other shows **Rule Templates**.

FIGURE 3-4 The bottom pane of the Analytics blade

8. As you can see in Figure 3-5, the following information is shown:
   - The **Active Rules** tab shows rules that have been enabled or created in your Microsoft Sentinel workspace.
   - The table shows the name of each analytic rule and allows you to filter analytics by using the filter bar.
   - The **Name** column shows the rule name that was provided when the rule was created.
   - The **Rule Type** column shows the type of analytic—**Anomaly**, **Fusion**, **Microsoft Security**, **ML Behavior Analytics**, **Near Real-Time (NRT)**, **Scheduled**, or **Threat Intelligence**.
   - The **Status** column shows whether the analytic rule is **Enabled** or **Disabled**.
   - The **Tactics** column shows which MITRE tactics the rule helped detect.
   - The **Techniques** column shows which MITRE technique the rule is used to detect.
   - The **Last Modified** column shows the date and time the rule was last modified. You can search by any part of the rule name. You can filter the rules by **Severity**, **Rule Type**, **Status**, **Techniques**, and/or **Tactics**.

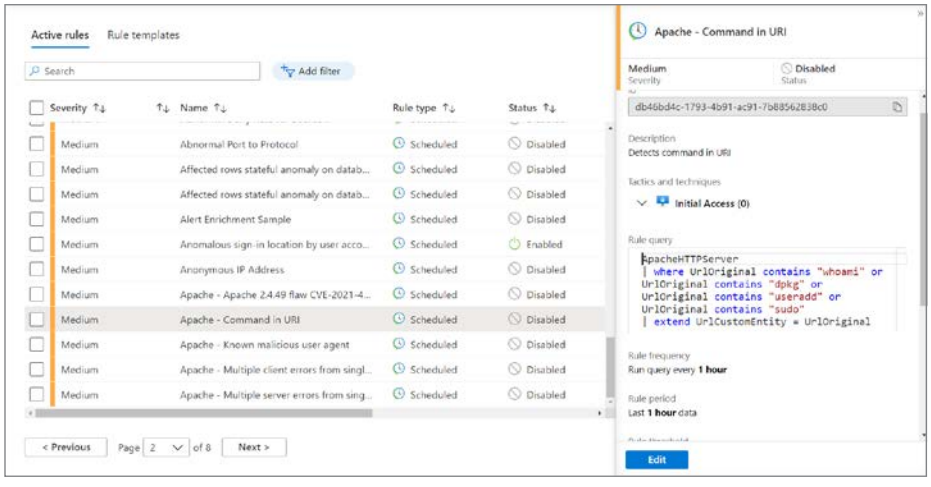


FIGURE 3-5 Bottom pane of the Analytics blade

9. The ellipsis column (**...**) provides a quick context menu that offers the following options: **Edit**, **Disable**, **Duplicate**, and **Delete**. Also, you can right-click the analytic to see the context menu, as shown in Figure 3-6.

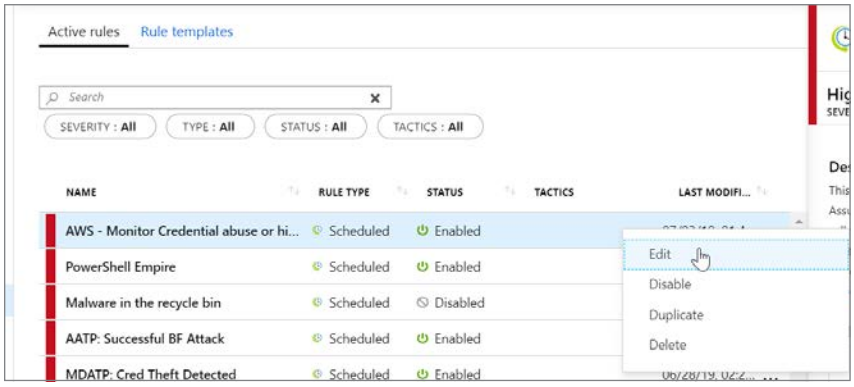


FIGURE 3-6 Context menu of the Analytics blade

10. Click the **Rule Templates** tab to see the list of templates available, as shown in Figure 3-7. The tab shows the available templates. Some of these templates are detections created by Microsoft, some are rules for Microsoft solutions, and some are community-based templates. We will cover the types of rule templates later in this chapter.
    - The **Name** column shows the rule name that was provided when the rule was created.
    - The **Rule Type** column shows the type of analytic: **Anomaly**, **Fusion**, **Microsoft Security**, **ML Behavior Analytics**, **Near Real-Time (NRT)**, **Scheduled**, or **Threat Intelligence**.
    - The **Required Data Sources** column shows which data sources are needed for the analytic rule.
    - The **Tactics** column shows which MITRE tactics the rule helped detect.
    - The **Techniques** column shows which MITRE technique the rule is used to detect.
    - You can search by any part of the rule name, and you can filter the rules by **Severity**, **Rule Type**, **Data Sources**, **Techniques**, and/or **Tactics**.

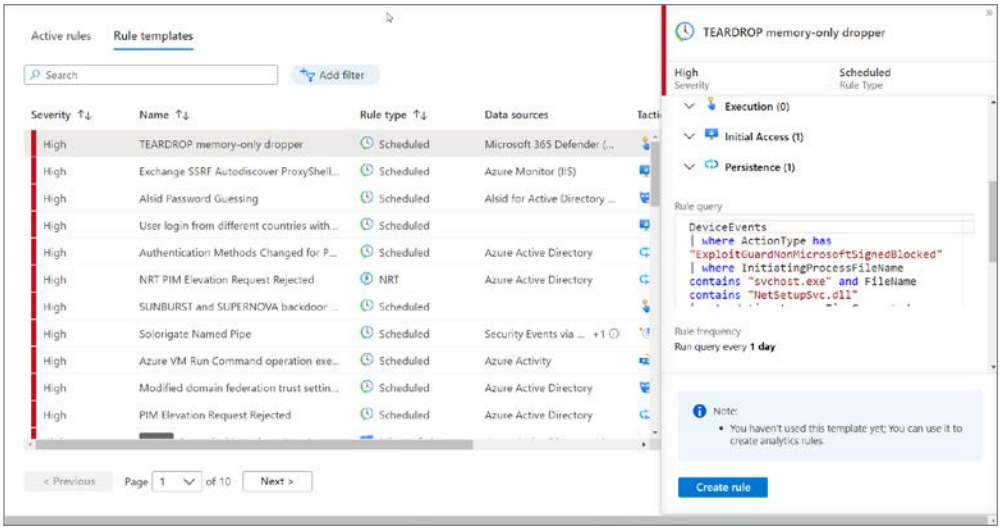


FIGURE 3-7 Bottom pane of the Analytics blade

Configuring analytic rules

If you are familiar with Microsoft Defender for Cloud, you know that the security alerts are built-in; in other words, you don't need to create rules in order to receive alerts. Microsoft Sentinel enables you to customize your own analytic rules based on your needs. These analytic rules will be the ones that will trigger alerts. Now that you are familiar with the Analytics blade, let's create your first analytic rule.

1. Open the **Azure portal** and sign in as a user who has Microsoft Sentinel contributor privileges.
2. In the search pane, type **Microsoft Sentinel** and click the Microsoft Sentinel icon when it appears.
3. Select the workspace on which Microsoft Sentinel has been enabled.
4. In the left navigation pane, click **Analytics.**
5. Click the **Create** button and select **Scheduled Query Rule**, as shown in Figure 3-8.

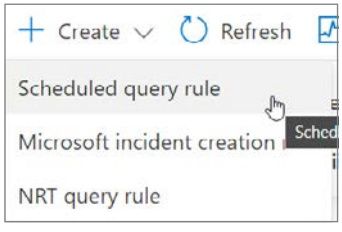


FIGURE 3-8 Create button

6. The first part of the rule creation wizard is the **General** section, as shown in Figure 3-9. The **Name** field is simply the name of the detection rule and the display name of the incident that will be generated if triggered. It is important to use a descriptive name that will allow your security analysts to understand what the alert is about. You can further describe what the incident is about by using the **Description** field to provide more detail for your security analysts. The **Tactics And Techniques** dropdown menu allows you to select the MITRE tactic(s)/technique(s) that the rule helps detect. The **Severity** dropdown offers four options: **High**, **Medium**, **Low**, and **Informational**. You can use this setting to override the alert severity for a connected data source that sends alerts; also, you can use this setting to set the severity for a created analytic. Severity should be used to help your security analysts prioritize and triage their responses and the incidents that are created. Lastly, you can set the **Status** to either **Enabled** or **Disabled**.

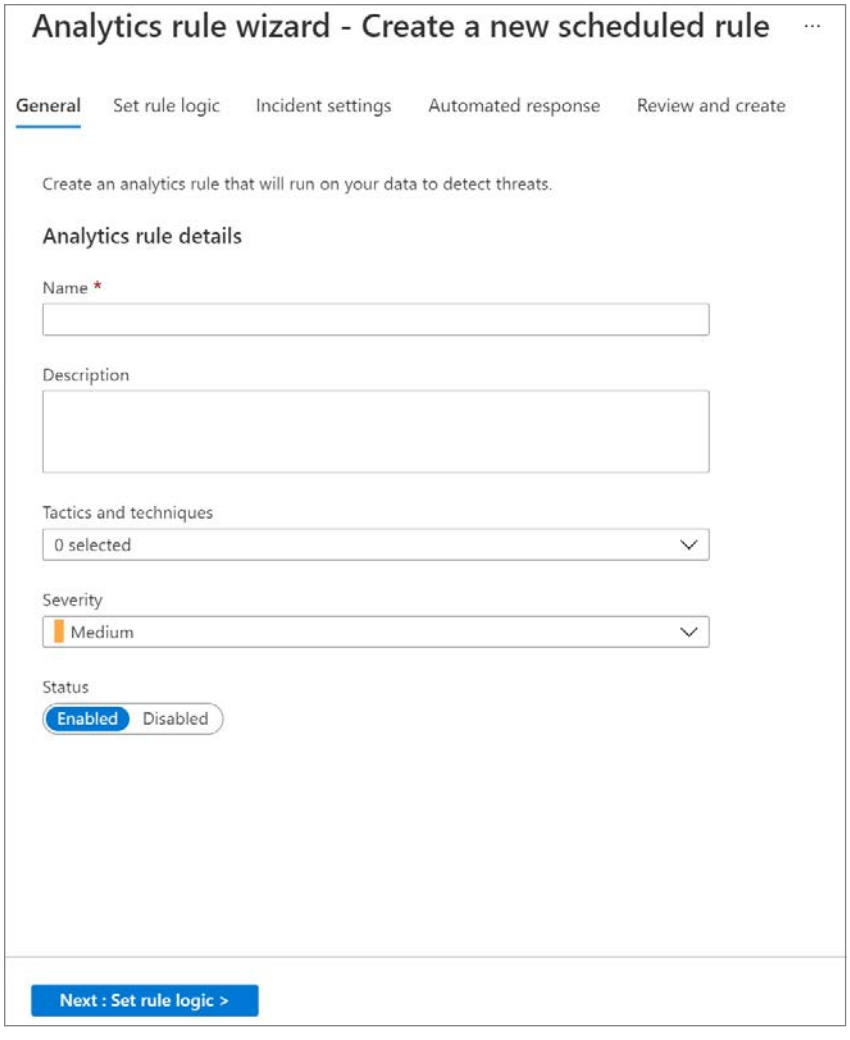


FIGURE 3-9 General section of Analytic Rule Wizard

7. The **Logic** section is shown in Figure 3-10. The **Rule Query** field is where you define what query you want to run against the Microsoft Sentinel workspace that will trigger and create an incident. Microsoft Sentinel stores the data in a Log Analytics workspace. To query the data in the workspace, you will use Kusto Query Language (KQL). Your query can be simple like `WireData | where RemotePortNumber == 443`, which will alert you when any computer connects outbound from port 443. For this sample query, you need to enable the **Azure Activity Data Connector** on the workspace. Your query can also be very complex, as shown in the example below:

```
AzureActivity
| where OperationName == "Create or Update Virtual Machine" or OperationName ==
"Create Deployment"
| where ActivityStatus == "Succeeded"
| make-series dcount(ResourceId) default=0 on EventSubmissionTimestamp in
range(ago(7d), now(), 1d) by Caller
```

The intent of this query is to trigger an alert when an anomalous number of resources is created in the Azure Activity Log.

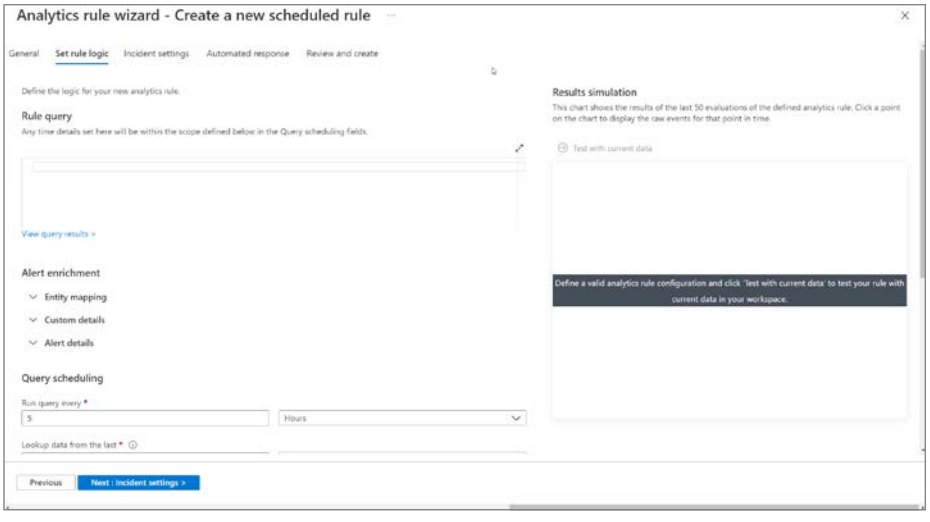


FIGURE 3-10 Logic section

TIP **For assistance with the query language, see the Query Language Reference at *https://docs.microsoft.com/en-us/azure/kusto/query/*.**

8. As you enter the query into the **Rule Query** box, the query window will validate the query you are writing in real-time. The **Results Simulation** on the right allows you to click **Test With Current Data**. When this is clicked, Microsoft Sentinel will simulate the query against your data and show how many alerts per day would be generated using the logic settings. This allows the analytics rule creator to test and adjust the rule before ever putting it into production. This can reduce the workload on the analysts and ensure they are alerted only on what matters. The **Alert Threshold** will show as a red line on the graphic. The blue line shows how many events would happen at a specific time. Each time the blue line passes above the red line, an incident or alert will be triggered, depending on the configuration of the rule. Figure 3-11 shows an example of the results line at the top in blue, and the threshold line at the bottom in red.

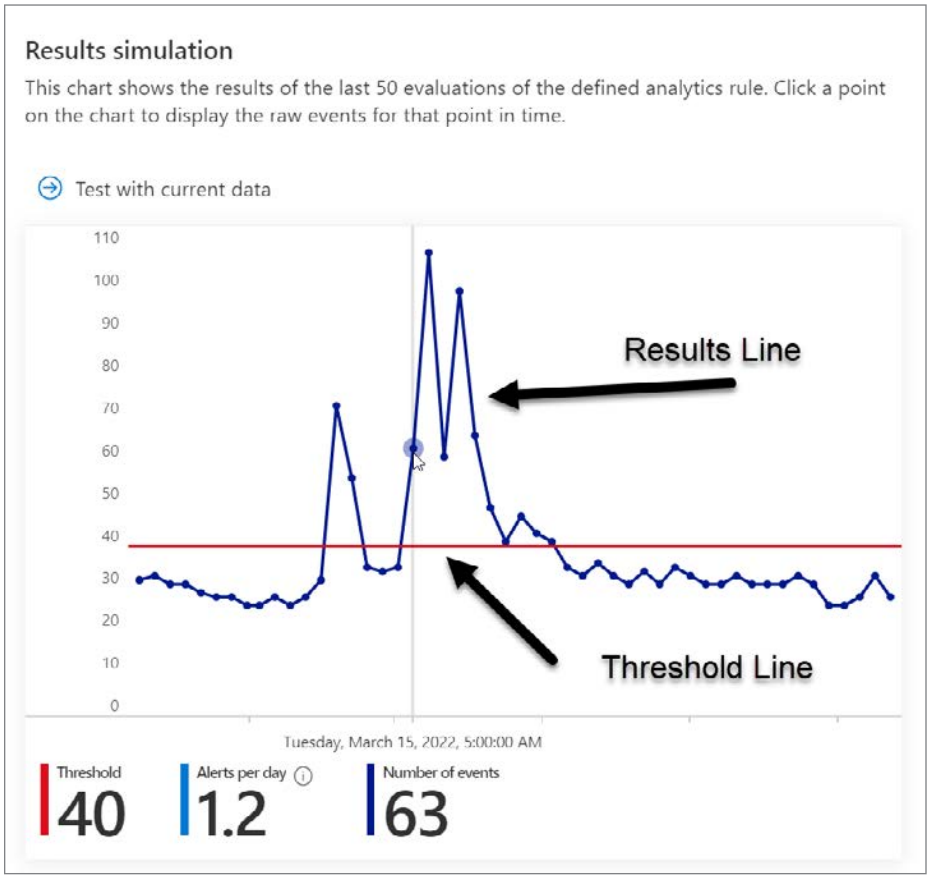


FIGURE 3-11 Alert simulation graphic

9. In the **Alert Enrichment** section, you can define the entities that are returned as part of the data that was queried in the **Rule Query Using Entity Mapping**. Entities are important because they allow you to select which field from the data returned represents a user, host, IP address, or other entity type. This information might be different column names across data sets, and the mapping allows you to normalize the data into entities. Entities are very important for incidents and investigation, which will be covered later in this book. Figure 3-12 shows the **Entity Mapping** section.

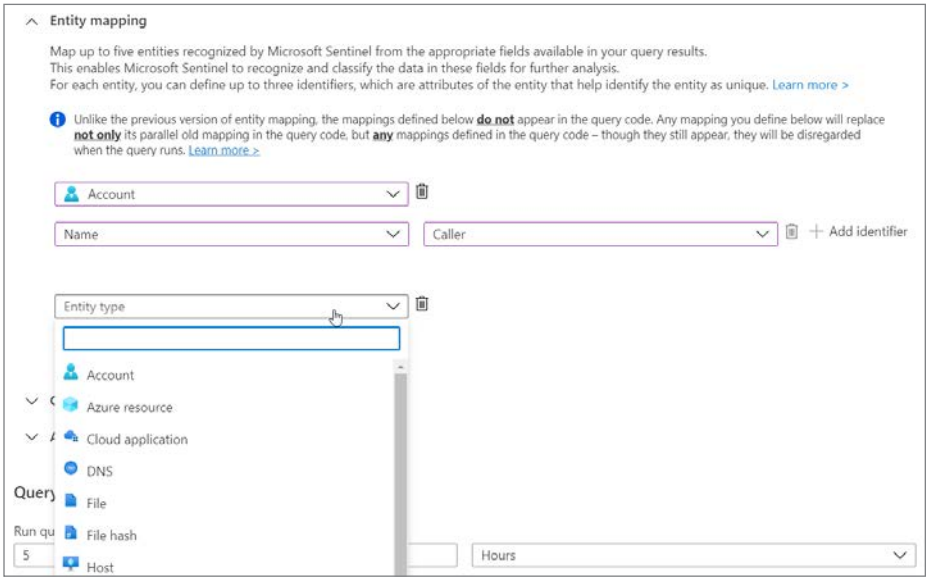


FIGURE 3-12 Entity Mapping section

Not all alert rules will have all entity types. For example, an alert rule based on firewall log data might only contain IP address entities. Mapping more entities when creating a rule will be useful when responding to incidents. Doing so will help the analysts understand which user or computer was involved or which IP address was used by the source machine.

In **Custom Details**, you can surface event data as part of the alert properties. You can map event fields to alert properties using key–value pairs. Figure 3-13 shows the **Custom Details** section.

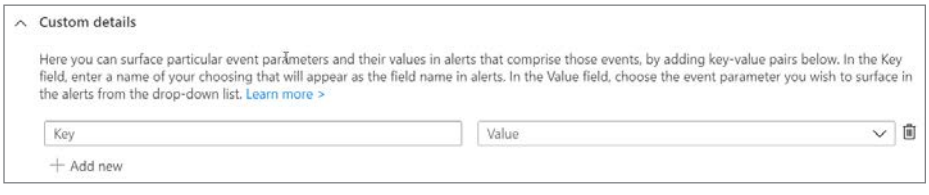


FIGURE 3-13 Custom Details section

In the **Alert Details** section, you can override or populate additional context in the alert rule's general properties. For example, you could inject the **Provider Name** into the alert/incident name. Figure 3-14 shows the **Alert Details** section.

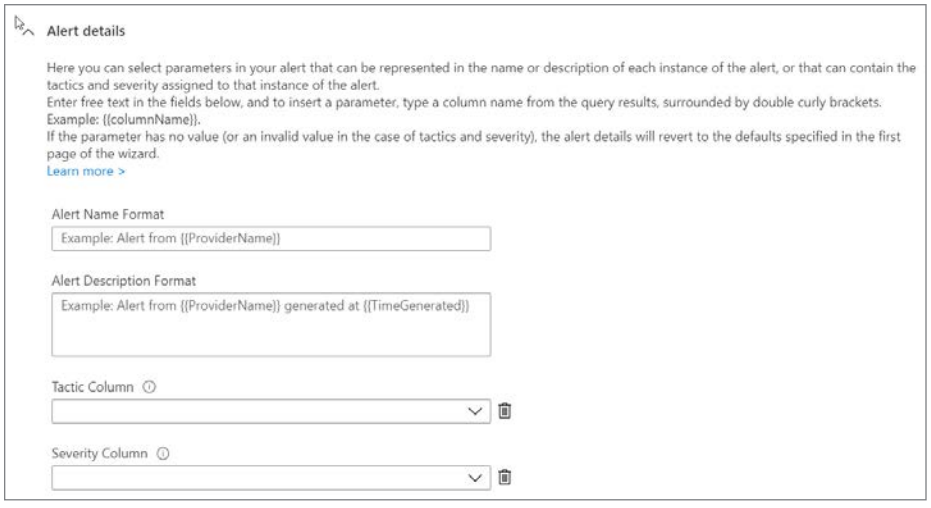


FIGURE 3-14 Alert Details section

10. The **Query Scheduling** section is where you set how often to trigger the query and how far back in the data to query against. The **Run Query Every** field defines how often you want to evaluate the query against your data. You might have a rule that runs every 5 minutes or once every 24 hours. Both options can be between 5 minutes and 14 days. The **Query Scheduling** section is shown in Figure 3-15.

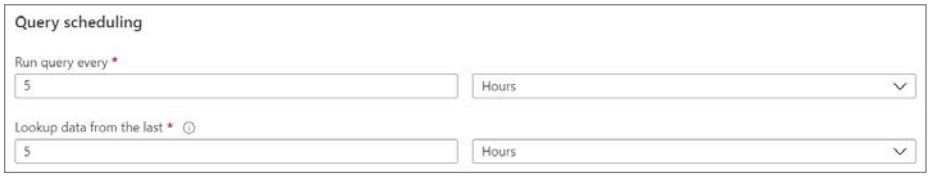


FIGURE 3-15 Query scheduling section

11. The **Alert Threshold** section is where you set the number of results required for the rule to generate an incident. The **Generate Alert When The Number Of Query Results** trigger supports the following operators: **Is Greater Than**, **Is Fewer Than**, **Is Equal To**, or **Is Not Equal To**. Then you define the number for the threshold. Figure 3-16 shows the **Alert Threshold** section.

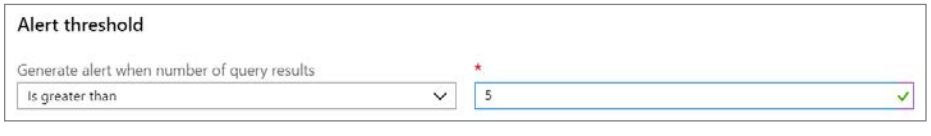


FIGURE 3-16 Alert threshold section

12. To tie the previous two steps together, you might want to trigger an alert for a user who exceeds 5 failed logins in a 15-minute window. You would configure the **Run Query Every** setting to 5 minutes, and then you would set the **Lookup Data From The Last** setting to 15 minutes. Lastly, set the **Generate Alert When Number Of Query Results Is Greater Than** to 5. This would run the query rule every 5 minutes and look at the last 15 minutes of data. If the failed logins crossed 5, it would generate an incident.
13. The **Event Grouping** section is where you can configure if you want to group all events into one alert or separate each event into its own alert. Figure 3-17 shows the **Event Grouping** section.

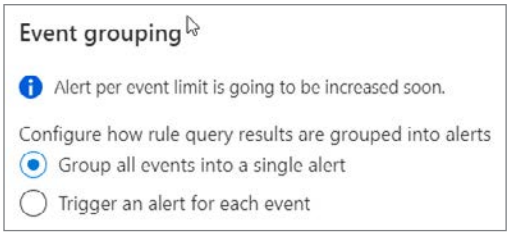


FIGURE 3-17 Event grouping section

14. The **Alert Suppression** section allows you to set alert suppression to **On** or **Off**. This option allows some basic suppression of the rule to prevent creating additional incidents if the rule is triggered when you want it to be suppressed. If you select **On**, the **Stop Running Query For** field appears. You can set this to anywhere between 5 minutes and 24 hours. Figure 3-18 shows the **Suppression** setting.

FIGURE 3-18 Suppression setting

15. The **Incident Settings** section allows you to configure the incident settings for the alert. First, you can choose whether to create an incident for each alert triggered by the rule. In most cases, you would create an incident for each alert, but there are cases where you might have two rules that create alerts and another rule that looks for both alerts created within a timespan to trigger an incident. These can be thought of as alert correlation rules. In the Alert Grouping section, you can group related alerts into the same incident. If disabled, each alert will be its own incident. If enabled, you can then choose the time limit for grouping, the grouping logic, and the re-opening incident options. In most cases, it is recommended to use the **Grouping Alerts Into A Single Incident If All The Entities Match** option, as shown in Figure 3-19.

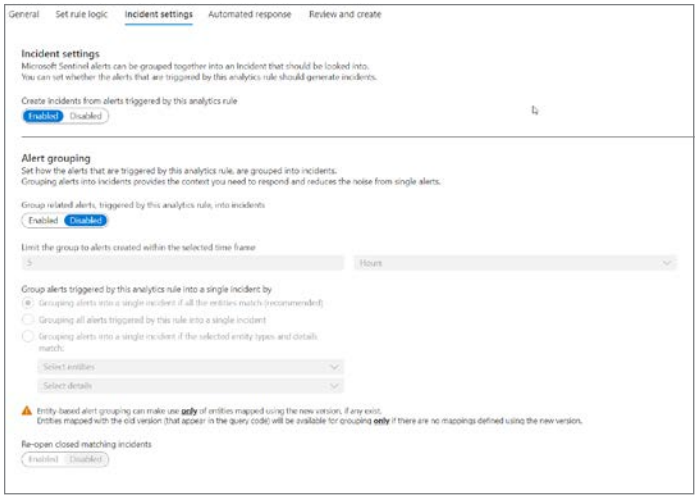


FIGURE 3-19 Incident settings section

16. The **Automated Response** section allows you to select a Playbook for alert automation or an automation rule for incident automation. Automation rules will be covered in Chapter 7, "Automating response." This allows you to automate the response to alerts or incidents. This automation could be to run a query to gather more data to enrich an incident, automatically respond by disabling an account, or even open a ticket in a third-party ticketing system. For a Playbook to be listed in the Alert Automation section, it must use the **Microsoft Sentinel Alert** trigger. Triggers will be explained in Chapter 7, "Automating response." Figure 3-20 shows the **Automated Response** section.

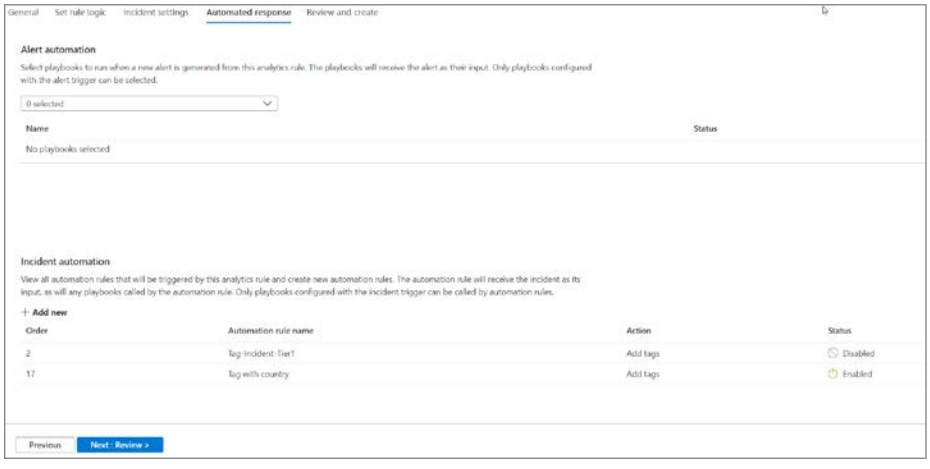


FIGURE 3-20 Automated response section

Although this book will not cover Azure Logic Apps in-depth, Chapter 7, "Automating response," will cover more details about how to create a Playbook.

NOTE See more about Azure Logic Apps at *https://docs.microsoft.com/en-us/azure/logic-apps/logic-apps-overview.*

17. The **Review And Create** section, as shown in Figure 3-21, allows you to review the settings you have configured in the wizard before creating new rules.

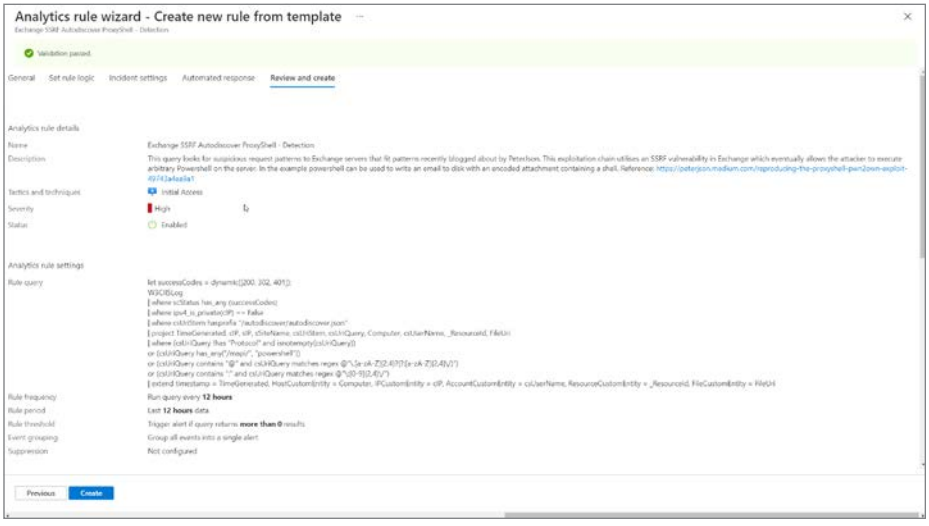


FIGURE 3-21 The Review And Create tab

Types of analytic rules

There are seven types of analytic rules built into Microsoft Sentinel: Anomaly, Fusion, ML Behavior Analytics, Microsoft Security, NRT, Scheduled, and Threat Intelligence. The following sections go into more detail on each.

Anomaly

Anomaly rules are built-in rules that use machine learning (ML) against your data to detect specific types of threats. These rules have a few parameters and thresholds that can be configured; the ML models are protected by Microsoft. You can enable these rules in Flighting mode first to see how they would perform against your data, and when you're ready, you can move them to Production.

Fusion

Microsoft has created a scalable ML correlation engine. This rule has no configurable settings. It can only be enabled or disabled in the workspace. The fusion rule will correlate low-fidelity alerts across alert rules into a single high-fidelity incident if the alerts are part of the same attack-kill chain.

Machine learning behavioral

These rule templates are based on Microsoft proprietary ML algorithms. Machine learning (ML) is applied to specific data sources and events to detect anomalous behavior in the data.

Microsoft security

In Microsoft Sentinel, analytic rules for Microsoft solutions are easy to create. This allows you to create an incident in Microsoft Sentinel from any existing security alert that comes from these solutions. You will not need to create individual analytic rules for Microsoft solution alerts.

These rule templates will create an incident whenever an alert is generated by the source Microsoft solutions. When you click **Create Rule**, you can filter by severity and/or text in the alert name. For example, this will allow you to only create incidents for high-severity alerts from Microsoft Defender for Cloud. Or, you might choose to create an incident if the alert contains a "pass" from Microsoft Defender for Identities. Figure 3-22 shows the table of built-in Microsoft rules.

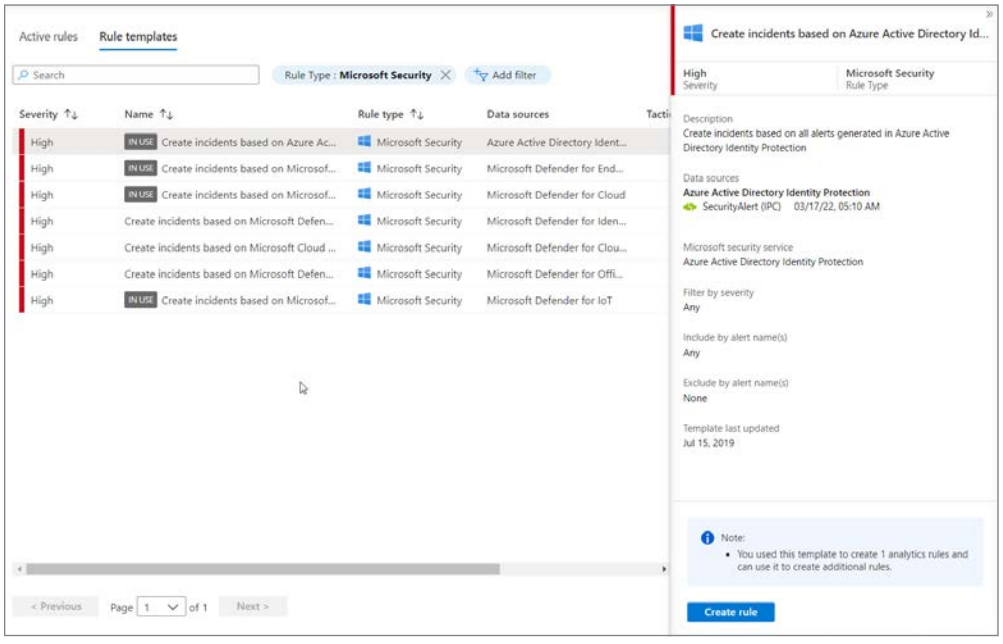


FIGURE 3-22 Microsoft Security rules

Near-real-time

NRT rules allow you to run detections up to the minute. Because scheduled rules only allow a query to run as low once every 5 minutes, NRT rules are hardcoded to run every minute on the last minute of data. Currently, these rules are limited to 20 per workspace. NRT rules can be useful for real-time alerting during an incident or very privileged detections that need immediate detection.

Scheduled

Scheduled analytics are your typical SIEM rule that runs on a timed basis and looks at certain periods of data. These rules trigger on a configured threshold.

Threat intelligence

The threat intelligence (TI) rule matches Microsoft-provided TI data against Common Event Format (CEF), DNS, and Syslog data. You don't need to write scheduled rules to match TI against these data sources. Simply enabling this rule will do all the matching for you.

Community

In the Microsoft Sentinel Community, Microsoft contributes sample rules created by various Microsoft Security teams. Customers can contribute sample rules as well. Typically, these rules are additional detections that are built on data sets, such as Windows Events, that are not already part of a Microsoft Security solution. Microsoft Sentinel will automatically sync the GitHub community detections that Microsoft has chosen, which will allow you to enable the rule and apply it to your environment.

Creating analytic rules

Now that you know all the components of an analytic rule, let's create one and see how this analytic will trigger an incident. Follow these steps to configure your first useful analytic rule in Microsoft Sentinel.

1. Open the Azure portal and sign in as a user who has Microsoft Sentinel Contributor privileges.
2. In the search pane, type **Microsoft Sentinel**, and click the Microsoft Sentinel icon when it appears.
3. Select the workspace on which Microsoft Sentinel is enabled.
4. In the left navigation pane, click **Analytics**.
5. Click **Create**, and then click **Scheduled Query Rule** in the top pane, as shown in Figure 3-23.

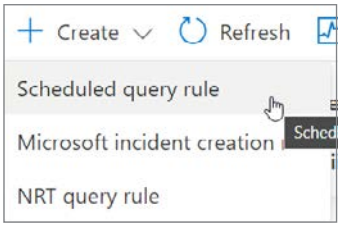


FIGURE 3-23 Scheduled query rule

6. Enter these settings and see Figure 3-24 for an example:
   - In the **General** section, enter **Azure VM Deletion** for the **Name**.
   - In the **Description** field, enter **A Simple Detection To Alert When Someone Deletes An Azure Virtual Machine**.
   - Set the **Tactic** to **Impact** and the **Technique** to **T1485–Data Destruction** and **T0826–Loss Of Availability**.
   - Set the **Severity** to **Informational**.
   - Leave **Status** as **Enabled**.
   - Click the **Next: Set Rule Logic** button.

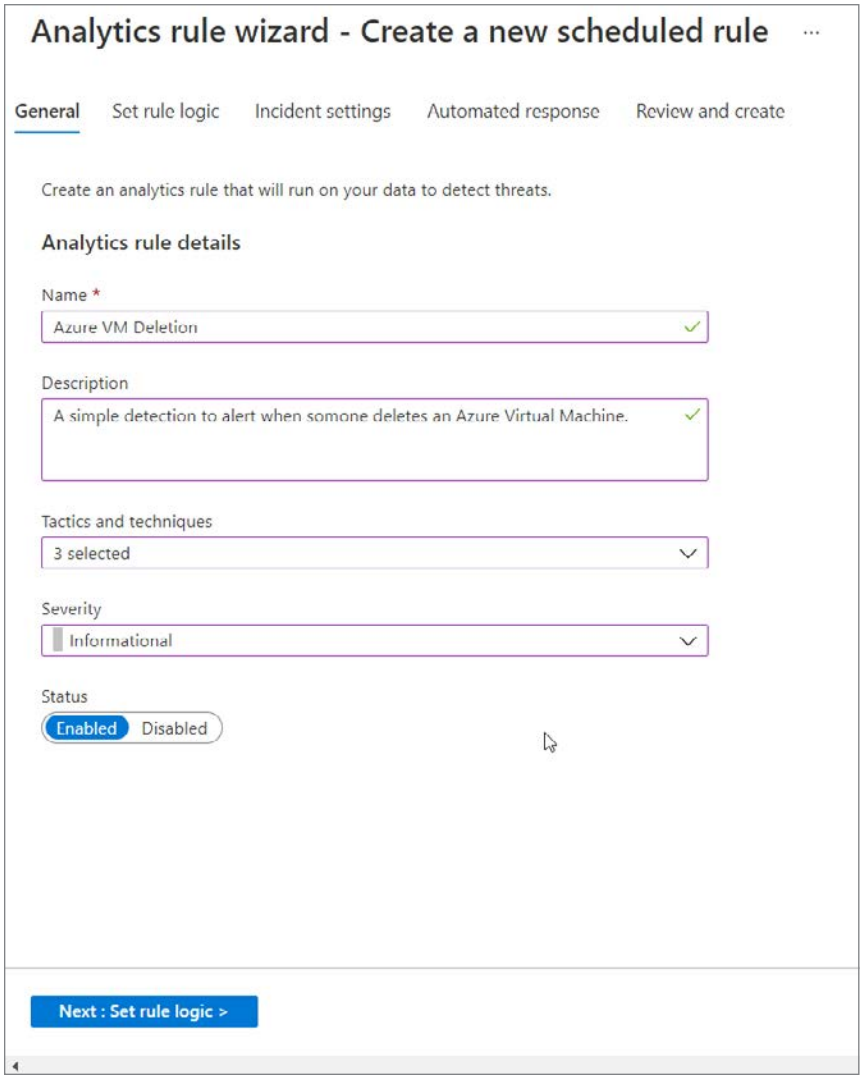


FIGURE 3-24 The General section of the Analytic Rule Creation wizard

7. Enter the following query in the Rule Query box.

```
AzureActivity
| where OperationName == "Delete Virtual Machine"
| where ActivityStatus == "Accepted"
```

8. In the **Entity Mapping** section, follow these steps (and see Figure 3-25 for an example).
   - Click the **Entity Type** dropdown menu and select Account.
   - Select AADUserId from the **Identifier** dropdown.
   - Click the Value box and notice that the **Value** dropdown menu enumerates all columns returned from your query, which eases the selection of columns representing each entity.
   - Select **Caller**.
   - Click **Add New Entity**.
   - Set the **IP Entity** to **Address = CallerIpAddress**.
   - Add another entity and set the **Host AzureID = ResourceId**.

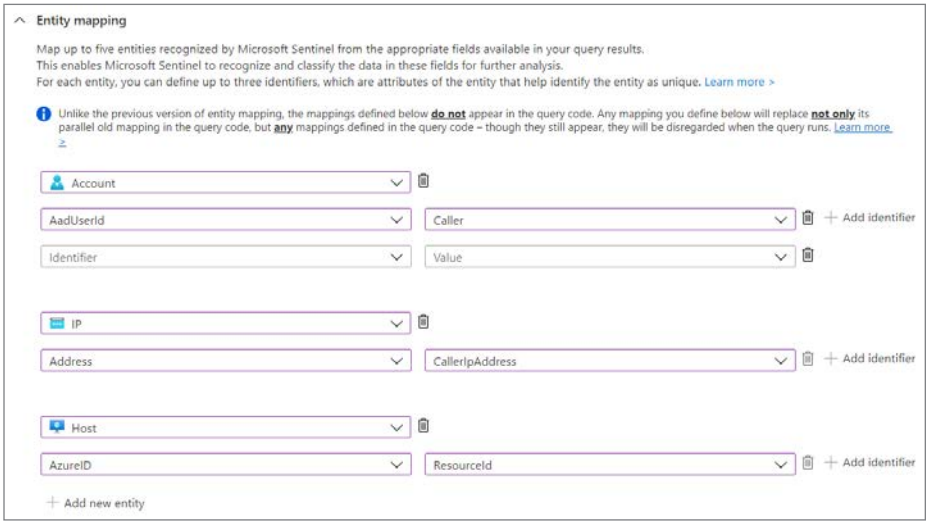


FIGURE 3-25 Entity Mapping section

9. In the **Query Scheduling** section, enter **5** in the **Run Query Every** field and select **Minutes**. Enter **5** for the **Lookup Data From The Last** and select **Minutes**.
10. In the **Alert threshold** section, enter **Is Greater Than 0** for the **Threshold**. Leave the other default settings.
11. Click **Next: Incident settings**. Leave the default settings to generate an incident each time this rule triggers.

12. **Click Next: Automated Response.** In the **Automated Response** section of the wizard, click **Next: Review**. We will not assign a Playbook or automation rule at this time.
13. Figure 3-26 shows the example analytic rule review page. Click **Save**.

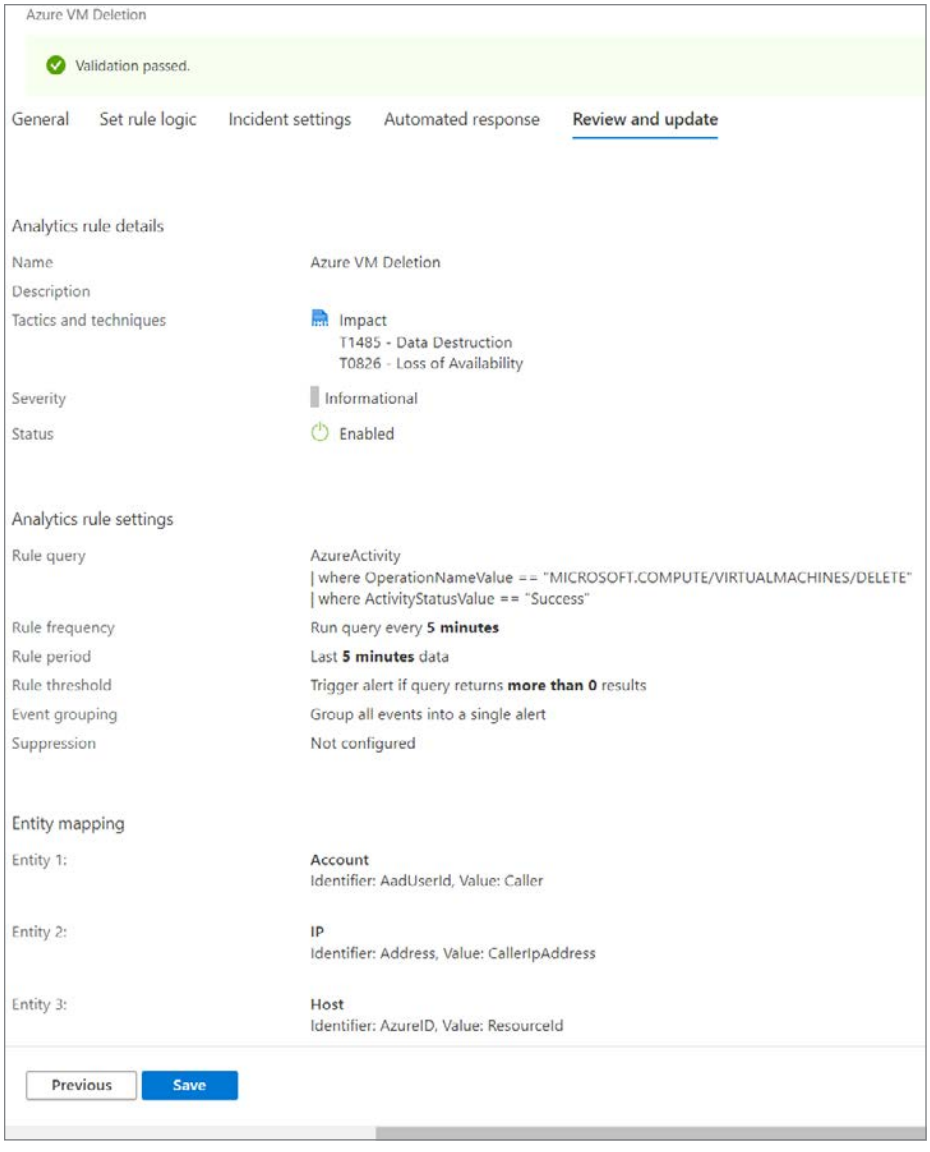


FIGURE 3-26 The Review And Update tab

Once you are back in the **Analytics** blade, you see the analytic you just created (see Figure 3-27).

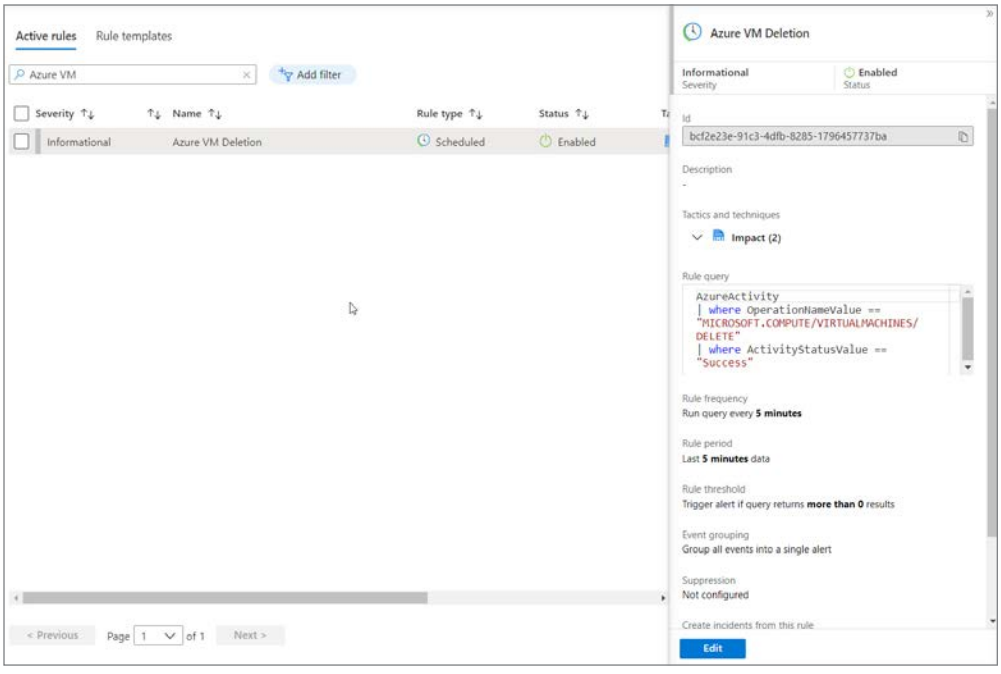


FIGURE 3-27 The Analytics blade in Microsoft Sentinel

Validating analytic rules

Now that you have created your first analytic, let's walk through validating it. Follow these steps:

1. Open the **Azure portal** and sign in as a user who has Microsoft Sentinel Contributor privileges.
2. In the search pane, type **Resource Groups** and click the icon when it appears.
3. Click the resource group you created in Chapter 2.
4. Select the desired virtual machine.
5. Click **Delete** in the top bar.
6. In the **Delete Resources** blade, type **yes** to confirm the deletion.
7. Click **Delete**.
8. It will take some time for the analytic to trigger because Azure Activity must first write the logs

NOTE To learn more about log data ingestion time, see *https://docs.microsoft.com/en-us/azure/azure-monitor/platform/data-ingestion-time#azure-activity-logs-diagnostic-logs-and-metrics*.

9. In the search pane, type **Microsoft Sentinel** and click the Microsoft Sentinel icon when it appears.
10. Select the workspace on which Microsoft Sentinel is enabled.
11. Click **Incidents**. You will see that an incident has been created, as shown in Figure 3-28.

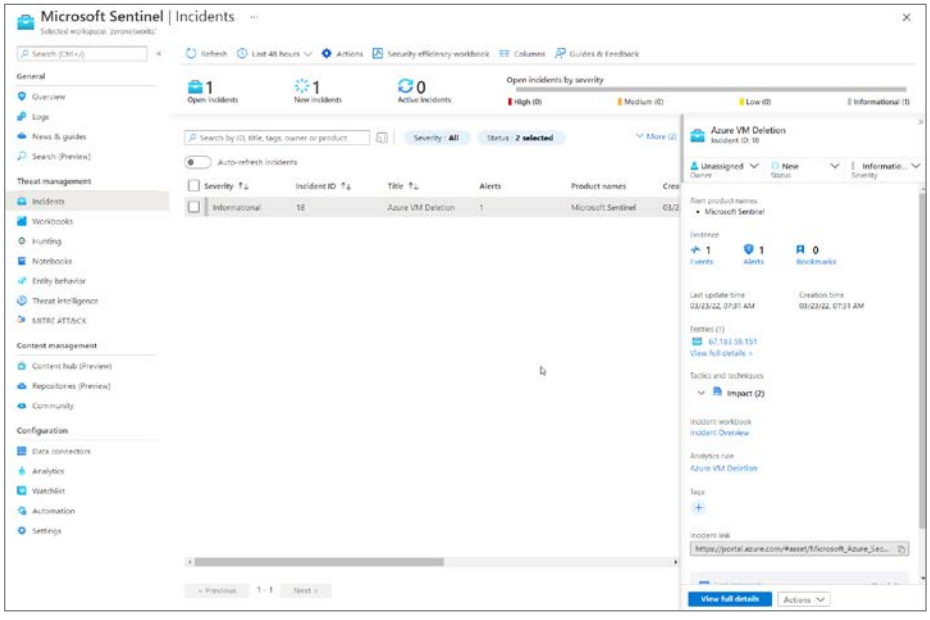


FIGURE 3-28 The Incident blade in Microsoft Sentinel

NOTE Incidents are covered in more depth in Chapter 4, "Incident management."

Chapter 4

Incident management

Security Incident Management describes the process of how incidents are detected, reported, assessed, and responded to. This chapter focuses on how Microsoft Sentinel empowers the SOC analyst to efficiently triage a security incident. The triage process greatly benefits from features like threat intelligence, Fusion, machine learning, and automation capabilities, which all contribute to the determination of whether the SOC analyst is looking at a true or false positive. In addition, User and Entity Behavior Analytics (UEBA) and other components like Watchlists can add valuable context to an incident. Chapter 3, "Analytics," described how rules are authored to create incidents, and Chapter 5, "Hunting," describes how the escalation of an incident continues when the hunting team takes over. This chapter will cover the triage and incident management process.

Understanding Microsoft Sentinel incidents

Microsoft Sentinel's security incident definition is based on National Institute of Standards and Technology (NIST) SP 800-12 Rev. 1, FIPS 200 and is described as follows:

> *An occurrence that actually or potentially jeopardizes the confidentiality, integrity, or availability of an information system or the information the system processes, stores, or transmits or that constitutes a violation or imminent threat of violation of security policies, security procedures, or acceptable use policies.*

Data ingested into Microsoft Sentinel can be generally divided into the following categories:

- **Raw and semi-unstructured data** Also known as *custom logs* in Microsoft Sentinel's terminology. Although the data is considered unstructured, a schema is generated automatically upon ingestion but is not validated or enforced (hence, "semi-unstructured data").
- **Normalized logs and events** Follows a strict schema of field names and values. Notable events are part of this type of data.

- **Alerts** As defined by the sending source, alerts are normalized in nature, and schemas are validated and enforced.
- **Incidents** As defined by the sending source, incidents are normalized in nature, and schemas are validated and enforced. Technically, incidents are created in Microsoft Sentinel upon ingestion.

An incident in Microsoft Sentinel is created in one of the following ways:

- A scheduled analytics rule (as described in Chapter 3), based on either one or more alerts or notable events
- An incident created by one of the data connectors based on one or more alerts
- A Fusion incident based on fused and correlated events
- An incident based on a Machine Learning (ML) behavior analytics rule
- An incident based on a match with one or more threat indicators
- An incident based on a hunting query
- A manually created incident through the Security.Insights API, PowerShell, or the Azure portal

What all these incidents have in common is that a notable event has been created as a data source. This can be through an agent, Azure resource or service, AWS, GCP event, or any ingested data stream. A notable event converts to an alert if certain (rules) conditions are matched, which will result in the creation of an incident, which by itself can contain one or more alerts. The exception is that the source can also send alerts or incidents directly. This is most common with first-party data connectors, which cover Microsoft sources like Azure Active Directory Identity Protection, Azure Information Protection, and so on. Another example would be incidents sent through the **Microsoft 365 Defender** connector (M365D), which contains alerts as well.

The benefit is that alerts and incidents created by these data connectors already contain correlated entities, such as a host, account, IP address, and the like. Scheduled analytics rules are flexible in the sense that you can define which entities should be associated with the incident. In Sentinel, this concept is called *entity mapping*.

Exploring and configuring the Incidents view

The incidents blade, which shows all incidents, can be customized to fit the needs of the analyst. A time and date range can be configured so that it is retained for the duration of the analyst session, even if the analyst navigates to another Sentinel blade. Figure 4-1 shows the **Last 24 Hours** default selection, which can be customized using a **Custom Range**.

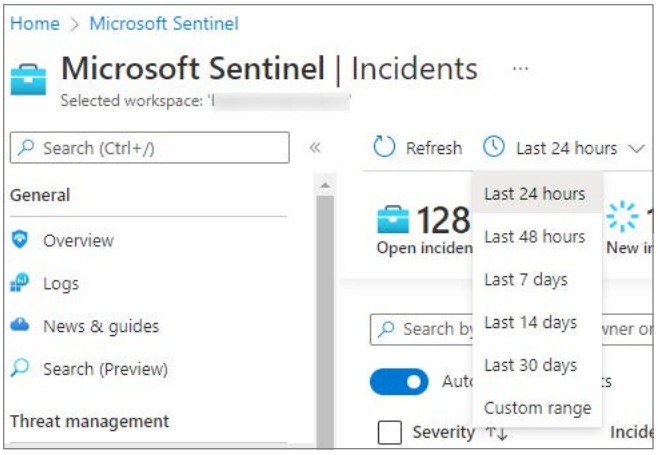


FIGURE 4-1 Selecting the time and date range for viewing incidents

To select the time and day range of incidents to view, follow these steps:

1. Open the **Azure portal** and sign in as a user who has Microsoft Sentinel Reader privileges.
2. Under **Threat Management**, select **Incidents.**
3. Collapse the arrow next to **Last 24 Hours** and make your selection based on the default choices, or select **Custom Range** to select your **From (UTC)** and **To (UTC)**, as shown in Figure 4-2.

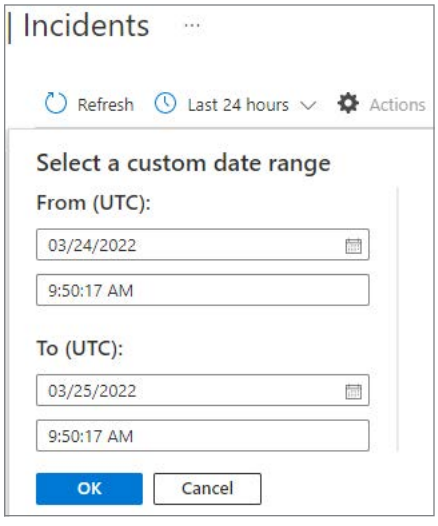


FIGURE 4-2 Selecting a custom range for viewing incidents

NOTE **Custom date ranges will always be shown in UTC.**

Your selection will update the **Open Incidents**, **New Incidents,** and **Active Incidents** count to reflect the time and day range. The filters for **Severity, Status, Product Name**, and **Owner** allow you to further filter incidents according to your needs. Figure 4-3 shows incidents being filtered by **Product Name**.

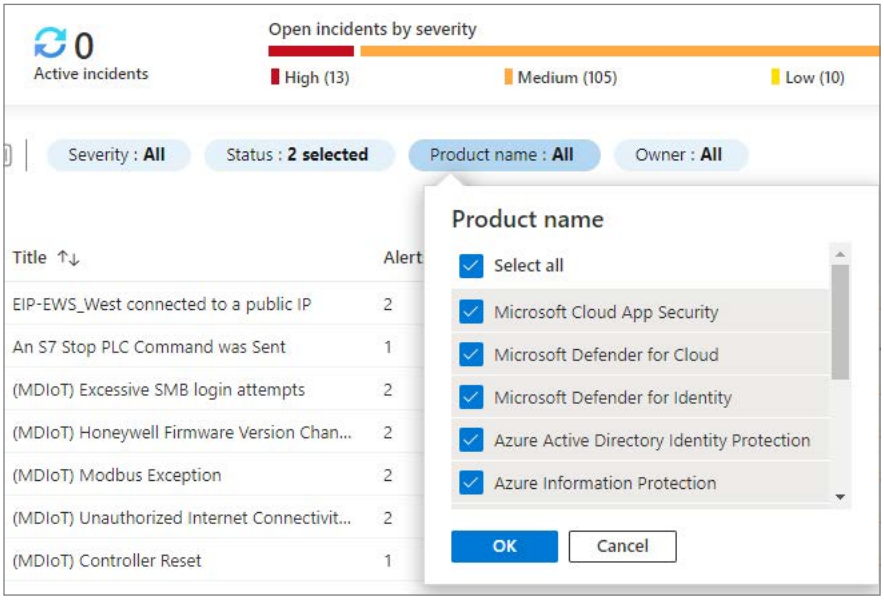


FIGURE 4-3 Filtering the incidents view by Product Name

In the top-middle area of the incidents view (next to the time and day range filter) are the **Actions** that can be taken. You can use the checkboxes to select multiple incidents. When multiple incidents are selected, clicking the **Actions** button enables you to change the **Severity**, assign an owner or group, change the status, or add Tags for multiple incidents at once.

As shown in Figure 4-4, next to the **Actions** button is the **Security Efficiency Workbook** option, which helps you monitor your SOC Key Performance Indicators (KPI), such as the mean-time to triage, meantime to closure, and so on.

FIGURE 4-4 Options available for incidents

Follow the steps below to access the **Security Efficiency Workbook**:

1. While still on the **Incidents** blade, select the **Security Efficiency Workbook** button.
2. Select your **Subscription** and **Workspace.**
3. Filter according to your criteria, such as by owner, tactics, or product name.
4. The Workbook shows different KPIs, such as **Time To Triage** and **Time To Closure**, which can be found if you scroll down through the Workbook (see Figure 4-5).

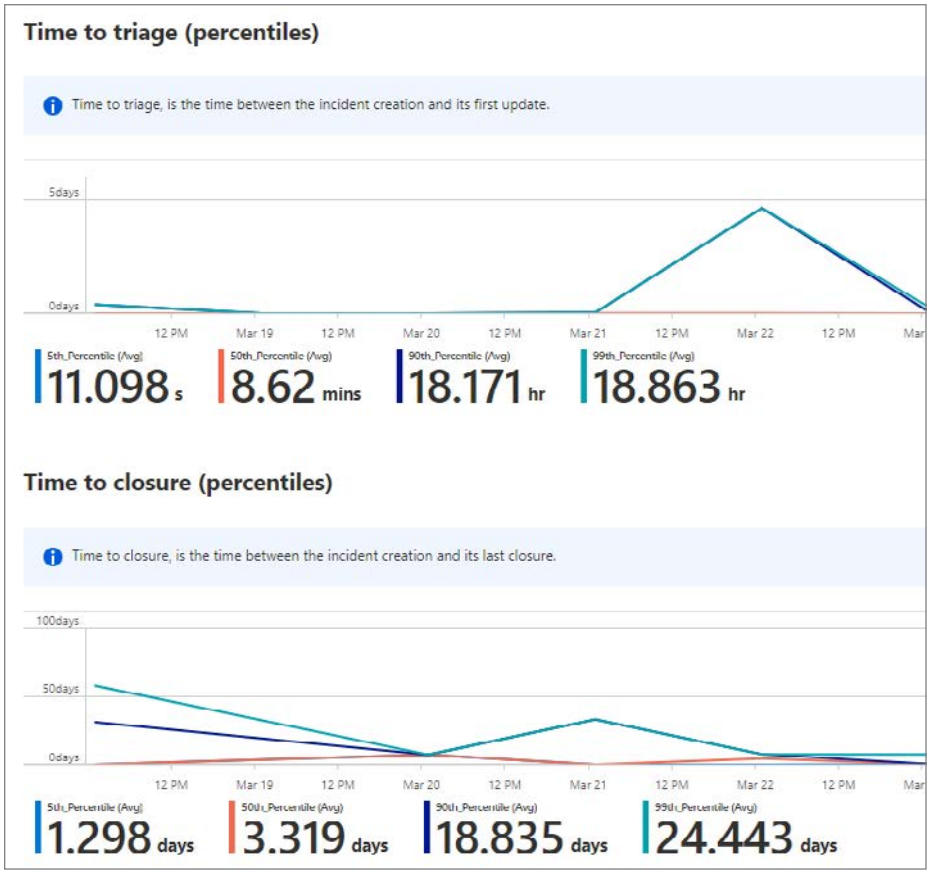


FIGURE 4-5 The Security Efficiency Workbook, showing the Time To Triage and Time To Closure

Back in the **Incidents Overview** blade, you will see the **Columns** option to the right of the **Security Efficiency Workbook** option. After clicking **Columns**, you can unhide or hide certain columns, and you can change the order. Figure 4-6 shows an example of unhiding the **Tactics** column and reordering it so it appears next to the **Incident ID** column.

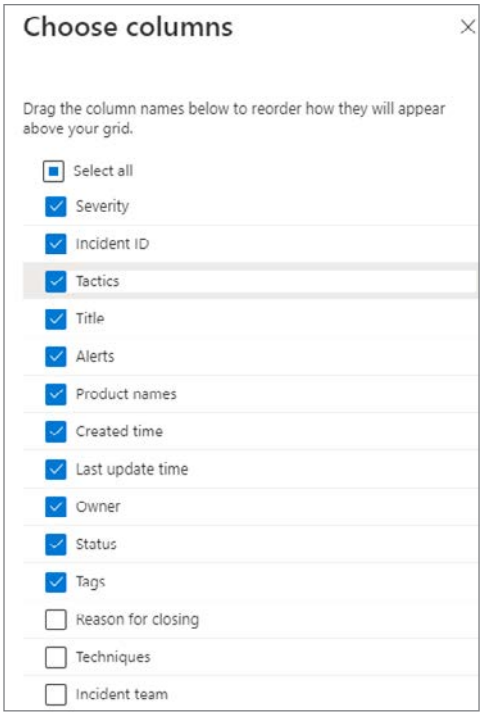


FIGURE 4-6 Choosing to hide, unhide, or reorder columns

After you click **Apply**, the incident column view will be updated based on your selection, as shown in Figure 4-7.

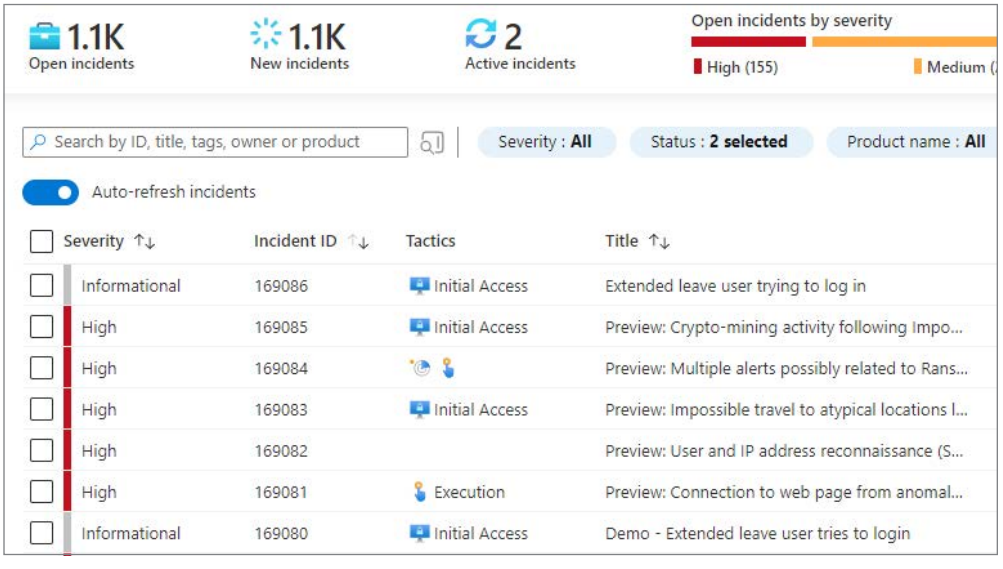


FIGURE 4-7 Tactics column added and reordered to appear next to the Incident ID column

Guides and feedback

The **Incidents** blade also offers you an opportunity to provide valuable feedback to the Sentinel Product Group, which is always considered when developing new features and improving the product. Here, you will find guidance on how incidents work in Sentinel, valuable links to explore, and the Sentinel community forum to share your ideas and suggestions. Figure 4-8 shows the **Guides & Feedback** pane with **Useful Links**, the **Vote Or Add Your Ideas** link, and a **Tell Us About Your Experience** text box, where you can share your experience with Sentinel.

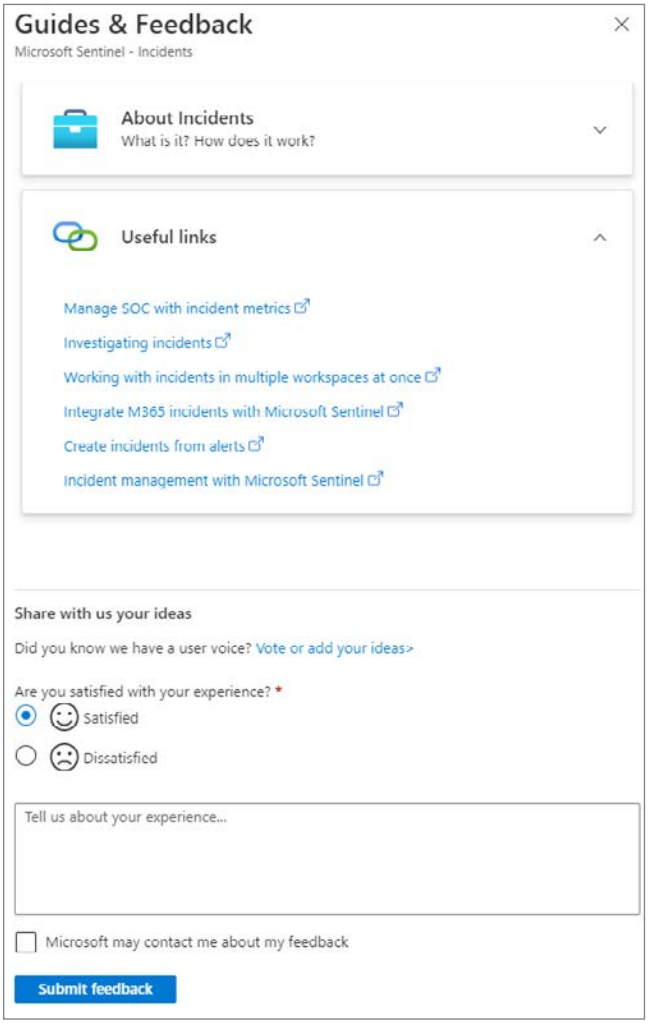


FIGURE 4-8 The Guides & Feedback pane

You can provide feedback or explore useful links by clicking **Guides & Feeback** in the upper middle part of the **Incidents Overview** pane. This is also the place to provide feedback or **Share With Us Your Ideas.**

Triaging incidents

The function of a Tier 1 analyst is to quickly determine if an incident is a true or a false positive, followed by resolving (dismissing) the incident or escalating it to a higher tier. Because of the rapid increase of signals and data streams, this must be a quick and efficient process to prevent analyst fatigue. The SOC analyst will start triaging incidents in the **Incidents** blade, as shown in Figure 4-9.

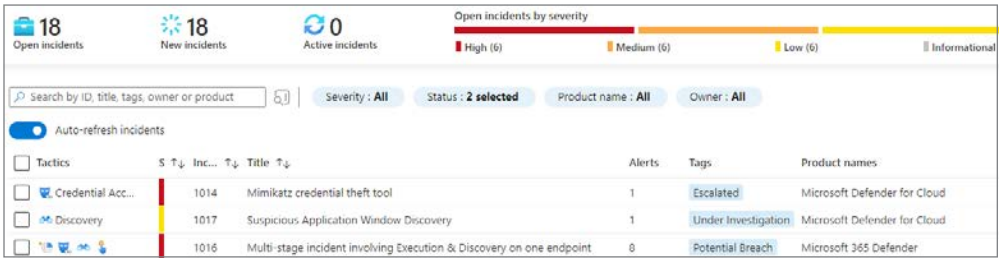


FIGURE 4-9 Incidents view with basic analyst triage information

To view incidents, open the **Azure portal** and sign in as a user who has Microsoft Sentinel reader privileges. Under **Threat Management**, select **Incidents**. Select an incident by clicking it.

Typically, an incident can be prioritized for triaging or routing based on different incident characteristics, such as MITRE tactics, incident severity, tags, comments, or the data source. This varies for each customer and their SOC processes. Even when the original analytics rule is configured with a specific severity, an automation rule (covered in Chapter 7, "Automating response") can update the severity, owner, tag, and comments for a more efficient triage and routing process.

The incident side panel shows the incident entities, MITRE tactics and techniques, the sending source, and the status of the incident. From this pane, several actions can be taken, such as assigning the incident to an analyst or changing the status or severity, as shown in Figure 4-10.

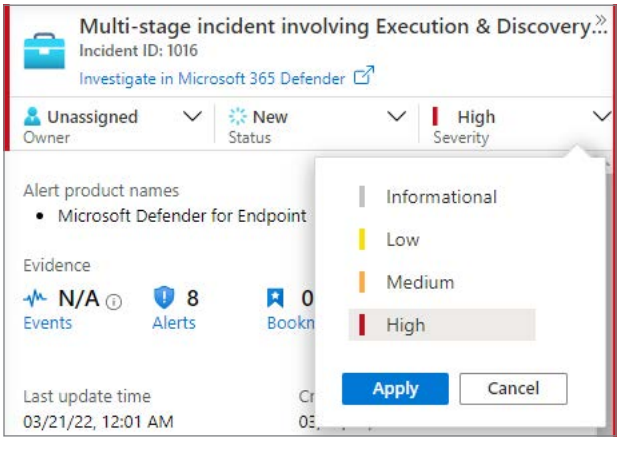


FIGURE 4-10 Incident actions

At the bottom of the incident pane is a link to the **Incident Overview** Workbook. If you have previously created a team for this incident, then the link to the **Incident Team** site will appear here. If **Tags** or **Comments** have been added (either manually or through automation), they can be found here, too. Clicking the **+** sign allows you to add a new tag. Further down, you can add a new comment, as shown in Figures 4-11 and 4-12.

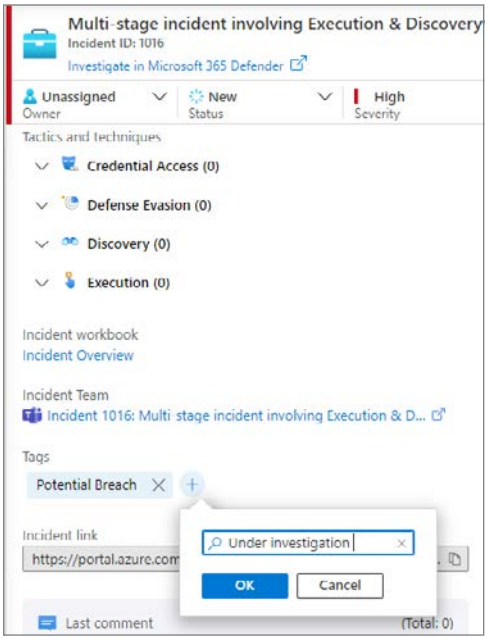


FIGURE 4-11 The Incident Overview Workbook, Incident Team, Tags, and incident URL

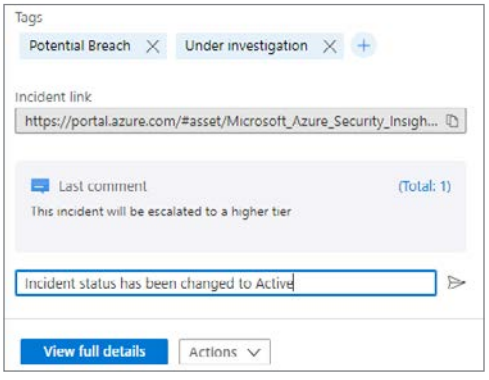


FIGURE 4-12 A new comment being added to the incident

While still in the Incident pane, you can view the full incident details by clicking on the **View Full Details** button, or you can select the **Actions** dropdown, which provides additional actions—**Investigate**, **Run A Playbook**, **Create Automation Rule**, or **Create Team (Preview)**, as shown in Figure 4-13.

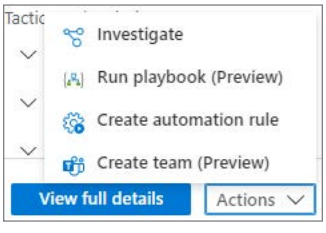


FIGURE 4-13 Incident actions, including investigate, run a playbook, create an automation rule, or create a Teams site

Another option to invoke incident actions is to select the ellipsis (**...**), as shown in Figure 4-14.

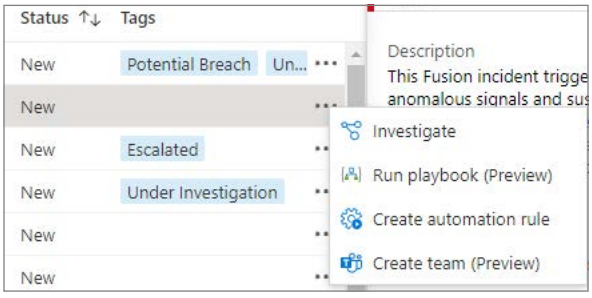


FIGURE 4-14 Incident actions invoked by clicking the ellipsis next to each incident

NOTE **The integration with Microsoft Teams and the Investigate feature will be covered later in this chapter in the "Teams integration" section.**

Searching for specific incidents

Microsoft Sentinel can capture a large number of incidents for the analyst to triage. Key in this process is the ability to quickly search through incidents. The search field allows you to search using the basic fields, like incident title, owner, and so on. Also, you can perform an advanced search, such as searching for specific entities. For example, you might want to search through all the incidents to find incidents with the same host. Figure 4-15 shows an example of searching for a host entity, `WIN2019`, where the columns have been reordered to show the tactics first.

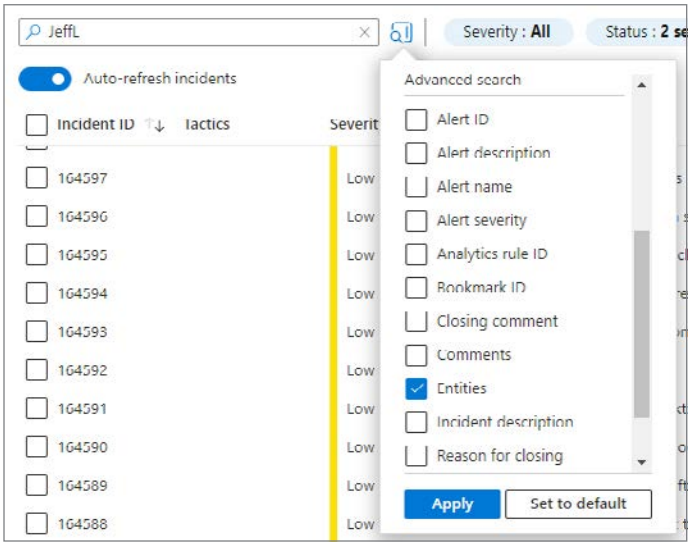


FIGURE 4-15 Search for the entity WIN2019 entity

To search for a specific incident that matches your search criteria, follow these steps:

1. While in the **Incidents** blade, select the magnifier icon and select your search fields to search through, as shown in Figure 4-16.

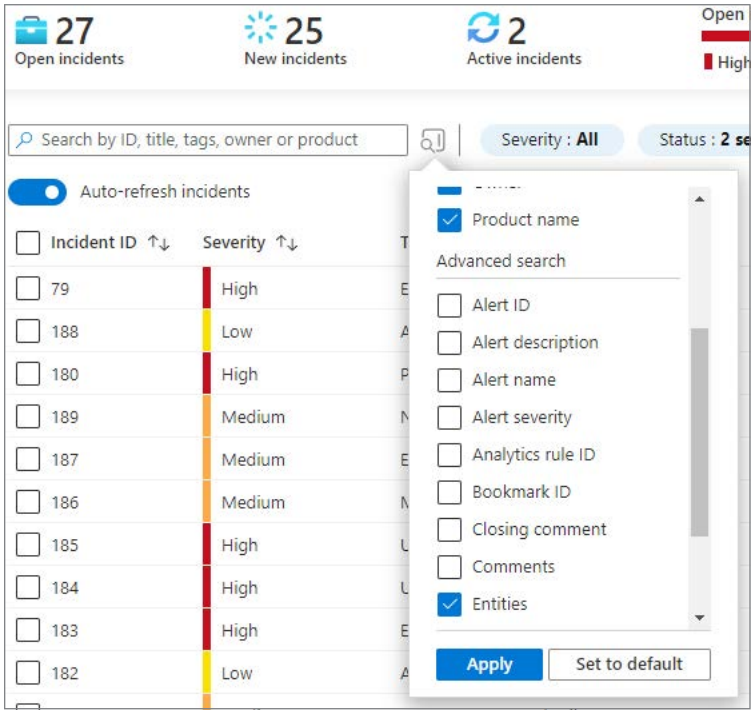


FIGURE 4-16 Incident search, with the columns reordered showing Incident ID first, followed by Severity

2. Type your search criteria and press Enter to show your search results.

Incident details

After the first round of initial triage, the SOC analyst most likely wants see more incident details. This can be done by clicking **View Full Details,** as shown in the previous sections. This opens the incident details view, where more details are shown for the SOC analyst. Here, you can find multiple tabs, like the timeline and other useful information.

As shown in Figure 4-17, to view the incidents details, follow these steps:

1. From the **Sentinel Overview** page, under **Threat Management**, select **Incidents**.
2. To select an incident, click once on the incident. This will make your selection active, which updates the right pane with the incident information.
3. In the incident pane on the right, select **View Full Details.**

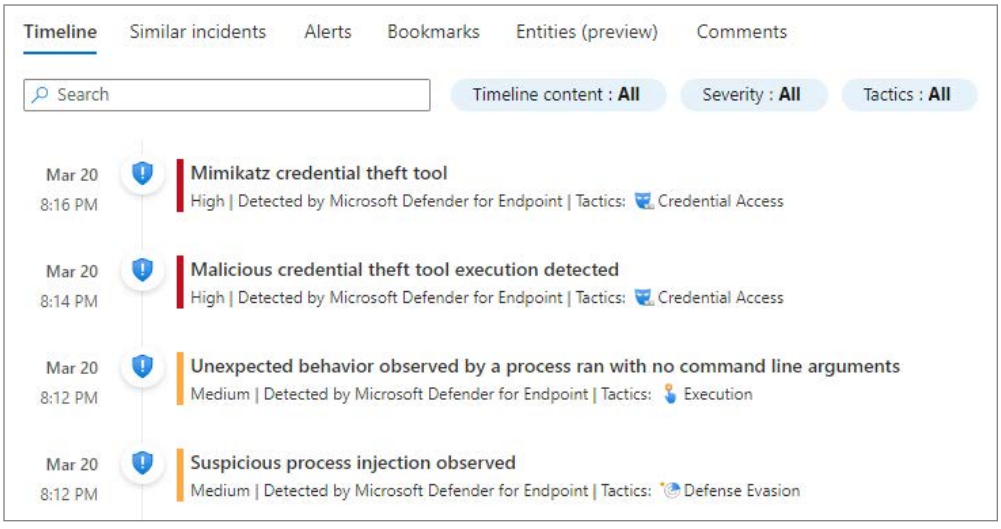


FIGURE 4-17 Incident timeline

The **Timeline** contains alerts, bookmarks, and activities. You can also choose the **Similar Incidents** tab, which shows incidents with common entities, incidents coming from the same rule, or incidents sharing the same custom details, as shown in Figure 4-18.

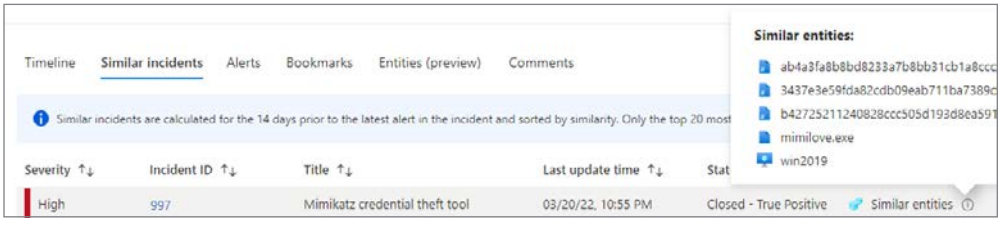


FIGURE 4-18 Similar incidents tab

NOTE **The Similar incidents view will also show closed or resolved incidents that are using the same analytics rule or have common entities to give you insights into historical patterns.**

The **Alerts** tab will show all alerts that are part of the incident. If an investigation results in the creation of a bookmark (a saved hunting query as discussed in Chapter 5, "Hunting"), then these will show up on the **Bookmarks** tab. The **Entities** tab will list all entities that are part of the incident, such as a computer, account, IP address, and so on. Finally, the **Comments** tab contains all the comments that the analyst has added as part of the investigation or that have been added through automation.

When you add a new comment, you can add comments using rich text options, such as bold, italic, and the like. Also, you can select different headings, add hyperlinks, or add images, as shown in Figure 4-19.

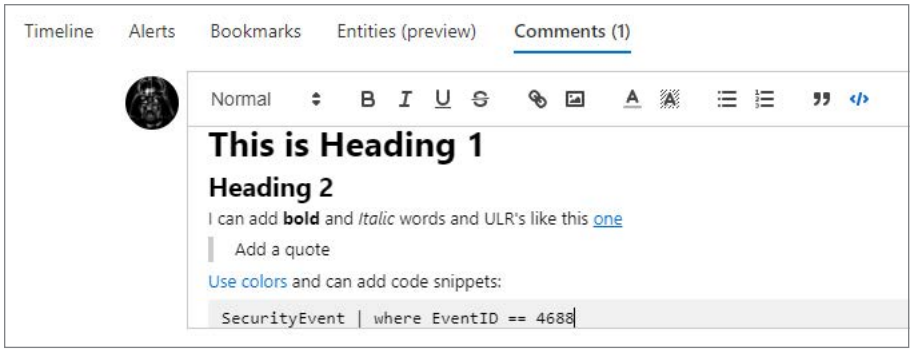


FIGURE 4-19 Adding rich text comments

The incident details view has been designed to be actionable. For example, under the **Entities** tab, by clicking on one of the entities, you will be routed to the **Entity Behavior** page. Figure 4-20 shows an example of a Windows host that has been part of an incident.

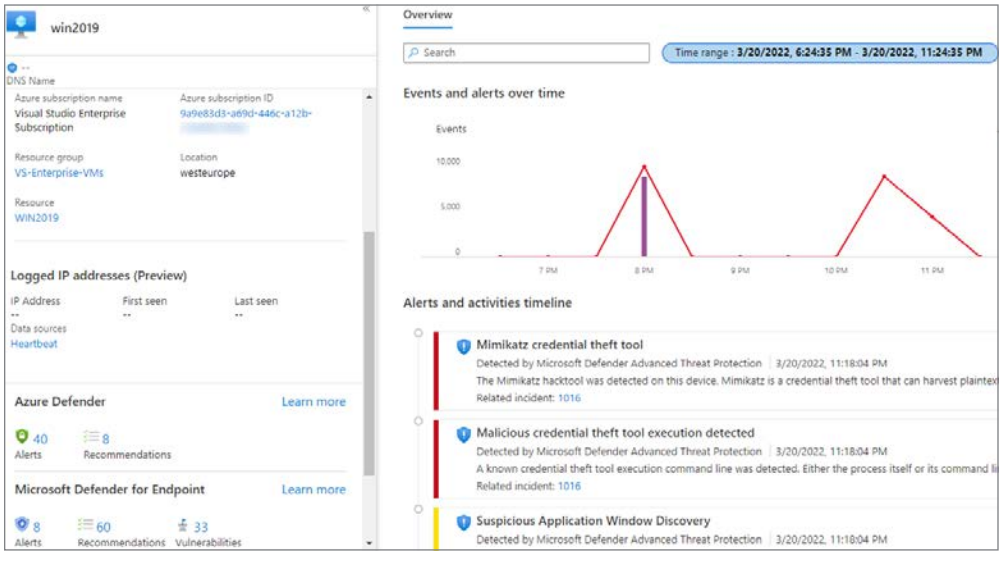


FIGURE 4-20 A detailed view of a host entity that is part of an incident

To open the **Entity page**, click one of the entities from left blade of the **Incidents Details** page, as shown in Figure 4-21.

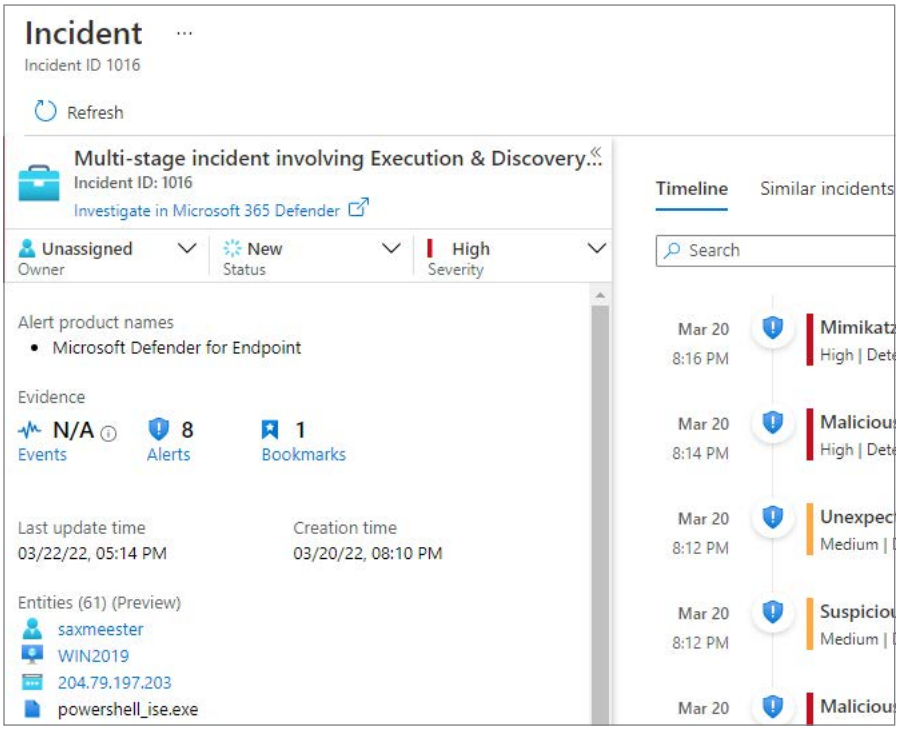


FIGURE 4-21 Incident details page with entities that can be explored with entity pages

NOTE **Future versions of Microsoft Sentinel will cover more entities beyond accounts, hosts, and IP addresses for the entity pages.**

For example, clicking the IP address shows geolocation information, log activities, and logged hosts, as shown in Figure 4-22.

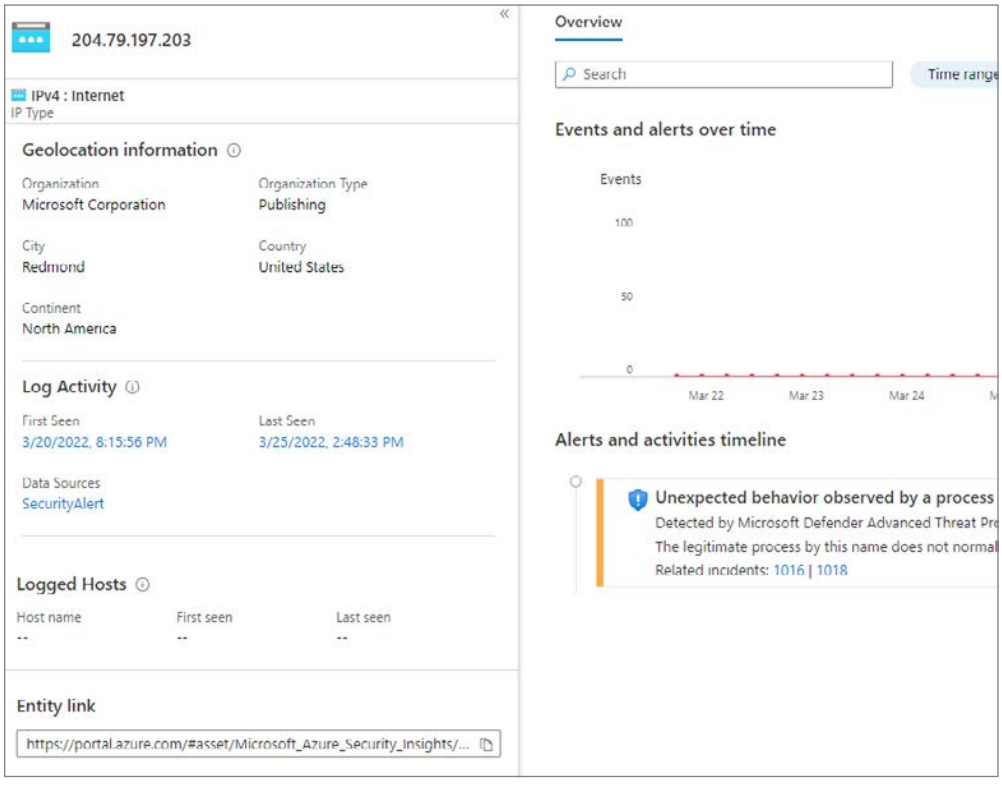


FIGURE 4-22 Entity page for the selected entity

The **Entity Behavior** pages allow you to investigate multiple entities like an IP address, host, or account. These entities will be enriched with alerts, activities timelines, and insights, and it also includes information from connectors, such as Microsoft Defender. For example, IP addresses will be enriched with geolocation information, and will show whether the IP address is matched with threat intelligence indicators.

The multi-stage incident, which was used as an example in the previous section, originates from the fusion detection rule, which is enabled by default and can be configured to include or exclude alert providers, as shown in Figure 4-23.

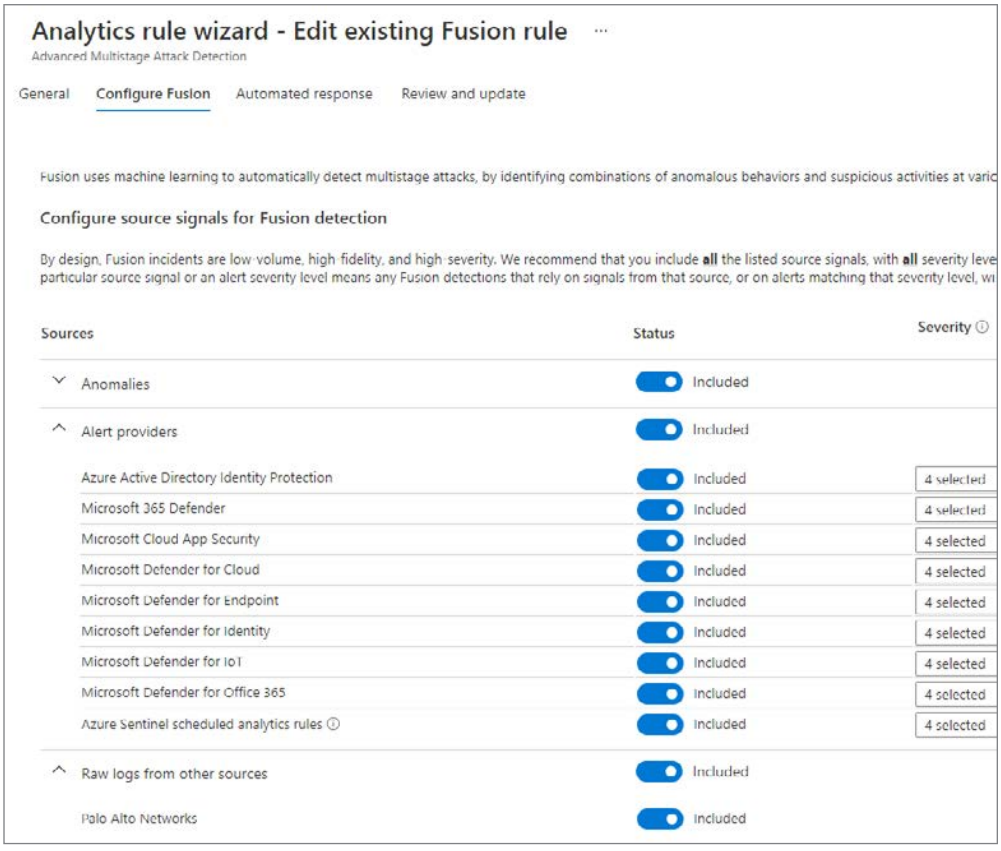


FIGURE 4-23 Fusion configuration to include or exclude alert providers

To configure fusion for the Advanced Multistage Attack Detection rule, follow these steps:

1. Open the **Azure portal** and sign in as a user who has Microsoft Sentinel contributor privileges.
2. Under **Configuration**, select **Analytics**.
3. In the search field, search for **Advanced Multistage Attack Detection**.
4. In the **Details** pane, select **Edit**.
5. Click the **Configure Fusion** tab.
6. Configure which alert providers or anomalies should be included.
7. Select the appropriate severity.

Teams integration

It has become more common for analysts to work together on the same incident. This is especially relevant in a multi-stage incident where signals are coming from multiple data sources. For efficient and smooth sharing of data, Microsoft Sentinel integrates with Microsoft Teams and leverages the Teams concept to easily work together.

Creating a Teams site can be done in the incident side panel or in the incident details by clicking the **Actions** dropdown and selecting **Create Team,** as shown in Figure 4-24.

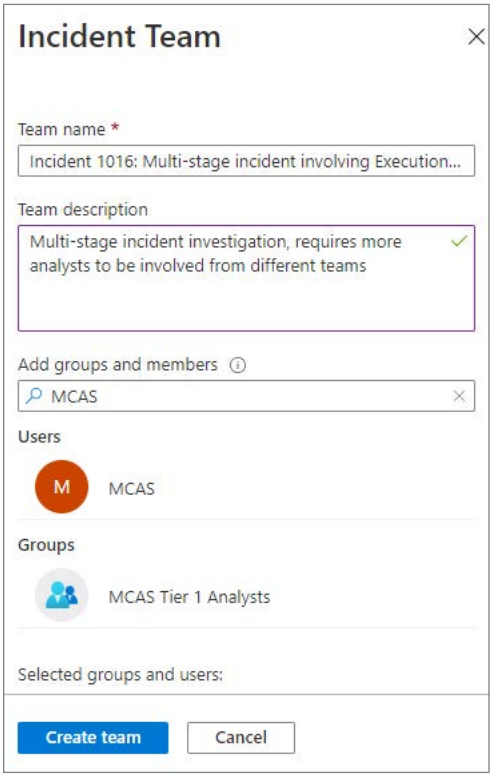


FIGURE 4-24 Microsoft Teams site creation for a group of analysts to work together

After the Teams site has been created, analysts can work together as they would do with any other Teams site. You can add conversations, channels, files, notes from OneNote, Wiki pages, and the like, and you can choose **Add More People** to work together, as shown in Figure 4-25.

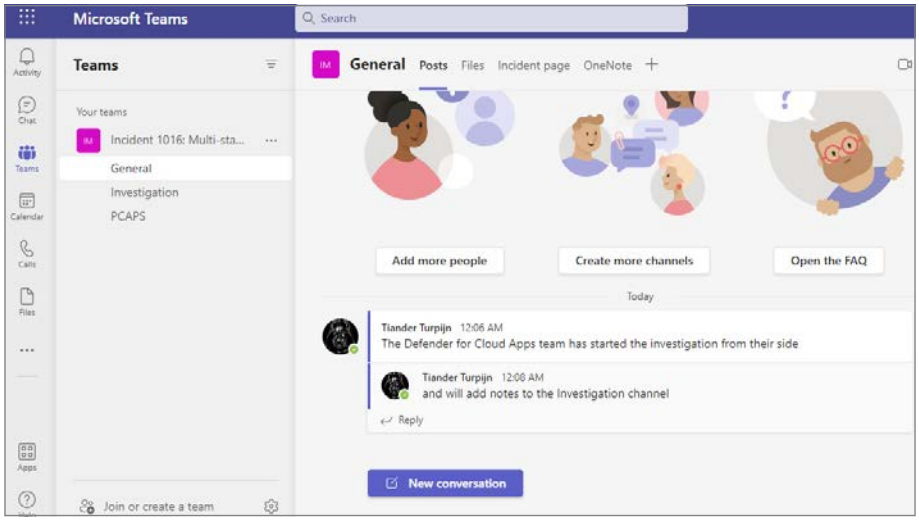


FIGURE 4-25 Microsoft Teams

NOTE **When an incident is resolved and closed, the Teams site will be archived automatically.**

Also, analysts can work together while investigating an incident by using the Teams whiteboard to investigate a Defender for IoT, as shown in Figure 4-26.

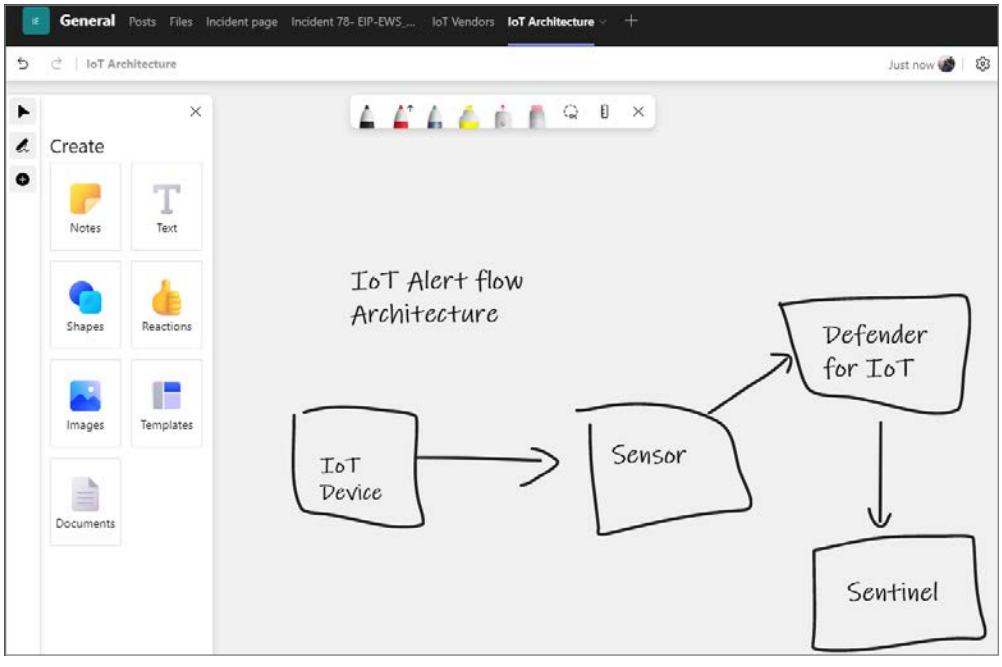


FIGURE 4-26 Analysts using a Teams whiteboard to collaborate

Graphical investigation

Some analysts may prefer a visual investigation of an incident because it allows the analyst to explore relationships with other alerts or entities that—in the context of the incident that is currently under investigation—are unrelated.

The investigation graphs allow you to reveal and connect these relationships so that they become part of the incident. In the next example, you will look at a Defender for IoT incident. Follow the steps below to start a graphical investigation:

1. Open the **Azure portal** and sign in as a user who has Microsoft Sentinel Responder privileges.
2. Under **Threat Management**, select **Incidents**.
3. Click one of your incidents.
4. In the **Incident Details** pane on the right, select the **Actions** button and select **Investigate**.

A graphical representation of the Defender for IoT incident is now shown with entity relationships (see Figure 4-27).

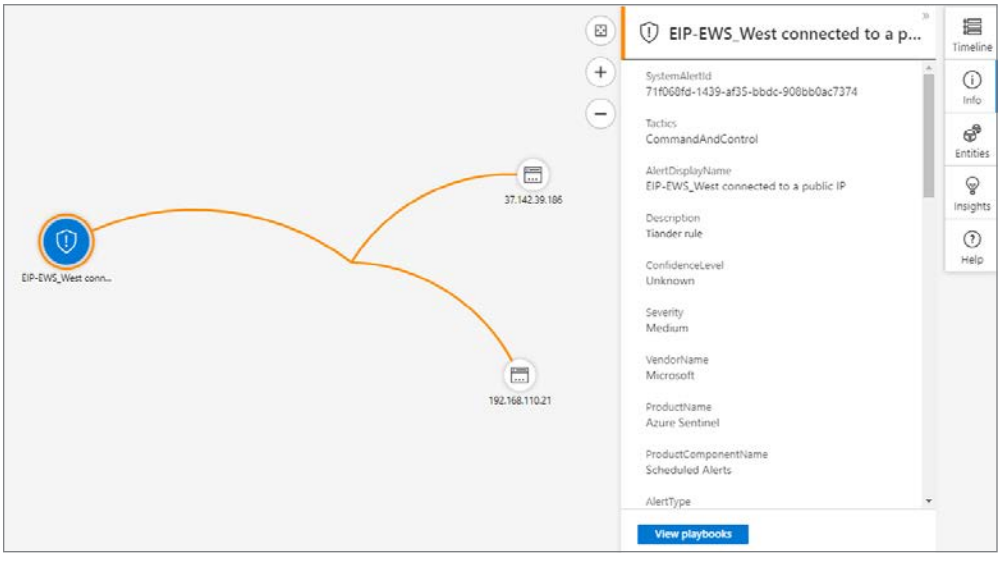


FIGURE 4-27 Graphical investigation of an incident and entity relationships

The view provides insights into the incident **Timeline**, which reveals **Entities** and **Insights**. Also, this view allows you to explore related alerts and entities. You can view related alerts and entities that are not part of the incident by clicking one of the entities and selecting **Related Alerts**, as shown in Figure 4-28.

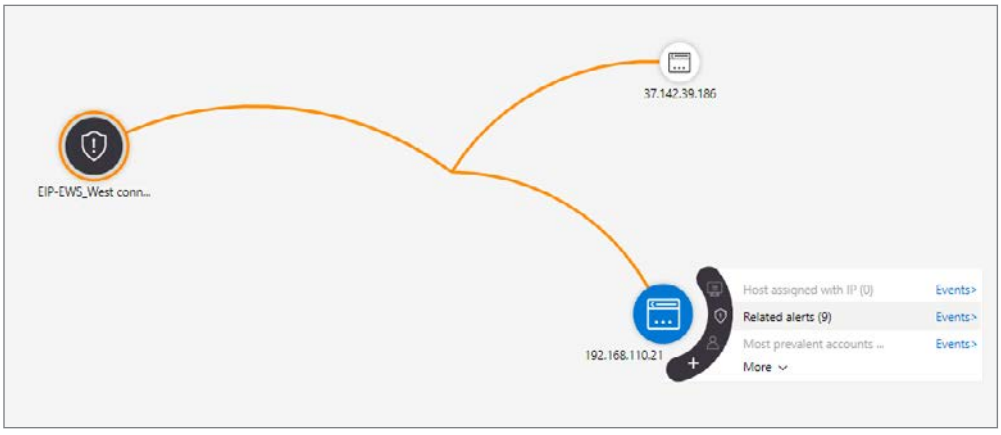


FIGURE 4-28 Exploring related alerts to an incident entity

To explore related alerts and entities, follow these steps:

1. While still in the investigation view, select an entity and select **Related Alerts**. This will add related alerts to the investigation view, as shown in Figure 4-29.

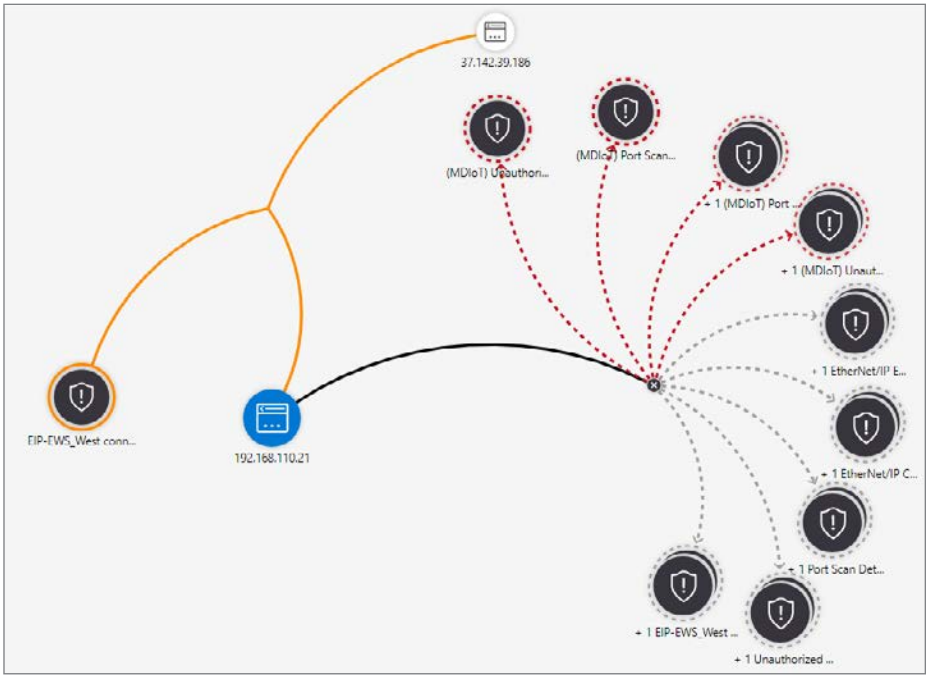


FIGURE 4-29 Related alerts are added to the investigation view

2. From this dashboard, you can continue your triage or investigation. The dotted lines show alerts that are currently not part of the incident you are triaging. While looking more closely at this Defender for IoT incident example, you might see an alert that you consider suspicious and related. For example, you might see the Port Scan alert, which you might want to add to your incident. To do this, you can select the alert, and on the floating menu, select **Add Alert To Incident**, as shown in Figure 4-30.

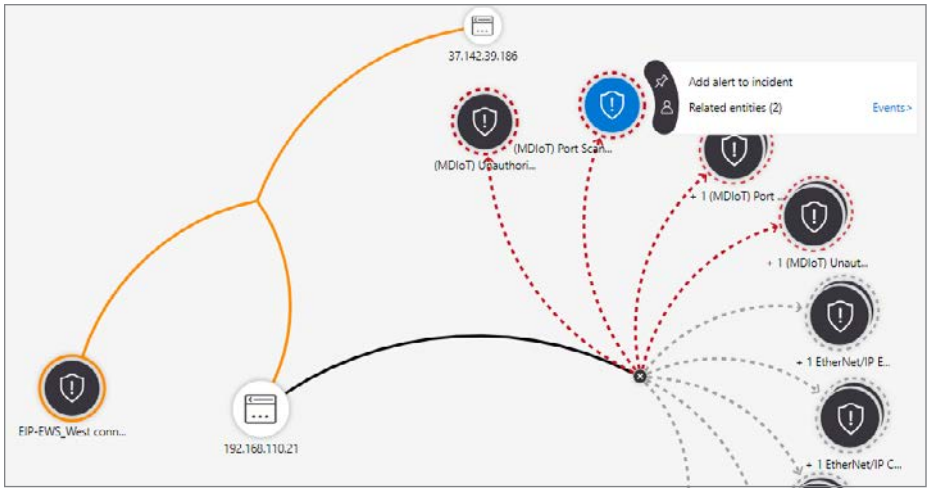


FIGURE 4-30 Adding related alerts to your incident

Once this action is performed, you will notice that the alert is added to the incident **Timeline,** as shown in Figure 4-31.

FIGURE 4-31 A related alert has been added to the incident

If the timeline does not show the added alert, select **Refresh** in the upper-left corner.

The incident details view, which shows the incident timeline, also provides filtering capabilities, which are especially useful if the timeline has multiple alerts or bookmarks. Also, you can filter for **Severity** and **Tactics**. Figure 4-32 shows **Tactics** filtering.

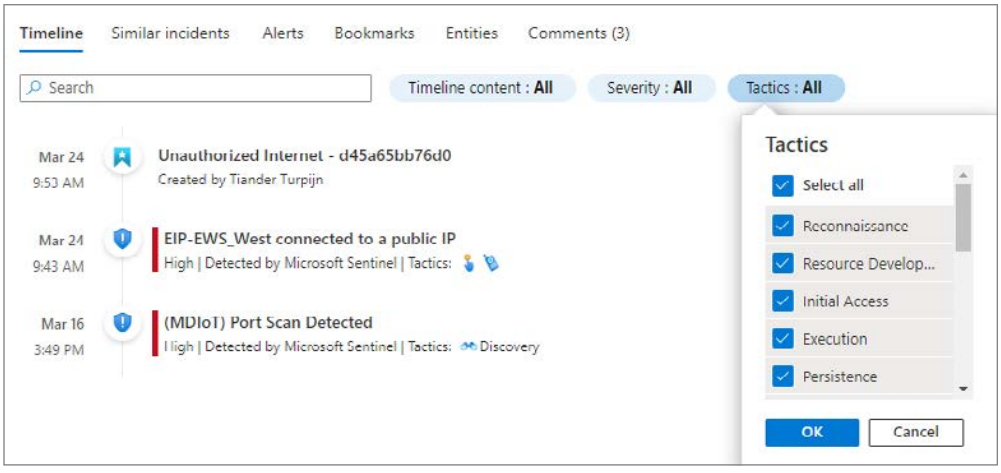


FIGURE 4-32 Filtering for MITRE Tactics in the incident details blade

Now that you have completed your triage and initial investigation, Chapter 5, "Hunting" discusses managing true positives or anomalous activities that require a deeper investigation—also known as threat hunting.

Chapter 5

Hunting

In the previous chapters, you have learned about authoring detection rules and how to triage and manage incidents. In this chapter, you will learn about hunting—threat hunting to be more specific. In the world of cybersecurity, *threat hunting* is about proactively searching for threats or a set of activities that you have not previously detected. This is the main difference between incident response (IR) and alert triage, where you are investigating a detection or an alert. The focus of this chapter is to learn about Microsoft Sentinel's threat-hunting capabilities.

Understanding threat hunting

Typically, investigating an incident or an alert starts with the assumption of a true positive. Threat hunting starts with a hypothesis. This might be a rising campaign (a coordinated email attack against one or many organizations), a Twitter message, a security blog, or any other information stream that triggers you to reassess the current state of your SOC's detection capabilities. Can you find traces of a potential breach or compromise? Would your organization be vulnerable to the threat you read about? To answer these questions, you would build a hypothesis. You would hunt for the so called "needle in the haystack."

Before you start building your hypothesis, it's important to understand the threat that you are looking for and what it would look like in your environment. You should build your hypothesis based on the following threat foundation:

- **Knowledge** About the threat itself, but also about how threat hunting in Microsoft Sentinel works.
- **Context** Which vulnerabilities is the threat based on? Is the threat specific to an operating system, version, or application? Under which circumstances can the threat occur?
- **Data depth** Do I have the appropriate level of logging or auditing enabled? Do I have the data ingested to find the threat?
- **Data breadth** Do I have sufficient data sources coverage? You don't want to pivot point to point.

Spending time on your hypothesis preparation is equally important as defining your focus and how achievable your hunting effort will be. Hunting for suspicious logins might sound interesting, but that is most likely too broad and needs more focus. Are you going to look for any type of login, like the ones that occurred last week or last month? The threat also needs to be relevant to your environment. Also, be prepared to not find anything. That is not a bad thing; threat hunting is a loop that you will be continuously executing and evolving over time.

This is the reason that documentation is an important aspect of threat hunting. What did you hunt for? How did you hunt? Which time range did you hunt? Your threat hunting documentation needs to be shared with your fellow hunters, which leads to becoming an efficient hunting team. Assuming that you have been hunting for the correct indicators of compromise (IoC), your next hunt based on a new time range might be successful. Ideally, you want to be able to simulate the threat or the circumstances that the threat depends on, so that you can validate your hunting against a simulation.

Knowing your environment and data

Before diving into threat hunting in Microsoft Sentinel, ask yourself if you have a good picture of your environment and the key assets in it. Which machines are key to your business and contain your crown jewels? Which privileges accounts do you have? What data sources and entities are available to you? It doesn't make sense to start threat hunting for AWS threats if you are not ingesting the right data, like missing account information. Maybe you don't have agents installed on your key machines or you are not collecting the appropriate level of information.

Knowing your environment and data will provide you with insights into what is anomalous for your environment. For example, if you have an internal web server, you would typically not expect public IP addresses connecting to your web server. Another example could be a patching account that would connect outside your patching maintenance windows in an unexpected context, like accessing a SQL database. Your insights would also cover what is common and expected in your environment, so you don't focus on those events. The power of Kusto Query Language (KQL), the search language in Microsoft Sentinel, helps you in identifying anomalous and non-anomalous behavior.

Threat hunting in Microsoft Sentinel

Based on your level of expertise and your own personal preferences, Microsoft Sentinel offers several options for threat hunting. This chapter introduces you to all of them. Microsoft Sentinel has a dedicated page for you to perform threat hunting. Follow the steps below to access the Hunting blade:

1. Open the **Azure portal** and sign in as a user who has Microsoft Sentinel Reader privileges.
2. Under the **Threat Management** section, click **Hunting**.

3. The **Hunting** page appears on the right side with several options to explore, as shown in Figure 5-1.

FIGURE 5-1 The Hunting blade top actions and counts

4. The number **192/224** represents the relevant hunting queries in your environment (192) based on the total number of queries (224). This means that 32 queries are not relevant in your environment, such as missing data sources.
5. In Figure 5-1, the **green arrow** with the number **8** represents the number of new queries added recently. Clicking the arrow shows which new queries have been added.
6. Above the number of queries is the time range you can select for running your queries. (The default is the **Last 24 Hours.**)
7. The other options are:
   - Click **New Query** if you want to create a new hunting query.
   - Click **Run All Queries** if you want to run all available queries.
   - If you select one or more queries, this will change into **Run Selected Queries.**
8. If you run queries, the **Result Count / Queries** run will show how many queries have returned results (**Result Count**) and how many queries have run (**Queries Run**). These numbers will be updated while the queries are running.
9. **Livestream Results** will show how many results have been returned when running a Livestream. (Livestream is covered later in this chapter.)
10. **My Bookmarks** show how many bookmarks have been saved. (Bookmarks are covered later in this chapter.)
11. In the above menu, you can select **Columns** if you want to customize the columns shown in the hunting blade. For example, when in the **Queries** tab, adding the column **Results** will show you the number of results next to **Results Delta Percentage,** which shows only the delta percentage.
12. **Guides & Feedback** provides you with additional information, useful links, and the ability to share ideas and feedback.

13. The **MITRE Framework** integration allows you to filter by tactics by clicking the **MITRE Tactics** icons, as shown in Figure 5-2, where **Credential Access** has been selected.

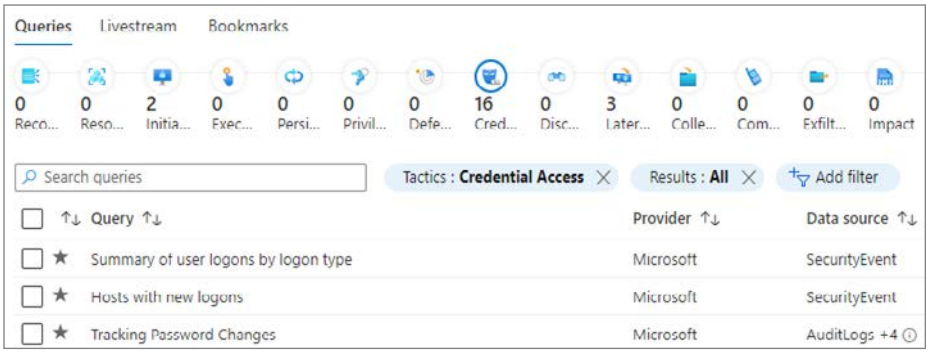


FIGURE 5-2 Hunting queries filtered by MITRE Tactics

14. You can also use the **Search Queries** field to type in your search criteria, such as searching for **Tracking Password Changes**, as shown in Figure 5-3.

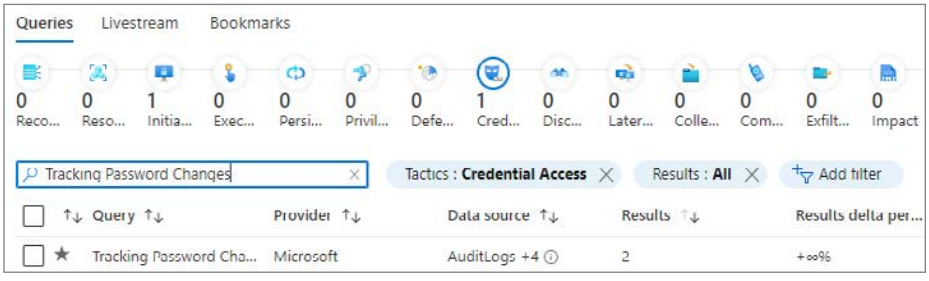


FIGURE 5-3 Use the search box to search for hunting queries

15. Next to the search field box, you can also click on **Tactics** to filter for the tactic name, as shown in Figure 5-4.

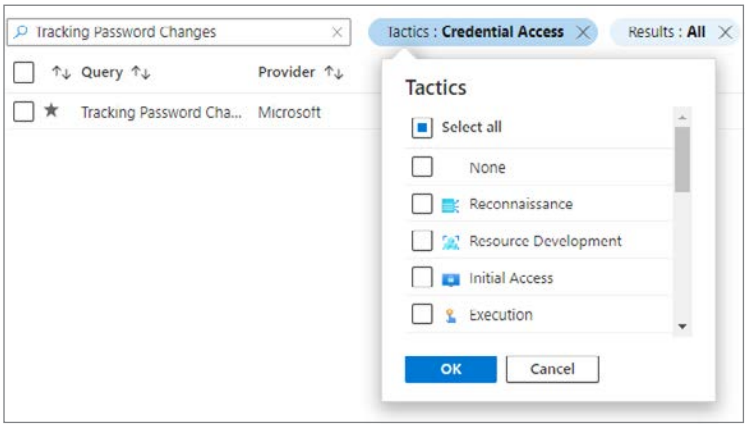


FIGURE 5-4 Selection of Tactics to filter

16. Next to the **Results** button, you will find the **Add Filter** button, as shown in Figure 5-5.

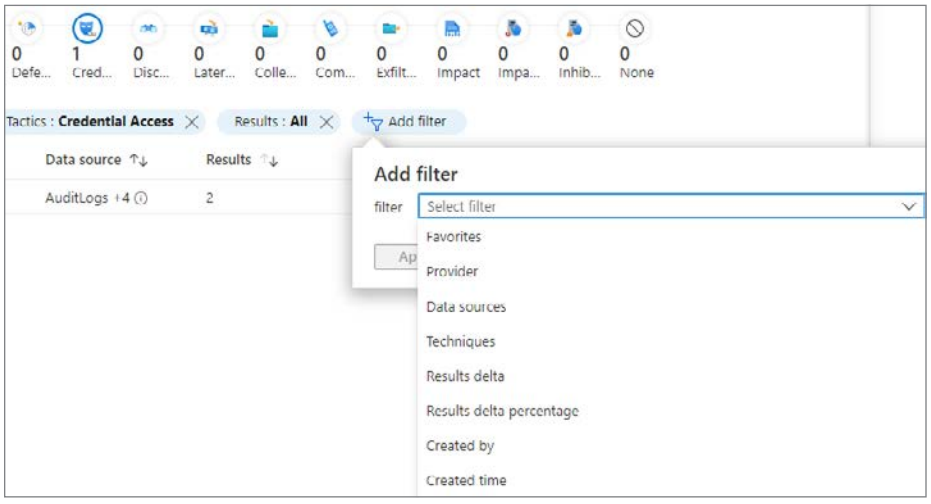


FIGURE 5-5 Add Filter button

NOTE **When no MITRE filtering has been applied, only the Add Filter button will be shown next to the Search Queries field.**

Now that you are familiar with the interface, you can start your first hunting experience in the next section.

Running your first hunting query

It's a good best practice to regularly visit the **Hunting** blade in Microsoft Sentinel to run queries and explore the results. Follow these steps to run queries in the **Hunting** blade:

1. Open the **Azure portal** and sign in as a user who has Microsoft Sentinel Reader privileges.
2. Under **Threat management**, click **Hunting**.
3. Change your time range to **Last 24 Hours**.
4. Click **Run All Queries**.
5. The **Result Count / Queries** run will be updated while the queries are running.
6. You can sort the columns, such as **Results** or **Results Delta Percentage**, as explained in the previous section.

Explore the queries that have results, as shown in Figure 5-6.

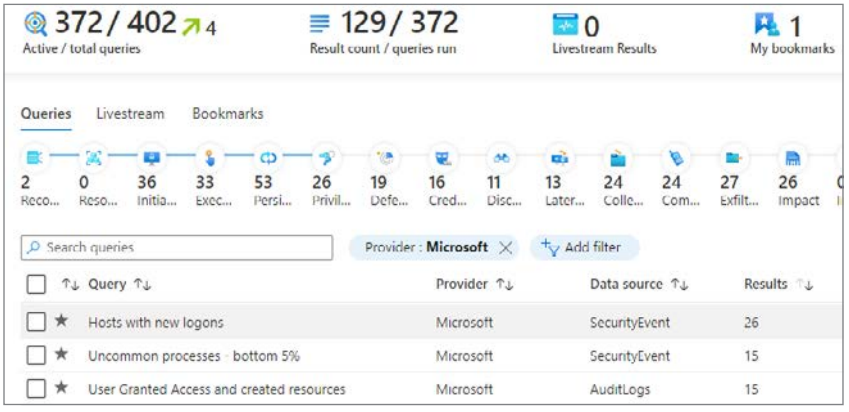


FIGURE 5-6 Hunting query results

1. One of the queries that might be interesting to explore is the one at the top with the title, **Hosts With New Logons**. Remember, in the previous section, we discussed how an attacker could move laterally using a compromised account, which was used for patching. This query could reveal a compromised patching account trying to access a new host to move laterally across the organization.
2. Clicking **View Results** in the right pane brings you to the query results, as shown in Figure 5-7.

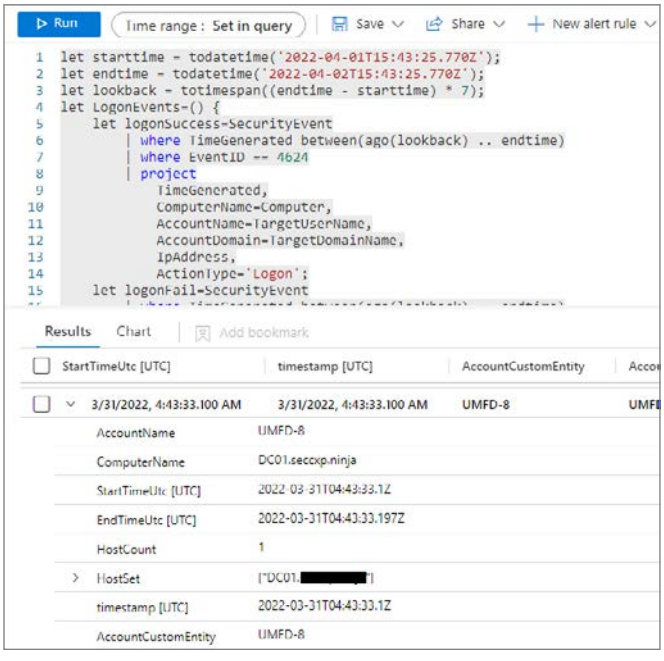


FIGURE 5-7 Hunting query details

3. From here, you can continue your investigation, exploring this further if it's a potential malicious event or moving on if you are satisfied that this login is expected.
4. Continue your exploration of hunting queries that returned results.

Hunting hypothesis example

Based on what you have learned so far, you are going to explore a sample hunting hypothesis based on the following scenario:

> *A malicious actor, using a brute-force attack, has compromised the credentials of one of your Windows administrators. These credentials were used to log in between 03/15/2022 and 04/05/2022.*

The first step is to revisit the threat-hunting fundamentals and questions for this hypothesis:

- **Knowledge** Do you understand the threat and the Microsoft Sentinel hunting options?
- **Context** What are the circumstances and dependencies that make this threat possible?
- **Data depth** Are you ingesting the right data to hunt for this threat?
- **Data breadth** Do you have enough data sources covering the data that you need?

Our hypothesis is based on a brute-force attack using a variety of tools that are publicly available. You are familiar with the hunting options in Microsoft Sentinel and have built your Kusto Query Language (KQL) knowledge. The target is susceptible to unauthorized login, potentially based on an open public port, such as RDP port 3389. You have installed Microsoft Sentinel agents on all your computers, and they are collecting Security Event data, which would capture security information, such as failed and successful logins. With this information, it seems that you have covered the hunting fundamentals.

To start looking for evidence for your hypothesis, you can start in several ways. One of them might be to look at failed logins over a specific time to explore anomalies. That would reveal a brute-force attack attempt. The following query looks at failed logins over a period of 7 days, summarizes the Account, Computer, and IP address and renders this into a time chart:

```
SecurityEvent
| where TimeGenerated >ago(7d)
| where EventID == 4625
| summarize FailedLogins = count() by Account,Computer, IpAddress
| sort by FailedLogins desc
| render timechart
```

The result of this query is shown in Figure 5-8.

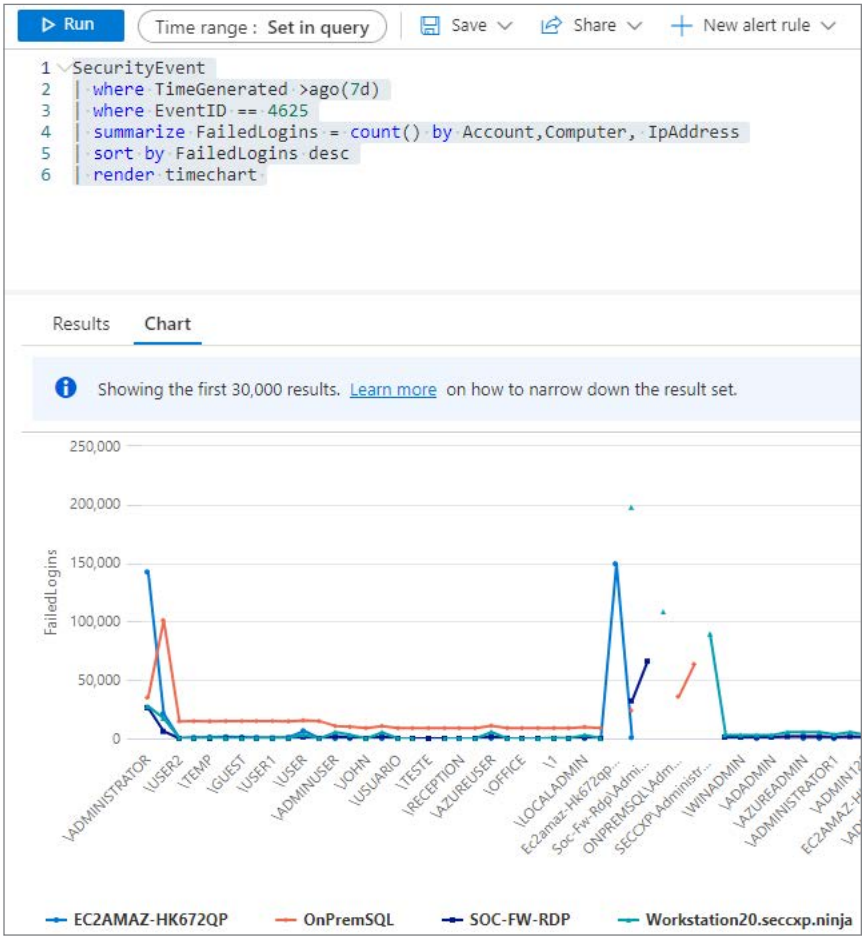


FIGURE 5-8 Result of running a query that could reveal a brute-force attack

When you hover over the time chart, you can focus on specific spikes and their data points. This can show you the account being used, the number of failed logins, and on which computer those logins occurred, as shown in Figure 5-9.

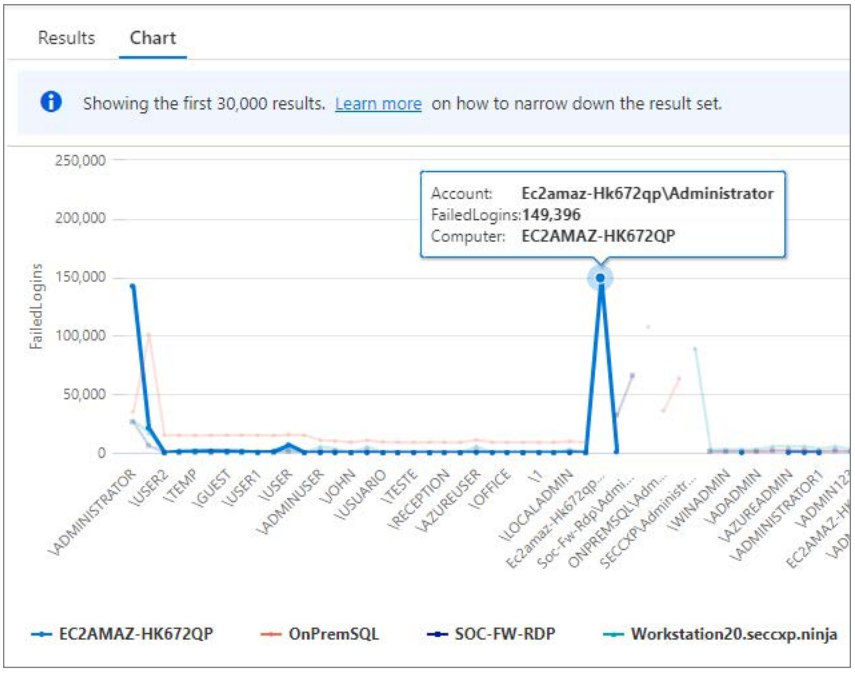


FIGURE 5-9 Hovering over the spikes reveals additional information

Follow these steps to execute the same query in your environment:

1. Open the **Azure portal** and sign in as a user who has Microsoft Sentinel Reader privileges.
2. Under **General**, click **Logs**.
3. Copy and paste the following query in the **Run** field:

```
SecurityEvent
| where EventID == 4625
| summarize FailedLogins = count() by Account,Computer, IpAddress
| sort by FailedLogins desc
| render timechart
```

4. Click **Run**.
5. Hover over your spikes to see additional information. You might want to explore increasing the number of days to look back. The default **Time Range** is set to **Last 24 Hours.**
6. You can click each of the server names to hide them from your graph to focus on specific ones.

Because this looks like a clear lead for you to investigate, you can continue your hunt based on the account and computer and investigate the IP addresses used for that attack.

> ***TIP*** **Explore the Microsoft Sentinel hunting queries in the GitHub repository by visiting *https://aka.ms/SentinelHuntingQueries*.**

So far, you have been looking at failed logins without clear indicators that an account was compromised. Your next hunt will be based on a more suspicious pattern that will show a successful login after a specific number of failed logins within a specific time range. This is a common pattern for a successful brute-force attack. To look for this pattern, you are going to leverage the **Hunting** blade, as discussed earlier. To support the hypothesis, you have created a hunting query that you run regularly. By clicking the star icon, you have added your query as a favorite. Besides the filtering capability for your favorite queries, all your favorite hunting queries will run each time you visit the hunting blade. As you can see in Figure 5-10, a result of 2 is shown, and the details pane shows the mapped entities and the MITRE Tactics and Techniques alignment.

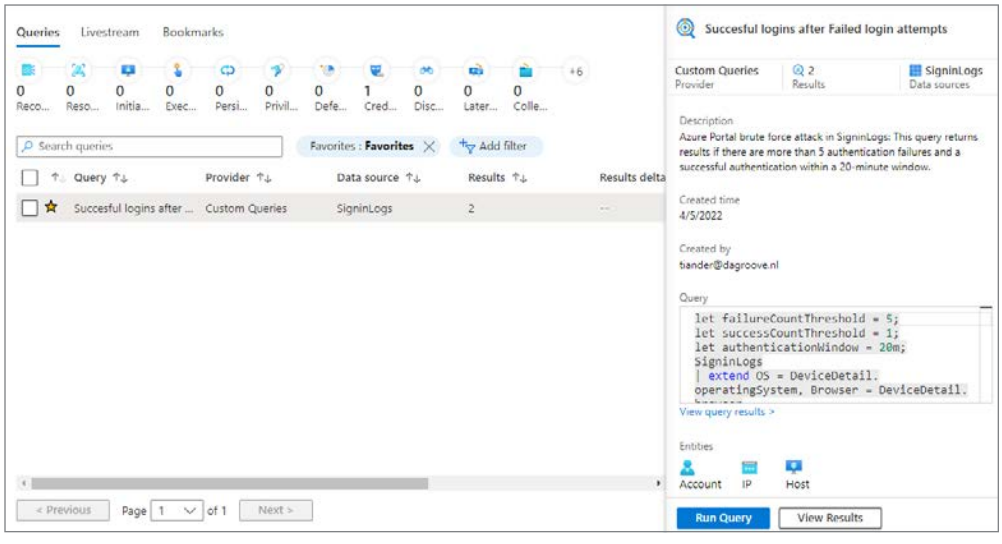


FIGURE 5-10 Hunting query result for a brute-force attack

Clicking **View Results** brings you to log search to see the query results, as shown in Figure 5-11.

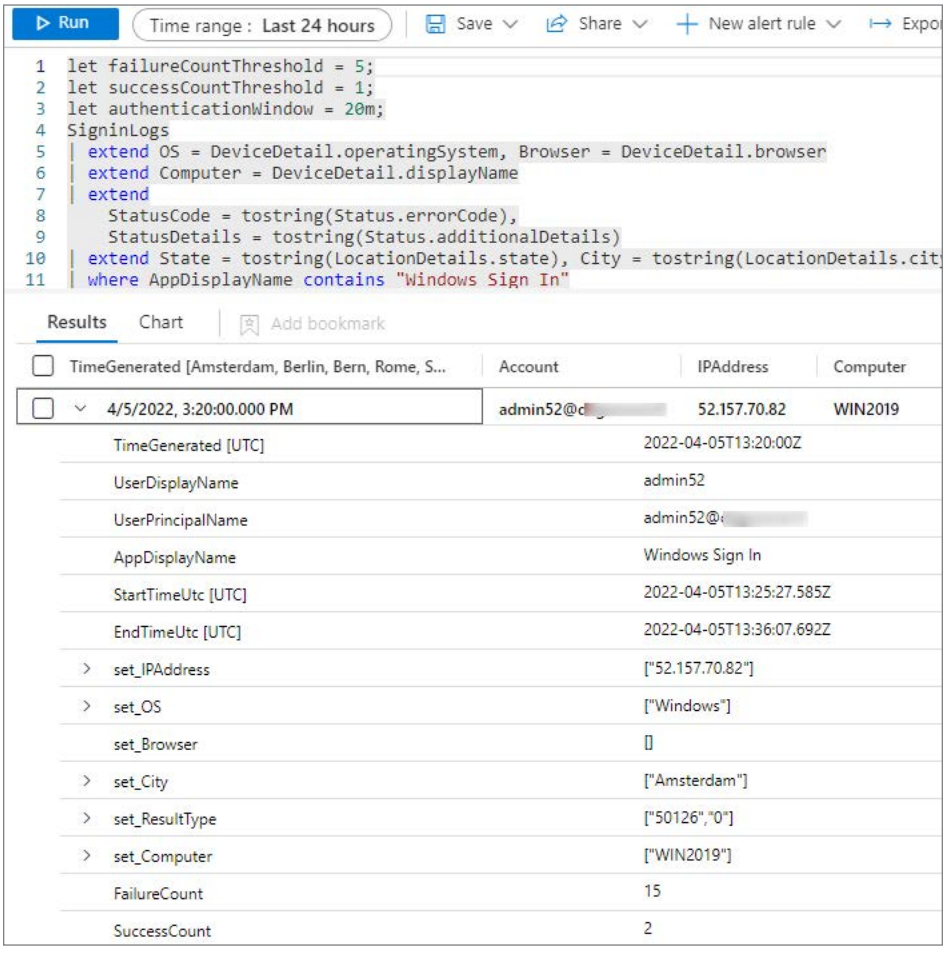


FIGURE 5-11 Hunting query detailed result for a brute-force attack

Because this looks very suspicious, look at the `FailureCount` and `SuccesCount` values within a duration of 20 minutes, and **Bookmark** this query so that you can add this evidence to your hunting. To create a hunting bookmark, select the query row and click the **Add Bookmark** button (see Figure 5-12).

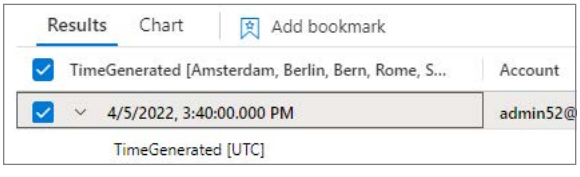


FIGURE 5-12 Adding a bookmark to your hunting

In the **Add Bookmark** details window, you can provide a **Bookmark Name**, **Event Time Mapping**, **Entity Mapping**, **Tactics & Techniques**, **Tags**, and **Notes** (see Figure 5-13).

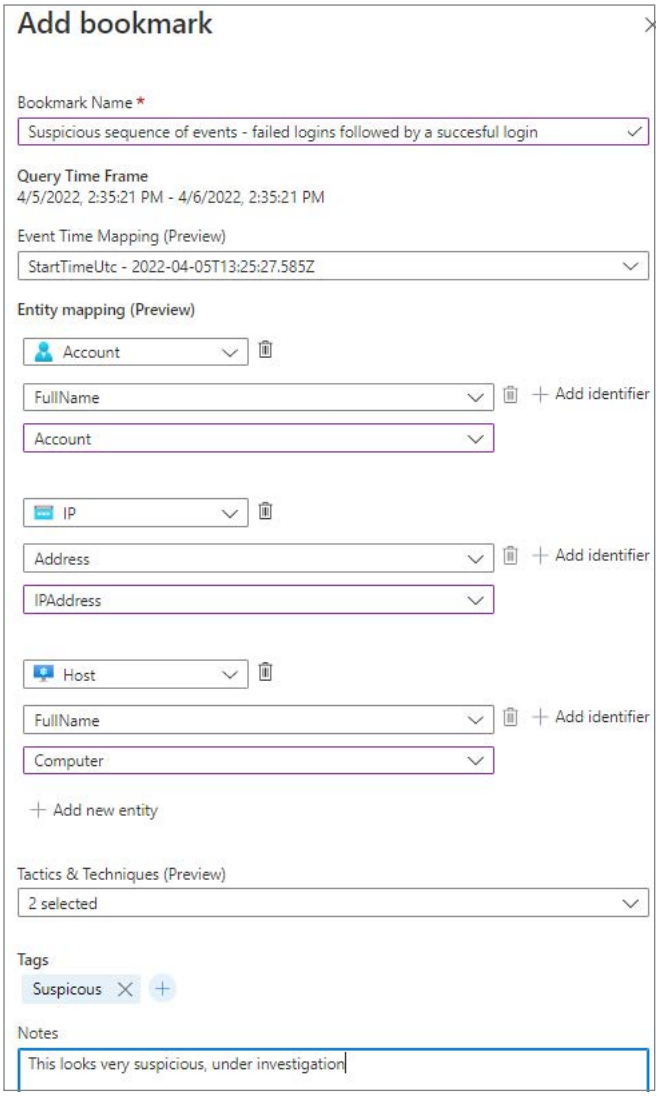


FIGURE 5-13 Adding bookmark details

From here, you can continue your hunting by changing the query or writing a new one. One possible route is to continue your investigation by going back to the **Hunting** page, selecting the bookmark that you just created, and exploring the graphical investigation by clicking **Investigate**, as shown in Figure 5-14.

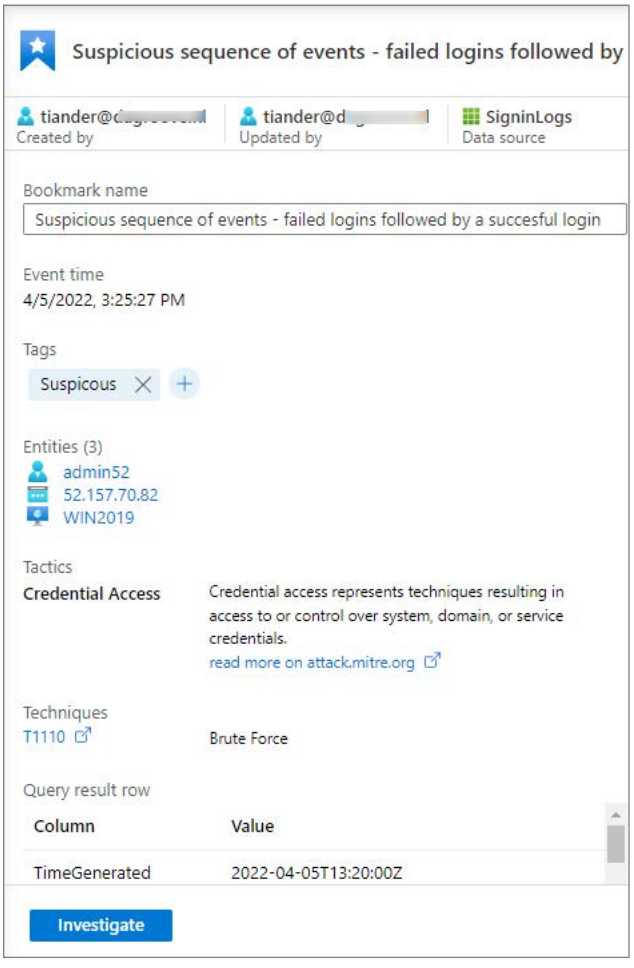


FIGURE 5-14 Bookmark details

This action brings you to the graphical investigation, which visually shows you the previously mapped entities (see Figure 5-15).

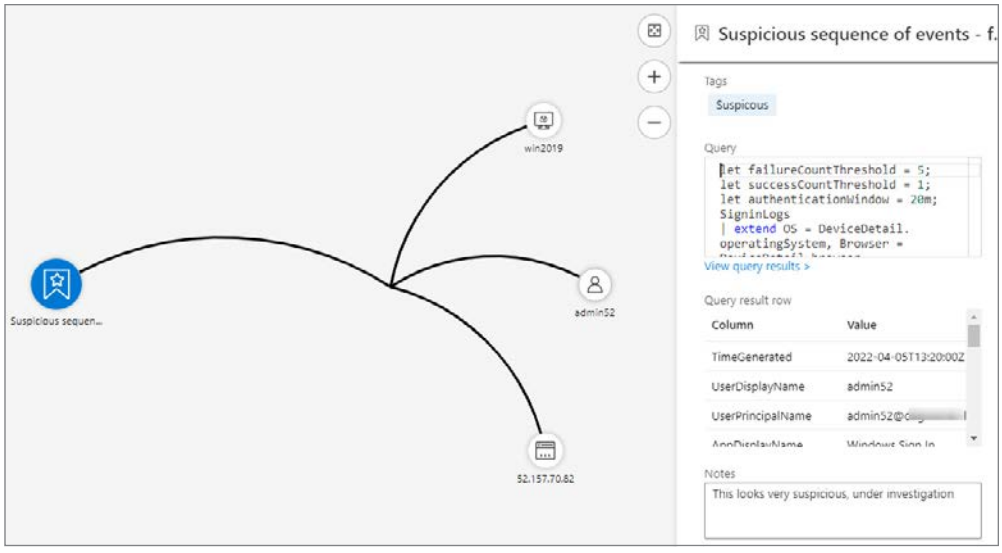


FIGURE 5-15 The Investigation graph visualizes the entities part of your bookmark.

This view allows you to continue your hunting process by exploring entity relationships. When you right-click the host entity (the Windows computer), and select **Related Alerts**, the view expands with related alerts, as shown in Figure 5-16.

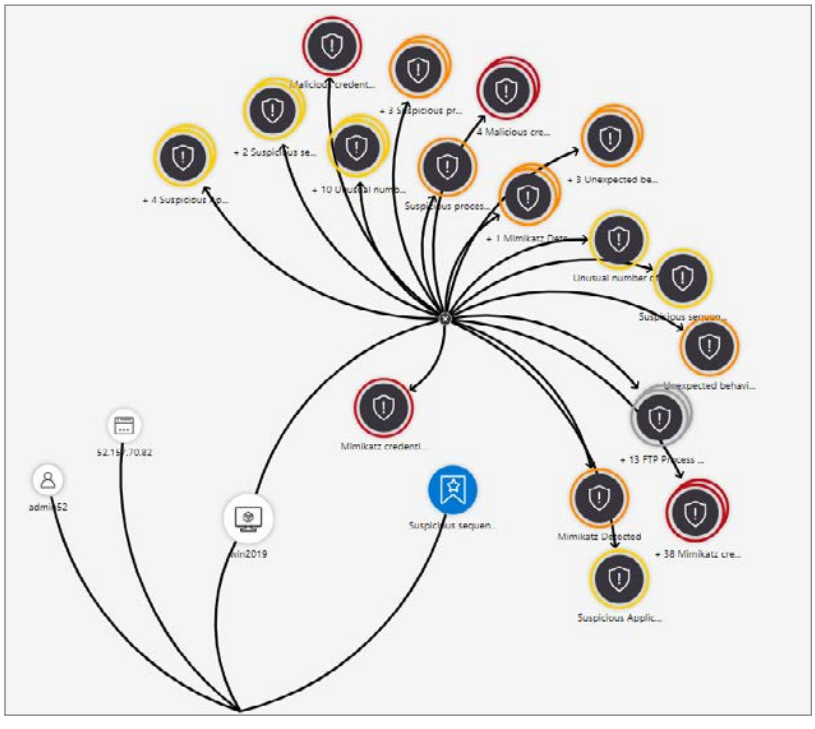


FIGURE 5-16 Investigation graph, expanded with related alerts

This view confirms your hypothesis, based on the Mimikatz-related alerts, which indicate credential theft. Go back to your bookmark and create an incident via the ellipsis (**...**) on the bookmark row, as shown in Figure 5-17.

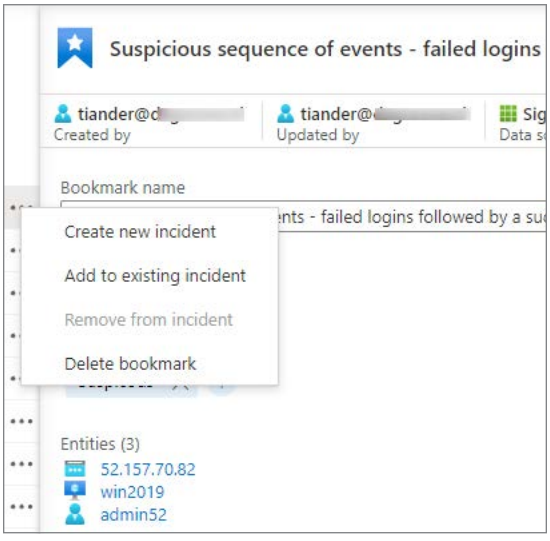


FIGURE 5-17 Create an incident from your hunting bookmark

You have now created a new incident from which you can continue your hunt and investigation, as shown in Figure 5-18.

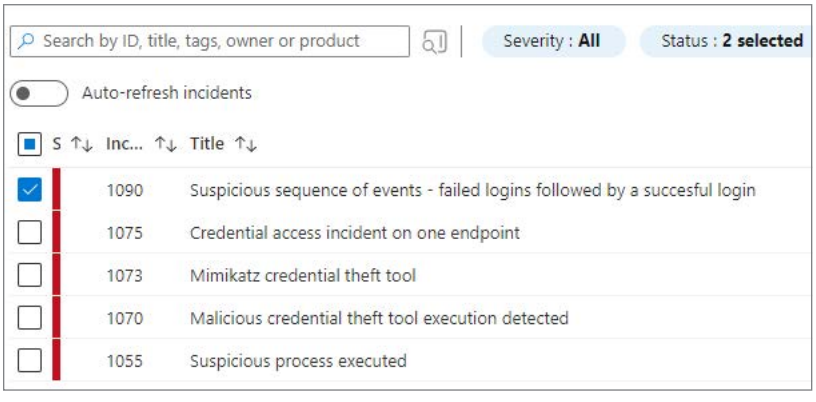


FIGURE 5-18 Incident created based on your bookmark

The following steps describe how you can create the hunting query that was used for the hypothesis example:

1. Open the **Azure portal** and sign in as a user who has Microsoft Sentinel Contributor privileges.
2. Under **Threat Management**, click **Hunting**.

3. With the **Queries** tab active, click **New Query**.
4. Provide a **Name** and **Description** for your hunting query.
5. In the **Custom Query** field, add the query shown in Listing 5-1.

LISTING 5-1 Hunting query to investigate suspicious logins

```
let failureCountThreshold = 5;
let successCountThreshold = 1;
let authenticationWindow = 20m;
SigninLogs
| extend OS = DeviceDetail.operatingSystem, Browser = DeviceDetail.browser
| extend Computer = DeviceDetail.displayName
| extend
    StatusCode = tostring(Status.errorCode),
    StatusDetails = tostring(Status.additionalDetails)
| extend State = tostring(LocationDetails.state), City = tostring(LocationDetails.
city)
| where AppDisplayName contains "Windows Sign In"
// Split out failure versus non-failure types
| extend FailureOrSuccess = iff(ResultType in ("0", "50125", "50140"), "Success",
"Failure")
| summarize StartTimeUtc = min(TimeGenerated), EndTimeUtc = max(TimeGenerated),
    makeset(IPAddress), makeset(OS), makeset(Browser), makeset(City),
makeset(ResultType), makeset(Computer),
    FailureCount=countif(FailureOrSuccess == "Failure"),
    SuccessCount = countif(FailureOrSuccess == "Success")
    by
    bin(TimeGenerated, authenticationWindow),
    UserDisplayName,
    UserPrincipalName,
    AppDisplayName
| where FailureCount >= failureCountThreshold and SuccessCount >=
successCountThreshold
| extend Account = UserPrincipalName
| extend IPAddress = set_IPAddress[0]
| extend Computer = set_Computer[0]
```

6. If you want to test against failed Azure portal logins instead of failed Windows logins, you can replace `where AppDisplayName contains "Windows Sign In"` with `where AppDisplayName contains "Azure portal"`. This provides results when a successful login follows failed logins to the Azure portal.
7. Map the **Account** entity with **FullName** and **UserPrincipalName**.
8. Map the **IP** entity with **Address** and **IPAddress**.
9. Map the **Host Entity** with **FullName** and **Computer**.
10. Under **Tactics & Techniques**, select your MITRE tactics and techniques.
11. Click **Save**.
12. When you click the star **icon**, you will save your new query as a **Favorite,** which runs every time you visit the **Hunting** blade.
13. Select the query and click **Run Selected Queries**.

14. When results are returned, click **View Results**.
15. Select the result row by clicking the checkbox and clicking **Add Bookmark**.
16. Provide the bookmark information and ensure that the entities are mapped as shown before.
17. After the bookmark has been created, it can take a couple of minutes for the entities to be correlated and added.
18. When the entities are mapped, you can click **Investigate**, which opens the graphical investigation blade.
19. From here, you can further explore related alerts and entities.
20. In the bookmark view, you can right-click your bookmark and select **Create New Incident**.

TIP **You can also add a bookmark to an existing incident by right-clicking a bookmark and selecting Add To Existing Incident.**

Livestream

The **Livestream** feature in Microsoft Sentinel allows you to create interactive sessions that let you run queries as events occur. Notifications from sessions when a match is found allow you to launch an investigation. Livestream is especially relevant when you suspect a breach attempt in progress. In the following basic example, you are going to monitor—through an interactive session—whether a new account is being created or deleted across your environment. You can create a new query for a Livestream**,** or you can right-click an existing query and select **Add To Livestream**, as shown in Figure 5-19.

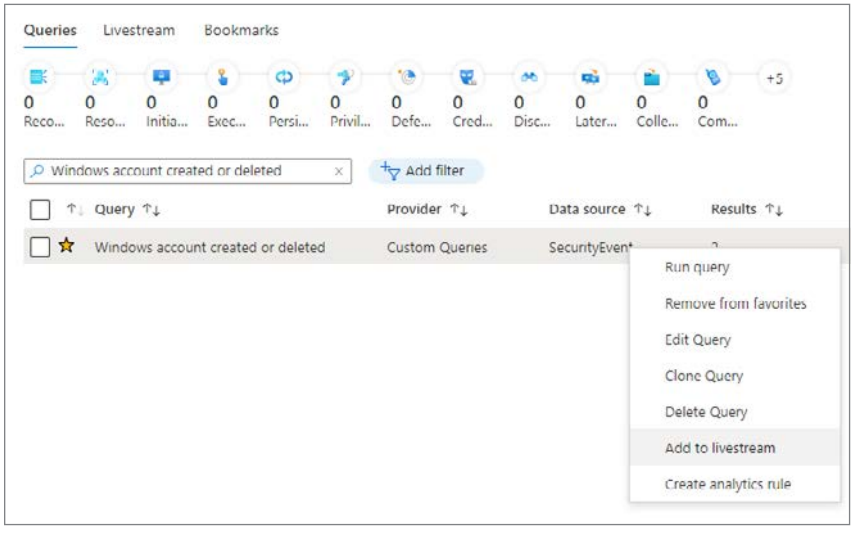


FIGURE 5-19 Add an existing hunting query to a Livestream

When the hunting query is added to a Livestream, it runs until you pause it. You can add multiple queries or add a new Livestream. When the query has a match, the column **Results** will be updated, as shown in Figure 5-20.

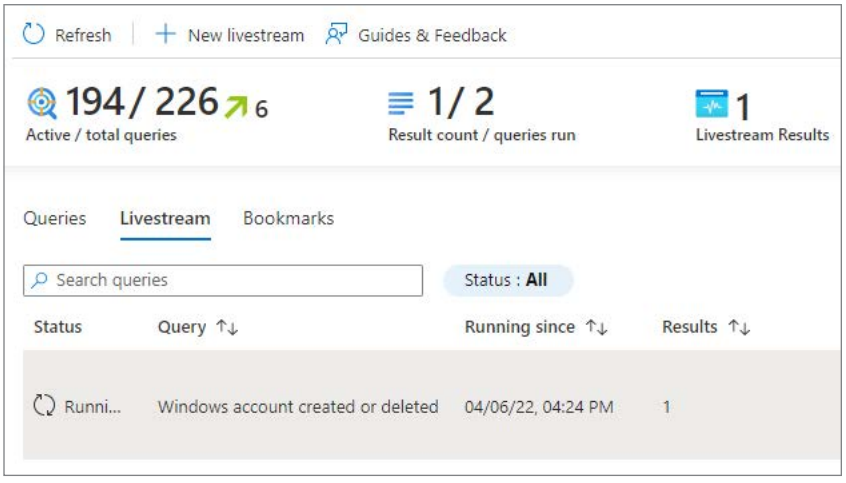


FIGURE 5-20 Livestream running

To explore the result, you can click **Open Livestream**. The Livestream session will be paused, and the initial result will be displayed, as shown in Figure 5-21.

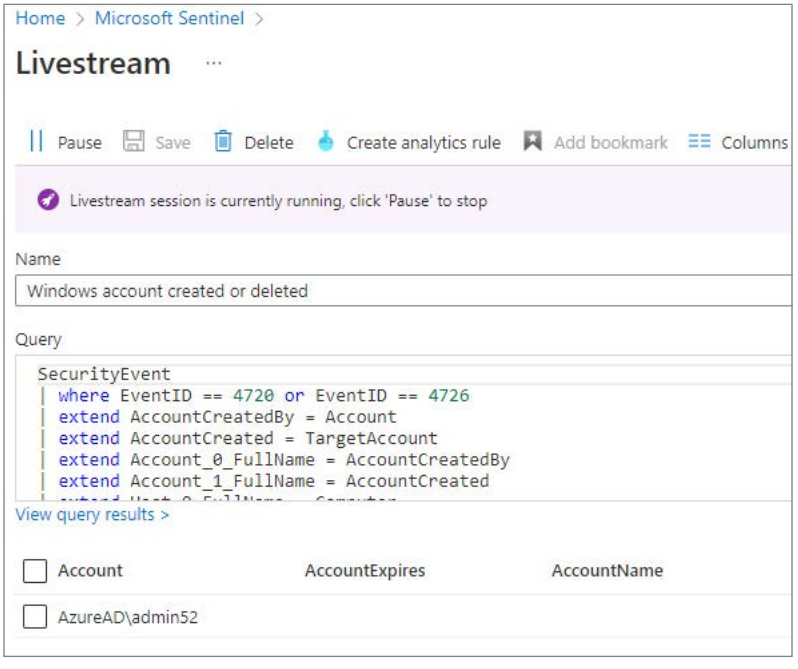


FIGURE 5-21 The result of a Livestream query, showing the account that was created during your monitoring session

Clicking **View Query Results** will open **Log Search** and display the full details of the query result, as shown in Figure 5-22.

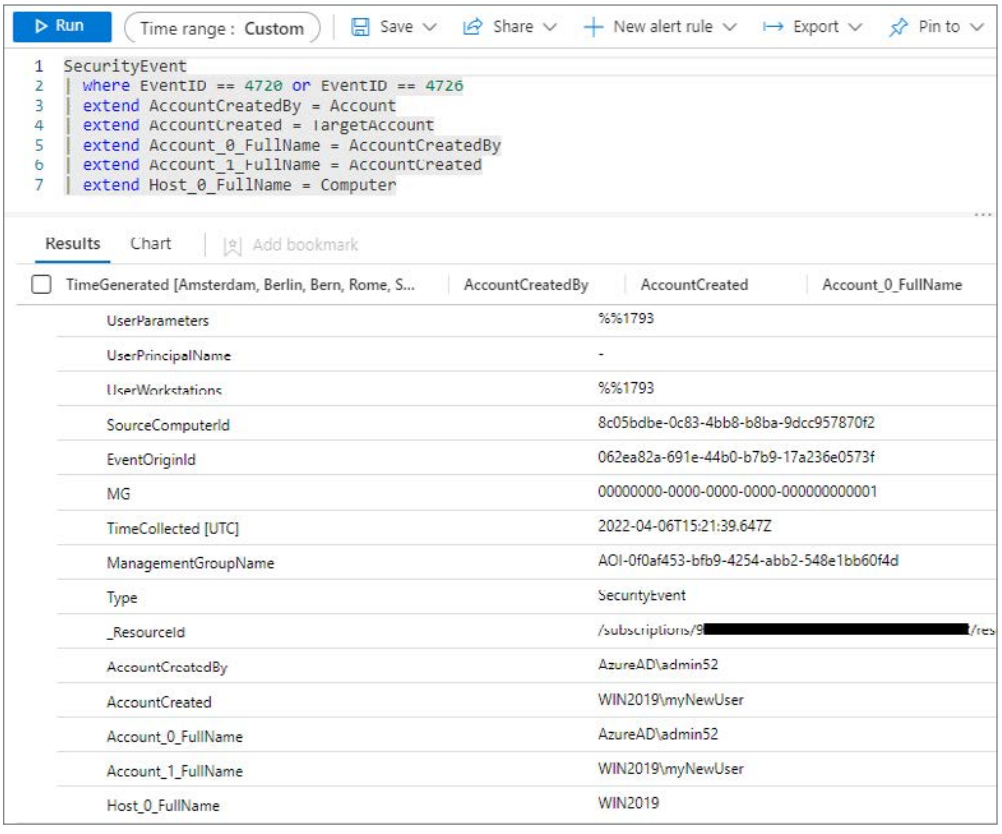


FIGURE 5-22 Livestream query results

To test-drive Livestream with this specific example, you need to have a Windows VM with the Microsoft Monitoring agent or Azure Monitor Agent installed and configured. Follow the steps below to run the Livestream example:

1. Open the **Azure portal** and sign in as a user who has Microsoft Sentinel Contributor privileges.
2. Under **Threat Management**, click **Hunting**.
3. Click in **New Query**.
4. Provide a **Name** and a **Description**.
5. In the **Custom Query** field, add the following query:

```
SecurityEvent
| where EventID == 4720 or EventID == 4726
| extend AccountCreatedBy = Account
| extend AccountCreated = TargetAccount
```

6. Map the **Account** entity with **FullName** and **AccountCreatedBy**.
7. Map the second **Account** entity with **FullName** and **AccountCreated**.
8. Map the **Host** entity with **FullName** and **Computer**.
9. Select your **Tactics & Techniques**.
10. Click **Save.**
11. Right-click your just-created query and select **Add To Livestream**.
12. Log in into your Windows VM.
13. Create a new account.
14. Go back to your Livestream.
15. Within 1 to 2 minutes, you should get a match notification from the Azure portal, and the column **Results** should be updated,
16. Click **View Results** to see the query result.

Using Livestream with Azure Key Vault honeytokens

A *honeytoken* is a fake resource that looks very attractive to an attacker but has no real value. To detect attackers in an early stage, a honeytoken can be deployed to monitor and learn from an attack or a breach. When a honeytoken has been accessed, the information can be used to track the attacker. A great example is the combination of using Livestream in combination with Azure Key Vault honeytokens.

> ***NOTE*** **At the time this book was printed, the Microsoft Sentinel Deception (Honeytokens) solution was in preview. See *https://aka.ms/SentinelHoneyTokens* for more information about using and deploying honeytokens.**

The **Honeytokens Deception** solution comes with specific detection rules, which will trigger an incident if a Key Vault key or secret is accessed. These rule queries can be used in a Livestream, too, if you suspect the compromise of a Key Vault. In the following example, a slightly modified query, based on the detection rule, is used in a Livestream session. Please note that for this to work, you need to deploy the solution first. Follow the steps below to create the Livestream instance:

1. Open the **Azure portal** and sign in as a user who has Microsoft Sentinel Contributor privileges.
2. Under **Threat management**, click **Hunting**.
3. Click **Livestream**.
4. Click **New Livestream**.
5. Provide a **Name** for your Livestream, such as `Honeytoken Livestream`.
6. Enter the query shown in Listing 5-2 in the **Query** field.

LISTING 5-2 Honeytoken Livestream Query

```
let fullAzureLogs =
    (AzureDiagnostics
    | union (print TimeGenerated=now(), ResourceType="", OperationName="",
ResourceId="", requestUri_s="",
    CallerIPAddress="", identity_claim_http_schemas_microsoft_com_identity_claims_
objectidentifier_g="",
    identity_claim_http_schemas_xmlsoap_org_ws_2005_05_identity_claims_name_s="" |
take 0))
    | where ResourceType == "VAULTS"
    | where OperationName !in ("KeyNearExpiryEventGridNotification",
"KeyExpiredEventGridNotification", "KeyNewVersionEventGridNotification",
"KeyBackup", "KeyCreate", "KeyList", "KeyRestore", "KeyResourcePut",
"KeyResourceGet", "KeyResourceList", "KeyRecover", "KeyGetDeleted")
    | project ResourceId=toupper(ResourceId), requestUri_s, OperationName,
CallerIPAddress
        , AccountObjectId=identity_claim_http_schemas_microsoft_com_identity_
claims_objectidentifier_g, AccountUPN=identity_claim_http_schemas_xmlsoap_org_
ws_2005_05_identity_claims_name_s
    | parse kind=regex flags=U requestUri_s with "https://" KVName ".vault.azure.
net/keys/" HoneyToken "[?|/]" *
    | parse kind=regex requestUri_s with "https://management.azure.com/" * "/
providers/Microsoft.KeyVault/vaults/" KVName2 "/keys/" HoneyToken2 "[?|/]" *
    | extend KVName=iff(isempty(KVName), KVName2, KVName), HoneyToken=toupper(iff(
isempty(HoneyToken), HoneyToken2, HoneyToken));
// basic alert on honeytoken access
fullAzureLogs
| join kind=inner (
    (_GetWatchlist("HoneyTokens")
    | union (print ResourceProvider="",ResourceId="",HoneyToken="",Properties="{}"
| take 0))
    | where todynamic(Properties).Type == "key"
    | extend ResourceId=toupper(ResourceId), HoneyToken=toupper(HoneyToken)
    )
    on ResourceId, HoneyToken
| summarize make_set(OperationName) by ResourceId, HoneyToken, CallerIPAddress,
AccountObjectId, AccountUPN
// enrich with Account that accessed the KV at this time if available
| join kind=leftouter (fullAzureLogs
    | distinct ResourceId, AccountObjectId, AccountUPN
    | where isnotempty(AccountObjectId))
    on ResourceId
| project ResourceId, HoneyToken, set_OperationName, CallerIPAddress,
    AccountObjectId=iff(isempty(AccountObjectId), AccountObjectId1,
AccountObjectId),
    AccountUPN=iff(isempty(AccountUPN), AccountUPN1, AccountUPN)
| extend AccountName=tostring(split(AccountUPN, "@")[0]), UPNSuffix=tostring(split
(AccountUPN, "@")[1])
| extend Severity=iff(set_OperationName contains "/decrypt?","High", "Medium")
```

7. Click **Play**, which causes the Livestream to run.
8. Open a new tab in your browser and open the Azure portal.

9. In the top search bar, search for **key vaults** and select the Key Vault where your honeytokens have been deployed.
10. Click **Keys.**
11. Click your deployed honeytoken key.
12. Go back to your Livestream session.
13. Within a couple of minutes, you should see something similar to what appears in Figures 5-23 and -5-24.

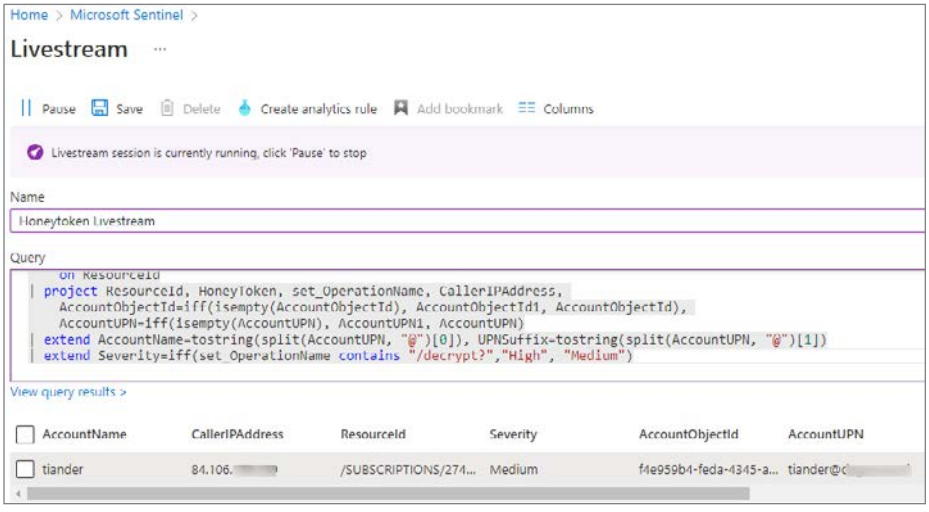


FIGURE 5-23 Livestream results based on a Key Vault key access (Livestream honeytoken 1)

HoneyToken	set_OperationName
DGSECRETKEYHT	["KeyListVersions","KeyGet"]

FIGURE 5-24 The honeytoken and operation name

NOTE **Hunting and searching for indicators of compromise can also be extended to other workspaces residing in other subscriptions or other tenants. See *https://aka.ms/crossworkspacequeries* for examples and more details. The same applies for Azure Data Explorer (ADX)—see *https://aka.ms/crossqueryadx*.**

Understanding cyberthreat intelligence

In the previous section, you learned about threat hunting, which is based on a hypothesis that a threat or a compromise is either in progress or has already occurred. The objective of Microsoft Sentinel is to get you ahead of your attackers. The incident management capability, as described in Chapter 4, "Incident management," is aimed at freeing up resources by bringing more efficiency into your SOC operations and becoming more proactive. One of those proactive approaches is integrating cyber threat intelligence (CTI) within Microsoft Sentinel. CTI is information describing known existing or potential threats to systems and users. How a SOC consumes or collects this information varies by organization.

These could be written reports, blogs, or other avenues that provide a detailed description of threat actor's motivations, infrastructure, techniques, IP addresses, domains, file hashes, and other artifacts associated with known cyberthreats. Indicators of Compromise (IoCs)—also known as *threat indicators*—are used most with Security Information and Event Management (SIEM) solutions, including Microsoft Sentinel. Threat indicators are considered *tactical* because they can be integrated with security products and can support automation at scale to detect potential threats and protect you against them. This section will cover integration with CTI, referred to in Microsoft Sentinel as *threat intelligence (TI)*.

Threat intelligence in Microsoft Sentinel

Threat intelligence (TI) in Microsoft Sentinel can be integrated through the following ways:

- **Microsoft Threat Intelligence data connector (in preview)** Based on the Microsoft Emerging Threat feed and the Bing Safety Phishing URL feed.
- **TAXII Data connector** Based on TAXII 2.0 and 2.1. This will send threat indicators from TAXII servers to Microsoft Sentinel.
- **Threat Intelligence Platforms (TIP) data connector** Integrates with Microsoft Graph Security API data sources. This will send TIP threat indicators to Microsoft Sentinel.
- **Threat detection** Using the built-in Microsoft Threat Intelligence Analytics rule and the rule templates based on TI data sources.
- **Workbooks** Providing visualization and support for hunting.

Threat indicators are sent to the Microsoft Graph Security API via your threat intelligence platforms or your custom solution. The TIP data connector then takes care of sending the threat indicators to Microsoft Sentinel. Because this requires interaction with your Azure tenant, requiring Global Administrator or Security Administrator permissions, there are specific instructions on how to set this up.

> ***NOTE*** **For guidance how to connect Threat Intelligence Platforms, see *https://aka.ms/SentinelTIPplatforms*.**

The most widely adopted industry standard for sending threat indicators is a combination of the Structured Threat Information Expression (STIX) data format and the Trusted Automated Exchange of Intelligence Information (TAXII) protocol. Microsoft Sentinel supports version 2.0 or 2.1. The TAXII data connector leverages a built-in TAXII client to import threat intelligence from TAXII 2.x servers.

> ***NOTE*** **A list of TAXII feeds can be found at *https://aka.ms/SentinelTAXIIfeeds*.**

Setting up the TAXII data connector is more straightforward compared to setting up the TIP data connector.

> ***NOTE*** **For guidance on how to connect the TAXII data connector, see *https://aka.ms/SentinelSetupTAXIIdataconnector*.**

By default, all imported threat intelligence indicators will be enriched with GeoLocation and WhoIs data. Integrating threat intelligence with Jupyter Notebooks is covered in Chapter 6, "Notebooks."

Configuring the TAXII data connector

Because setting up the TAXII data connector is more straightforward than the TIP data connector, and you can use a free open-source feed for testing, we will explore threat intelligence in Microsoft Sentinel using a feed from Anomali Limo.

Follow these steps to configure the TAXII data connector:

1. Open the **Azure portal** and sign in as a user who has Microsoft Sentinel Contributor privileges.
2. Under **Configuration**, click **Data Connectors**.
3. In the **Data Connector** search box, search for **threat intelligence**.
4. Select the **Threat Intelligence–TAXII** data connector and click **Open Connector Page**.
5. Under **Configuration**, enter the following values:
   - **Friendly Name (For Server)** **Ransomware-IP-addresses**
   - **API Root URL** *https://limo.anomali.com/api/v1/taxii2/feeds/*
   - **Collection ID** **135**
   - **Username** **guest**
   - **Password** **guest**
   - **Import Indicators** **All available**
   - **Polling Frequency** **Once a minute**

6. Your configuration should look like Figure 5-25.

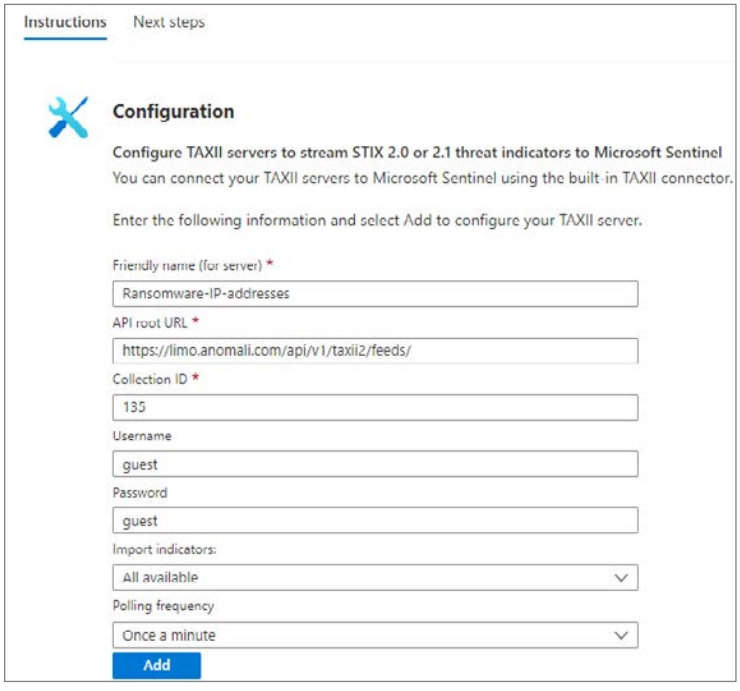


FIGURE 5-25 The TAXII data connector configuration

7. Click **Add**.
8. In a couple of minutes, **Ransomware-IP-addresses** threat indicators should appear in the **Threat Intelligence** blade, as shown in Figure 5-26.

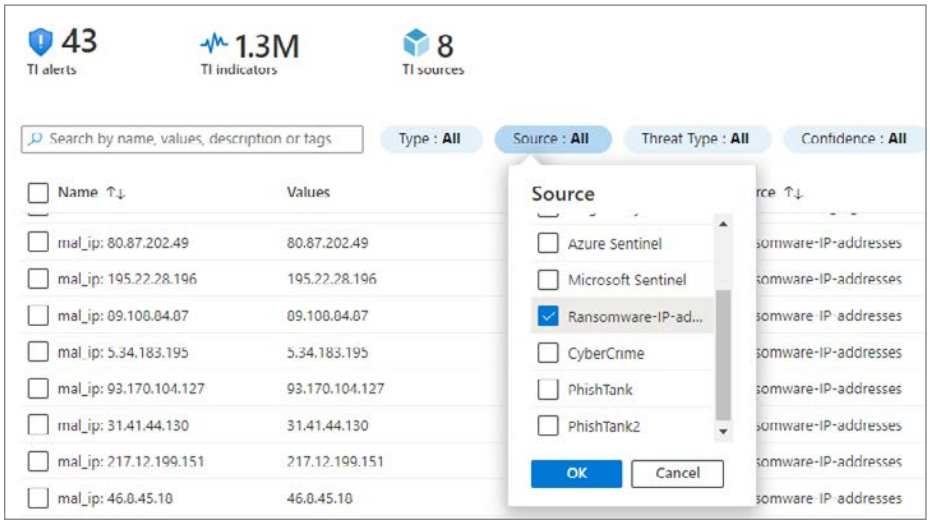


FIGURE 5-26 Ransomware IoCs

9. Under **Threat Management**, click **Threat Intelligence**.
10. Click **Source**, select **Ransomware-IP-Addresses**, and click **OK**.
11. Explore the list of IoCs brought in by the Anomali Limo feed.

Enabling the threat intelligence rules

Now that you have configured the TAXII data connector, it is important to enable the threat intelligence rule templates. This ensures the creation of a security incident if an IoC match is found.

Follow the steps below to enable those rules:

1. Open the **Azure portal** and sign in as a user who has Microsoft Sentinel Contributor privileges.
2. Under **Configuration**, click **Analytics**.
3. Click **Rule Templates**.
4. Click **Add Filter**.
5. In the **Filter** selection field, select **Data Sources**.
6. In the **Value** dropdown, deselect **Select All**.
7. Scroll down and select **Threat Intelligence–TAXII**. If you have configured the TIP data connector, select **Threat Intelligence Platforms**.
8. Click **Apply**.
9. Click a rule to test, such as **TI Map IP Entity To VMConnection**.
10. Observe the required Data Sources in the details pane at the right. This specific rule is dependent on the VM Insights solution from Azure Monitor**.**

> ***TIP*** If you would like to configure VM Insights, see *https://aka.ms/SentinelConfigureVMinsights*.

11. Click **Create Rule**.
12. On the **Set Rule Logic** tab, under **Query Scheduling**, change **Run Query Every** to **5 Minutes**. Set **Lookup Data From The Last** to **5 Minutes**.

> ***NOTE*** This value is not appropriate for production environments and should only be used for testing purposes.

13. Continue the wizard and accept the default values.
14. Click **Create**.

The rule that you have just created will run every 5 minutes, will look up data from the last 5 minutes, and should be deactivated after testing.

> ***NOTE*** **The TimeGenerated field for Indicators of Compromise (IoCs) is refreshed every 14 days to make them available for analytic rules. This only applies to active IoCs with an expiration date of today or later.**

Creating a custom threat indicator

Before you can test your TAXII data connector configuration and the TI alert rule that you have just created, you will create a custom threat indicator so that you can test the creation of a TI based incident.

> ***NOTE*** **The following exercise requires a VM with the Microsoft Monitoring Agent or Azure Monitor Agent installed and the Azure Monitor VM Insights solution enabled. To test, you need to know the IP address that you use to connect to your VM.**

Follow the steps below to create a custom threat indicator:

1. Open the **Azure portal** and sign in as a user who has Microsoft Sentinel Contributor privileges.
2. Under **Threat Management**, click **Threat Intelligence**.
3. In the upper pane, click **Add New**.
4. This opens a new pane on the right.
5. Under **Types**, select **ipv4-addr**.
6. Enter the IP address that you will use to connect to your test VM.

7. Fill in the remainder of the fields with values of your choice, as shown in Figure 5-27.

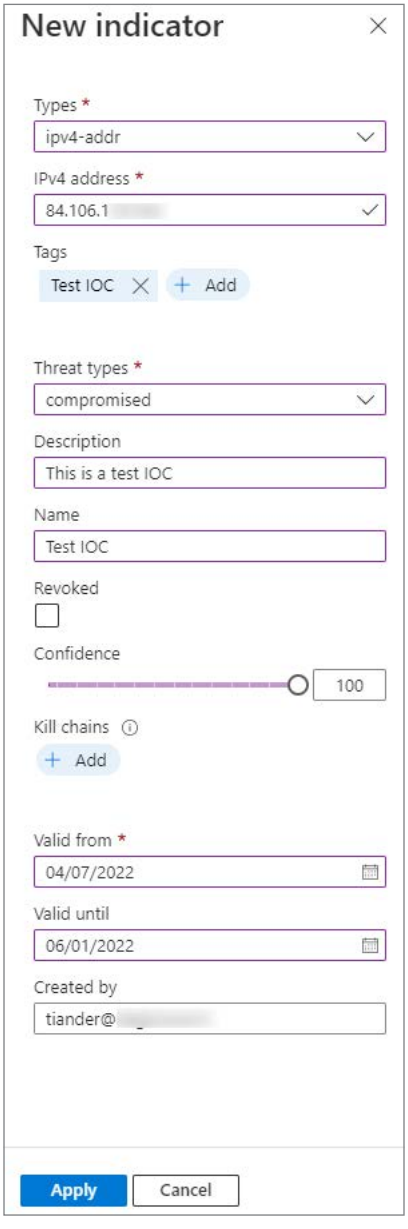


FIGURE 5-27 Creating a custom threat indicator

8. Click **Apply**.
9. Log in to your test VM.

10. After approximately 5 minutes, a new incident should be created, as shown in Figure 5-28.

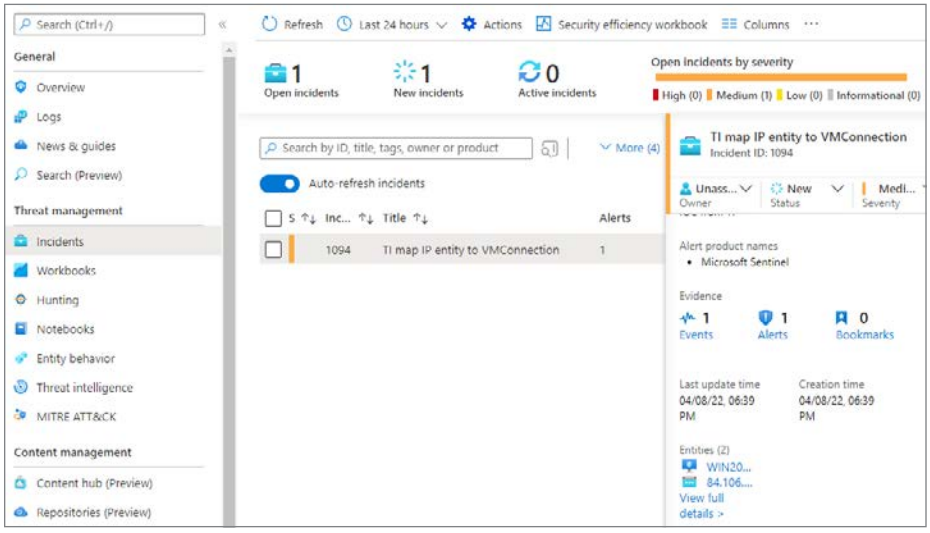


FIGURE 5-28 New IoC incident

11. Log out of your VM and disable the rule that you have created to generate this incident.

From here, you can start triaging and hunting as you have learned in the previous chapters. Threat indicators are stored in the `ThreatIntelligenceIndicator` table. Using Kusto Query Language (KQL) and Log Search, you can explore more information about threat indicators, as shown in Figure 5-29.

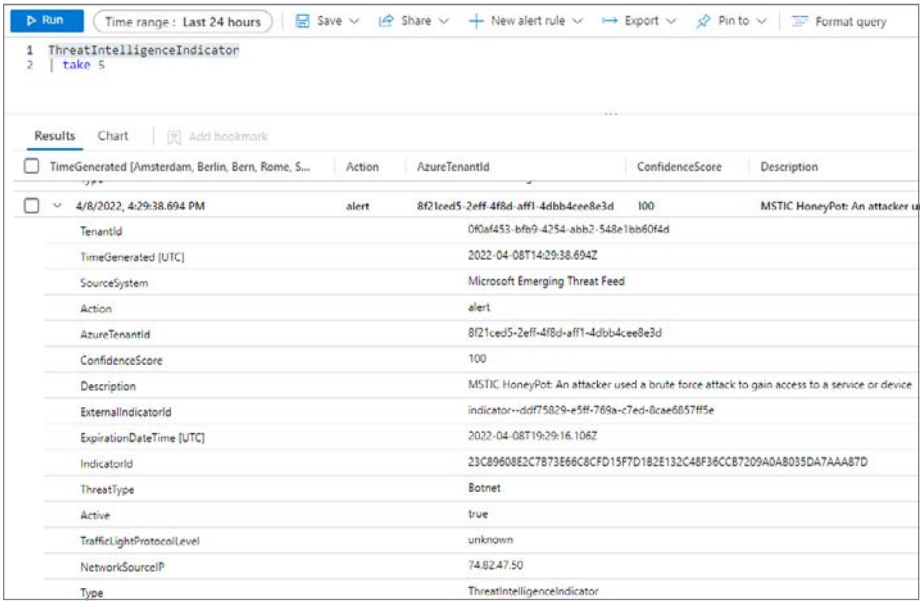


FIGURE 5-29 The ThreatIntelligenceIndicator table

Interactive TI and hunting dashboards

By now, you have probably learned that threat hunting comes in many forms and flavors. Using interactive dashboards and being able to visualize data in different ways are important components of every SOC operations and hunting team. In Microsoft Sentinel, the threat intelligence dashboard is known as threat intelligence workbooks and provides an additional hunting dimension.

You are going to explore two workbooks—one provided by Microsoft, and one provided by the Microsoft Sentinel open-source community. Follow the steps below to start exploring:

1. Open the **Azure portal** and sign in as a user who has Microsoft Sentinel Reader privileges.
2. Under **Threat Management**, click **Workbooks**.
3. Click the **Templates** tab.
4. In the **Search** field, type **Threat**, which will return a couple of more interesting results for you to explore.
5. Click **Threat Intelligence**; this is the out-of-the-box standard workbook.
6. In the right pane, click **Save** and select the location for the workbook to be saved in. If you have already saved this workbook, click **View Saved Workbook**.
7. Explore the different tabs, as shown in Figures 5-30 and 5-31.

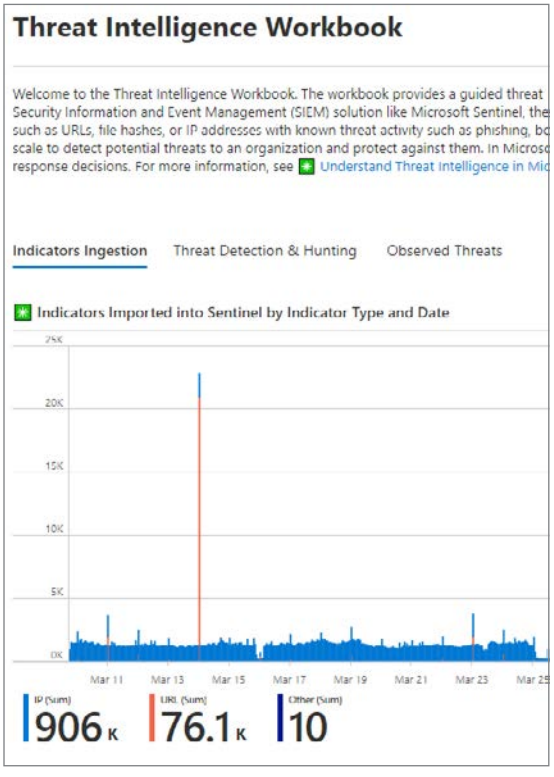


FIGURE 5-30 The Threat Intelligence Workbook, showing imported indicators

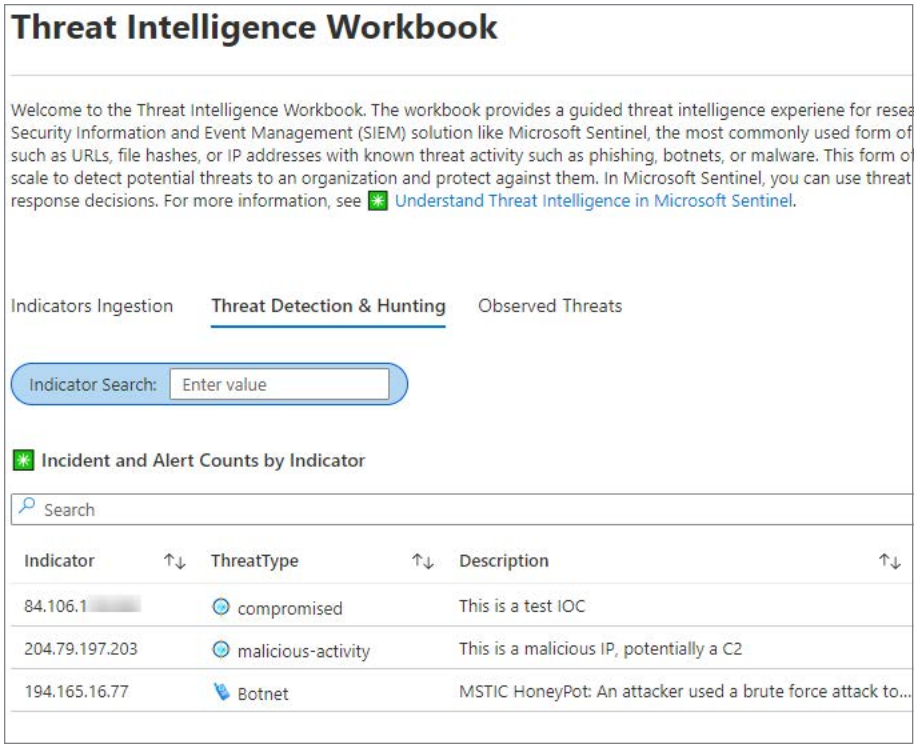


FIGURE 5-31 The Threat Intelligence Workbook

8. To explore a more interactive workbook, authored by the community, go back to the **Workbooks** blade.
9. Click the **Templates** tab.
10. In the **Search** field, type **Investigation Insights**.
11. Click **Save** or **View Saved Workbook**.
12. Select your **Subscription**, **Workspace**, and **TimeRange**.
13. Under **Investigate By**, click **Entity**.

14. This shows Entity Insights, like Investigate IP Address and Investigate Host, as shown in Figures 5-32 and 5-33.

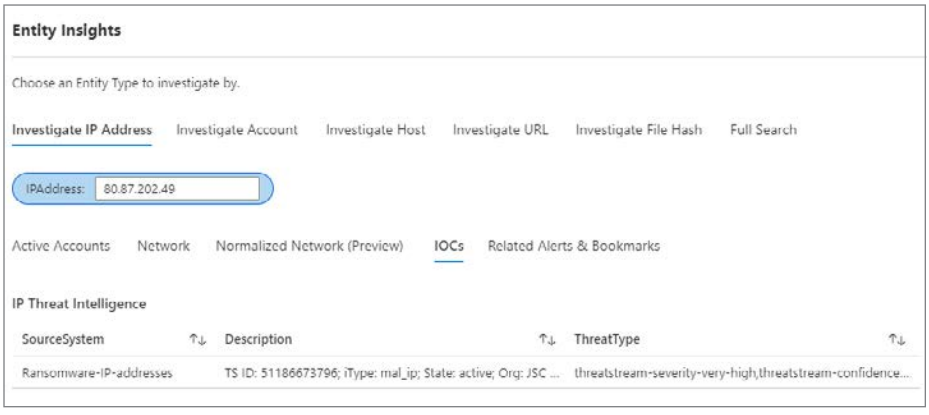


FIGURE 5-32 The Investigation Insights Workbook, showing entities to investigate

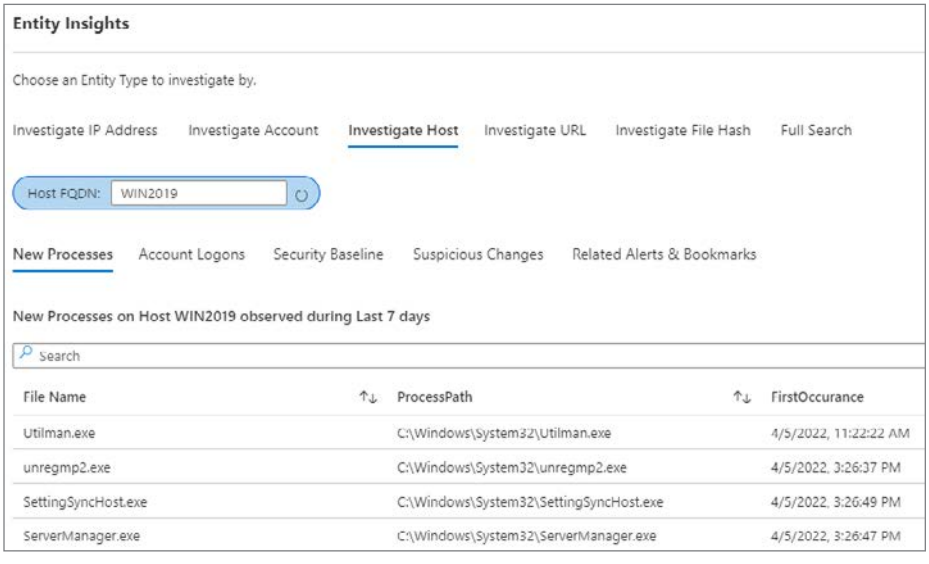


FIGURE 5-33 The Investigation Insights Workbook, showing host information

Chapter 6

Notebooks

In Chapter 5, you learned about hunting in Microsoft Sentinel. This chapter is going to cover another hunting option, using Notebooks. More precisely, Jupyter Notebooks. Besides hunting, you will explore other options, like enrichment and extending your incident triage experience using Notebooks.

A lot has already been written about Jupyter Notebooks, hereafter referenced as Notebooks. In summary, Jupyter is an interactive development and data manipulation environment. A Notebook is generally referenced as a document that integrates live code, equations, computational output, visualizations, and other multimedia resources, along with explanatory text in a single document.

The intent of this chapter is to provide you with practical information and guidance to start exploring Notebooks.

Understanding Microsoft Sentinel Notebooks

When Notebooks were introduced in Microsoft Sentinel, it initially caused some confusion related to their positioning and purpose—specifically, related to the concepts of *Workbooks* and *Playbooks*. If you are new to Microsoft Sentinel and are being introduced to Workbooks, Playbooks, and Notebooks, it generally creates confusion.

TIP **More information about the history of Notebooks, including documentation, can be found here: *https://jupyter.org***

Table 6-1 clarifies and positions the three different features, although Notebooks are being used more and more in the incident triage phase:

TABLE 6-1 Notebooks compared to Workbooks and Playbooks

	PLAYBOOKS	WORKBOOKS	NOTEBOOKS
Roles	■ SOC engineers ■ Analysts	■ SOC engineers ■ Analysts ■ SOC managers	■ Threat hunters ■ Tier 2-3 analysts ■ Data scientists ■ Security researchers
Usage	■ Automation of repeatable tasks ■ Ingestion - bring in external data ■ Enrichment (TI, GeoIP, lookups, etc.) ■ Investigation ■ Remediation	■ Visualization ■ Reporting	■ Querying Microsoft Sentinel & external data ■ Enrichment (TI, GeoIP, WhoIs lookups, etc.) ■ Investigation ■ Visualization ■ Hunting ■ Machine Learning & big data analytics
Pros	■ Best for single, repeatable tasks ■ No coding knowledge required	■ Best for viewing Microsoft Sentinel data ■ No coding knowledge required	■ Best for more complex chain of repeatable tasks ■ Ad-hoc, more procedural control – easy to pivot due to the interactive characteristics and the use of Python, a procedural language ■ Rich Python libraries for data manipulation & visualization options ■ Machine Learning & custom analysis ■ Easy to document & share analysis evidence
Cons	■ Not suitable for ad-hoc & complex chain of tasks ■ Not great for documenting & sharing evidence	■ Limited external data integration options	■ Higher learning curve, potentially requires Python knowledge

There are multiple options to run a Notebook, like running your own Jupyter server, running it on JupyterHub, or running it in a Docker container. In this chapter, you will explore running Notebooks in Microsoft Sentinel.

Referencing *Microsoft Sentinel Notebooks*, instead of just *Notebooks*, should be considered as running a Notebook within Microsoft Sentinel's integrated environment.

NOTE **Microsoft Sentinel Notebooks can be run in any Jupyter-compatible environment.**

To be able to run Notebooks within Microsoft Sentinel, you have the option to run Notebooks on the Azure Machine Learning (AML) platform or Azure Synapse Analytics (in preview at the time of writing this chapter). The latter is more suitable for large-scale data processing. Because this is not the case for the samples we explore in this chapter, we will configure an AML environment in the next section.

Configuring an AML workspace and compute

To start exploring Notebooks in Microsoft Sentinel, you will first set up an AML workspace and create a compute resource to run your Notebooks. Please note that you will need AML contributor permissions to follow along.

> ***NOTE*** **For more information on the required AML permissions and roles, see *https://aka.ms/AMLpermissions*.**

Follow these steps to start:

1. Open the **Azure portal** and sign in as a user who has Microsoft Sentinel contributor and AML contributor privileges.
2. Navigate to the **Microsoft Sentinel** page.
3. Under **Threat management**, click **Notebooks**.
4. In the top-middle pane, under **Configure Azure Machine Learning**, click **Create New Azure ML Workspace**, as shown in Figure 6-1.

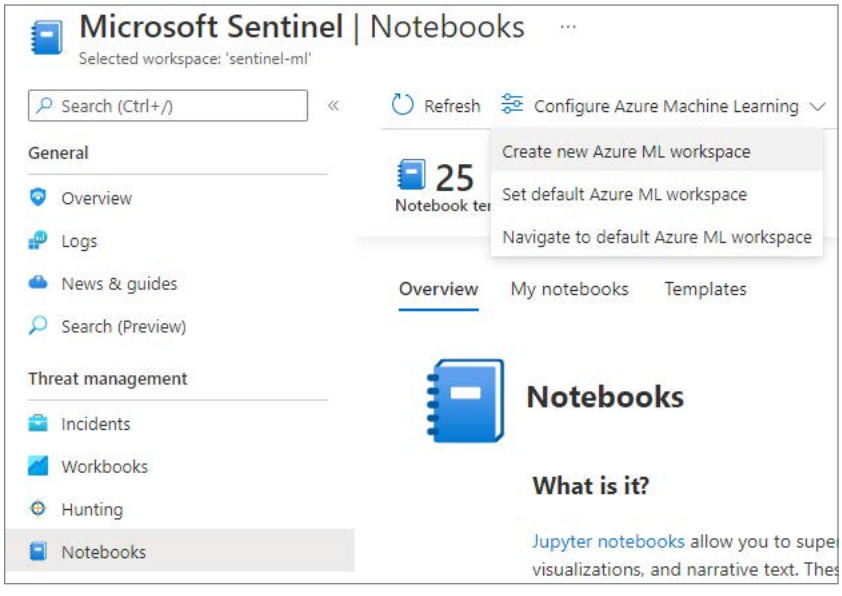


FIGURE 6-1 Create a new Azure ML workspace

5. In the create wizard that follows, provide your custom values, as shown in Figure 6-2. (Some of them are auto-populated, but that can be changed.)

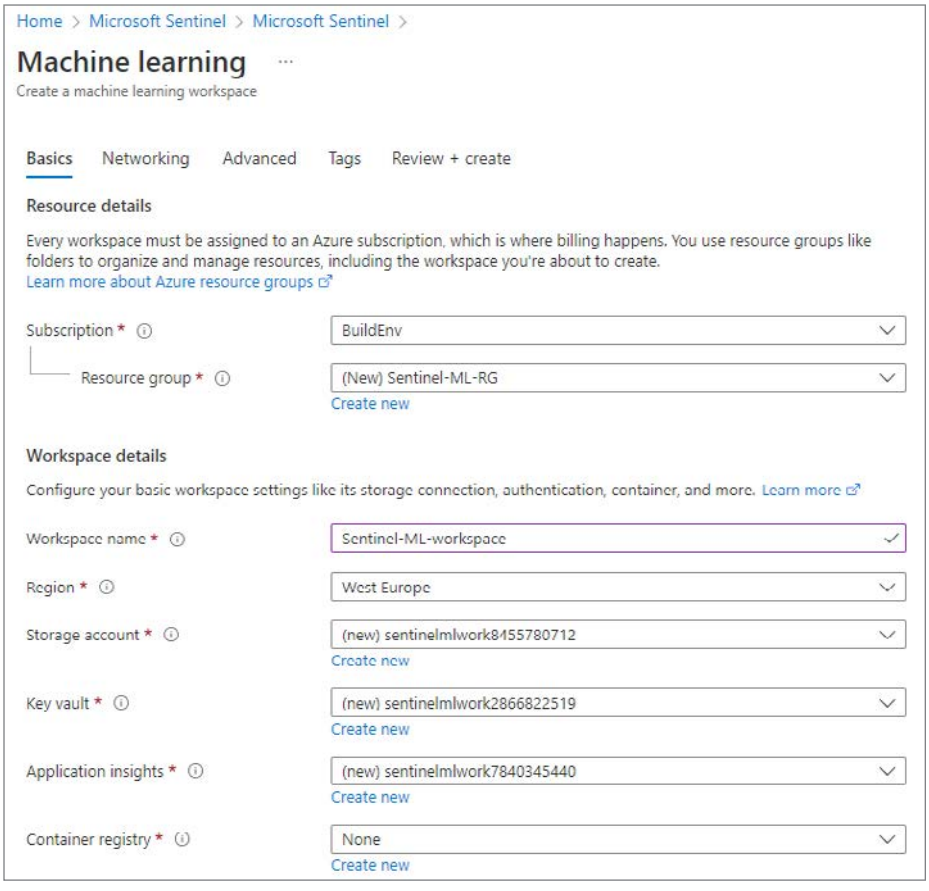


FIGURE 6-2 Create A Machine Learning Workspace wizard

NOTE **As a security best practice, a Key Vault should be used to store sensitive information, like your workspaceId, workspaceKey, an API key, or any information that needs to be protected. How to do this will be covered in one of the sample Notebooks. Values retrieved from a Key Vault will not be stored in your Notebook.**

6. Click **Next**, and on the **Networking** tab, select your preferred endpoint configuration, as shown in Figure 6-3.

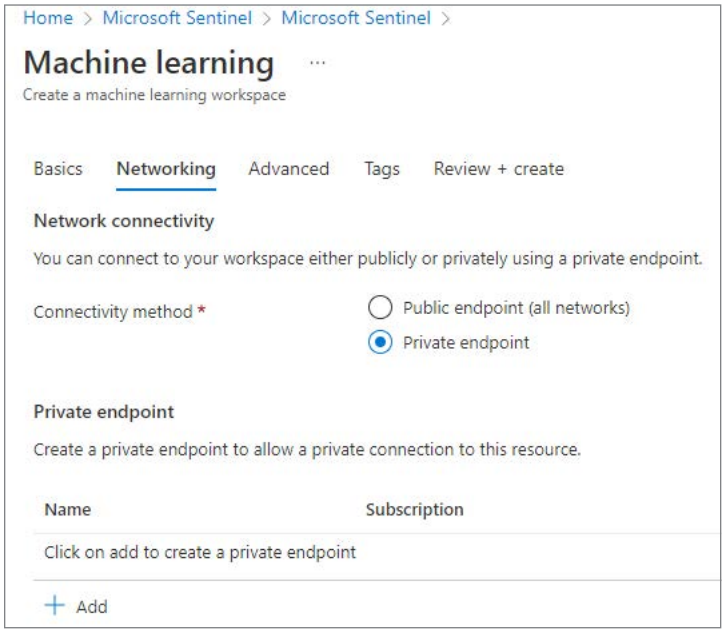


FIGURE 6-3 The Networking tab

NOTE **A public endpoint enables data access to your workspace from outside the virtual network and is publicly routable, whereas a private endpoint is a network interface that uses a private IP address from your virtual network and connects you privately and securely.**

7. Optionally, you can configure the remainder of the wizard and click the **Create** button, as shown in Figure 6-4.

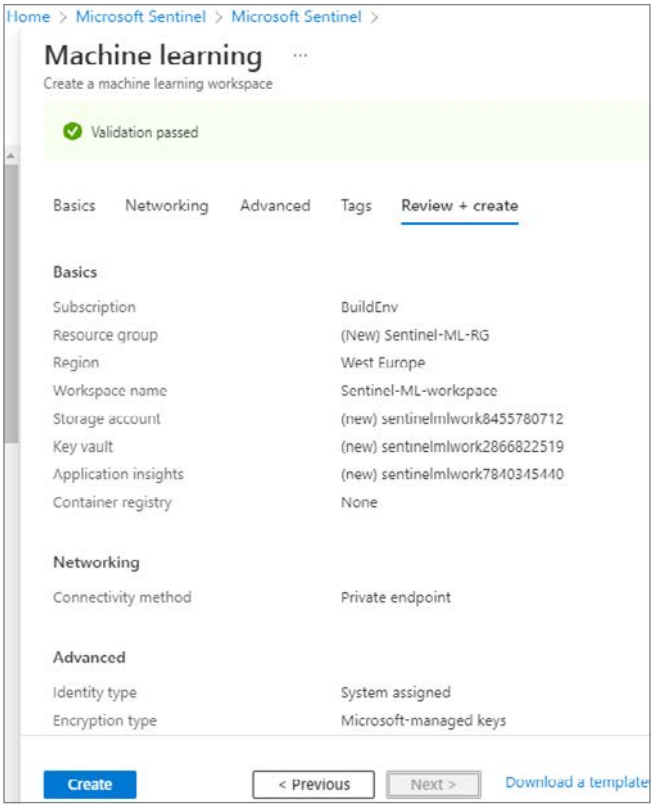


FIGURE 6-4 Create A Machine Learning Workspace wizard summary

8. When the deployment is done, go back to the **Notebooks** page by clicking your browser's back button
9. Click the **Templates** tab, as shown in Figure 6-5.

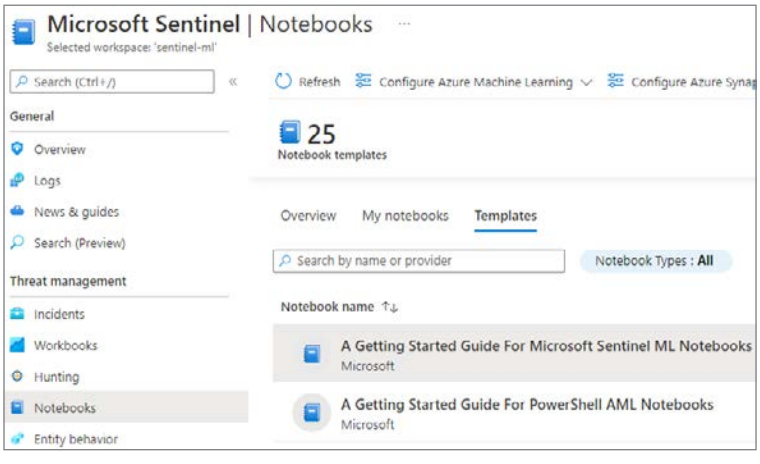


FIGURE 6-5 Notebook templates

10. This tab shows the Notebook templates, based on a selection of the GitHub Notebooks repository. Click once on the **A Getting Started Guide For Microsoft Sentinel ML Notebooks** option. In the right pane, click the **Create From Template** button, as shown in Figure 6-6.

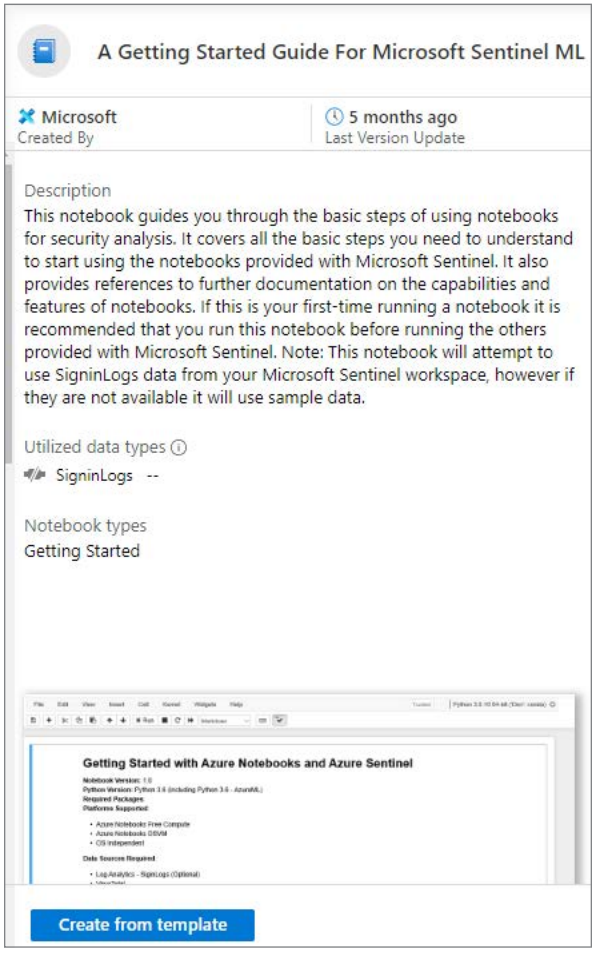


FIGURE 6-6 Create a Notebook from a template

11. Select your created AML workspace and click the **Save** button, as shown in Figure 6-7.

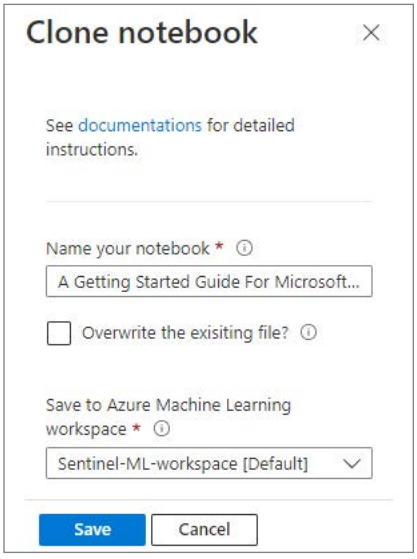


FIGURE 6-7 Save the cloned Notebook

12. Now that you have cloned and saved the Notebook, click **Launch Notebook**, which will open the **Microsoft Azure Machine Learning Studio** page.
13. Before you can run a Notebook, you need to create a **Compute Instance**, which will run your Notebook.
14. In the upper-right pane, click the **+** sign, as shown in Figure 6-8.

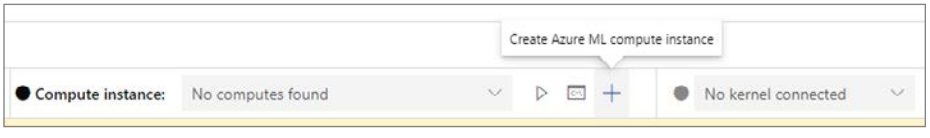


FIGURE 6-8 Create a compute instance

15. On the **Configure Required Settings** page, provide your **Compute Name**, your **Virtual Machine Type,** and **Virtual Machine Size**, as shown in Figure 6-9.

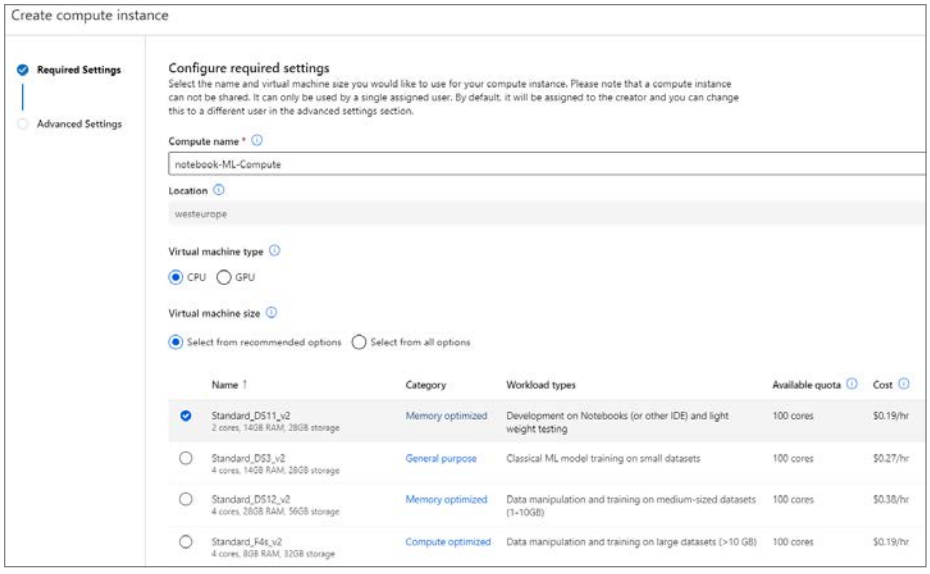


FIGURE 6-9 Create Compute Instance wizard

16. The **Advanced Settings** show options to provide a **Startup And Shutdown Schedule** (recommended) and additional options, as shown in Figure 6-10.

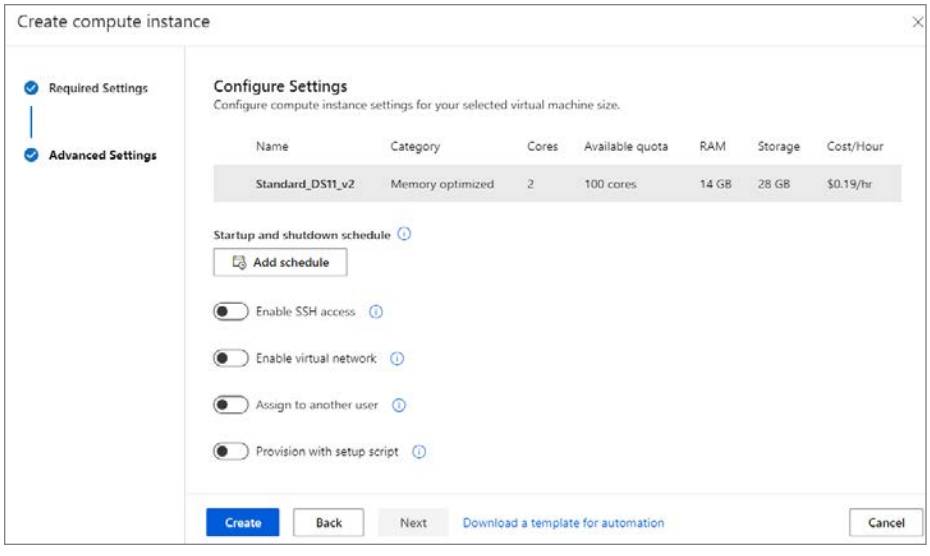


FIGURE 6-10 Create a compute instance wizard advanced settings

TIP **For testing purposes, it is recommended that you select the Standard_DS11_v2 VM size and configure a startup and shutdown schedule to be cost-effective.**

17. Click the **Create** button to create your compute instance.
18. The compute instance status will change to **Creating**, as shown in Figure 6-11.

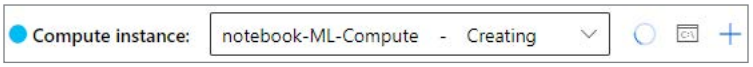


FIGURE 6-11 Compute instance creation in progress

19. Wait until the creation of your compute is complete and the status is **Running.**
20. In the yellow notification bar, click **Authenticate**, as shown in Figure 6-12, which allows you to use Azure SDK.

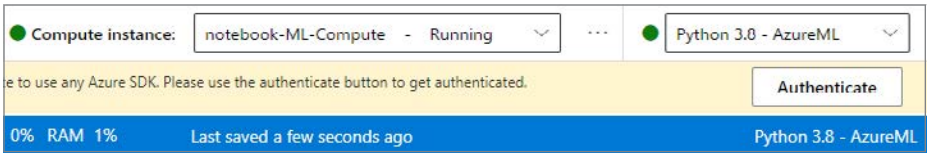


FIGURE 6-12 Authenticate the compute to use any Azure SDK

This completes the setup and configuration of the AML workspace and the required compute. You are almost ready to explore Notebooks!

Configuration steps to interact with your Microsoft Sentinel workspace

For your Notebooks to interact with Microsoft Sentinel, the last preparation step is to create a configuration file, which contains at least your workspace key and workspace identifier. `MSTICpy`, which will be covered later in this chapter, is a Python package developed by Microsoft's Threat Intelligence Center (MSTIC) security analysts and engineers, which will make this step very easy.

In the previous steps, you have cloned the Getting Started with Azure ML Notebooks and Microsoft Sentinel Notebook. It is highly recommended that you go through this Notebook, which is intended to give you an interactive introduction. Completing this Notebook will jumpstart your learning experience. It also contains the required steps to configure your connection to Microsoft Sentinel.

> ***TIP*** **You can run a cell by either clicking the Run icon, which is shown on the left side of the cell if you hover over it or by selecting Shift+Enter.**

As stated in the Notebook, it is important to explore and run each cell in sequence because certain cells have dependencies. After you have completed the previous steps, you will notice that the `msticpyconfig.yaml` file has been generated and now contains configuration information that was automatically populated based on your current Azure connection, as shown in Figure 6-13.

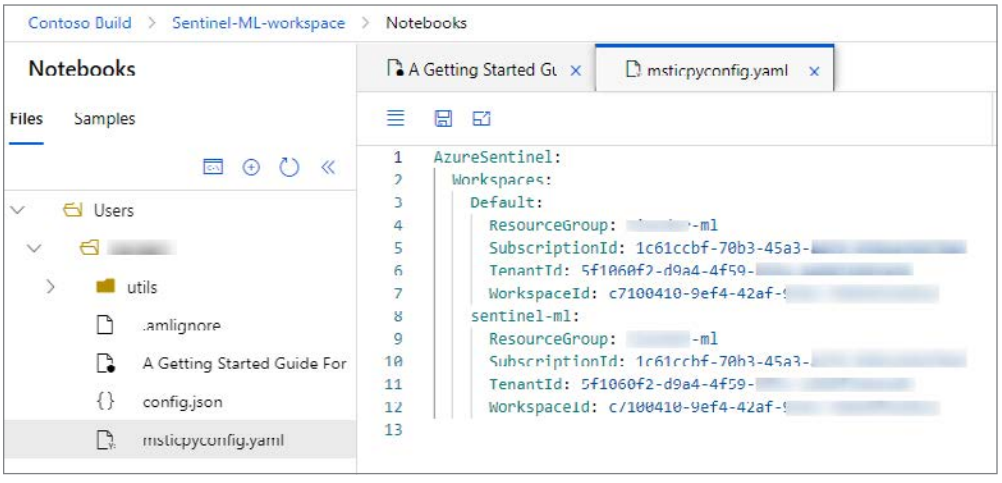


FIGURE 6-13 The msticpyconfig.yaml file

> ***TIP*** **You might need to refresh the file listing for your `msticpyconfig.yaml` file to show; do this by clicking the refresh icon in the left panel.**

To be able to query your Microsoft Sentinel workspace, you use the `QueryProvider` object. This step is listed in "A Getting Started Guide For Microsoft Sentinel ML Notebooks," in section 3.3, "Load a QueryProvider for Microsoft Sentinel." Upon running this cell, you will see this confirmation: `...Loading KqlMagic extension...done`. The next section, "Authenticate to the Microsoft Sentinel workspace," will perform the actual authentication against Microsoft Sentinel, as shown in Figure 6-14.

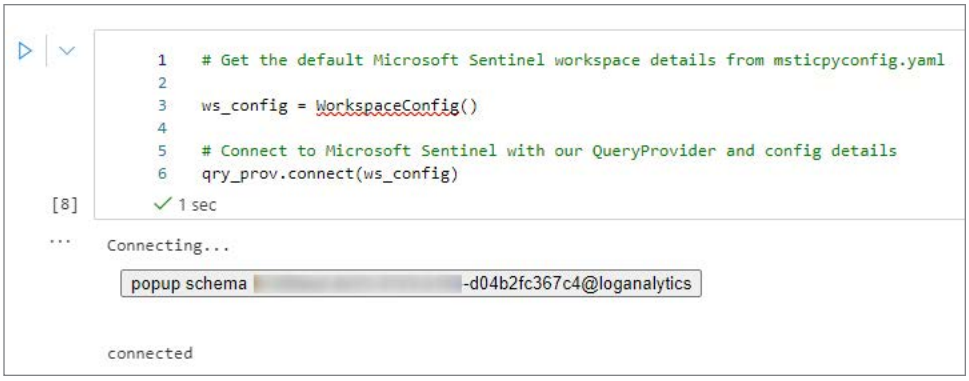


FIGURE 6-14 Authenticating to Microsoft Sentinel

If your token has expired, you will be prompted to authenticate using a device login; if not, the authentication will be seamless.

> ***TIP*** **If you load a new Notebook, you will be prompted again to log in using device authentication. If you want to log in only once, you can create a new cell and run** `!az login`**.**

Running a cell sometimes installs new packages, which requires a kernel restart for the changes to be applied. If running a cell fails because of an updated package, you might want to restart the kernel and try again.

The MSTICpy library

As briefly mentioned earlier in this chapter, the `MSTICpy` library, developed by MSTIC's security analysts and engineers, provides a rich set of Python tools that are intended to be used for security investigations and hunting.

> ***TIP*** **For more information on** `MSTICpy`**, see *https://aka.ms/MSTICpydocs*.**

Because several sample Notebooks provide an excellent overview and examples on how to use `MSTICpy`, this section will cover a couple of practical examples.

To quickly test several Notebooks from the Microsoft Sentinel GitHub repo, you can easily clone the repository, as shown below:

1. Ensure that your ML compute is running, and you are in the Microsoft Azure Learning Studio environment, as you have done in the previous steps

2. Click the **Open Terminal** icon, as shown in Figure 6-15.

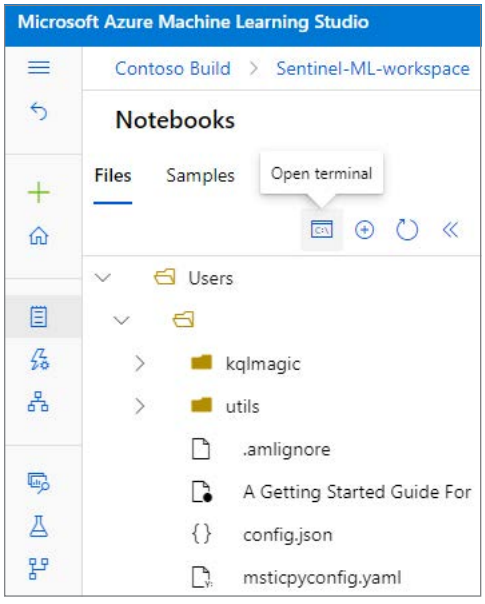


FIGURE 6-15 Open Terminal

3. This will open a terminal in a new tab. From here, you can create a new folder, or you can browse to an existing one, where you can store the Notebooks you are about to clone.
4. Optionally, you can create a folder, navigate to that folder, and copy and paste the following code to clone the GitHub Notebooks, as shown in Figure 6-16:

```
git clone https://github.com/Azure/Azure-Sentinel-Notebooks.git
```

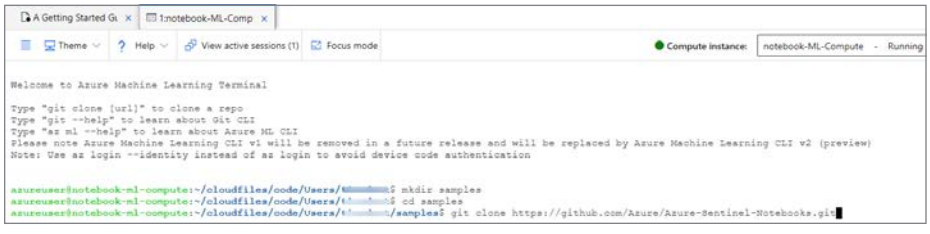


FIGURE 6-16 The git clone command

5. Press **Enter** to execute the command, which should show output similar to that shown in Figure 6-17.

```
Cloning into 'Azure-Sentinel-Notebooks'...
remote: Enumerating objects: 2227, done.
remote: Counting objects: 100% (355/355), done.
remote: Compressing objects: 100% (96/96), done.
remote: Total 2227 (delta 321), reused 259 (delta 259), pack-reused 1872
Receiving objects: 100% (2227/2227), 28.74 MiB | 11.12 MiB/s, done.
Resolving deltas: 100% (1417/1417), done.
Updating files: 100% (219/219), done.
azureuser@notebook-ml-compute:~/cloudfiles/code/Users/[redacted]t/samples$
```

FIGURE 6-17 Output of the git clone command

6. Refresh the folder view in the left pane, which will show your newly created folder and cloned Notebooks, as shown in Figure 6-18.

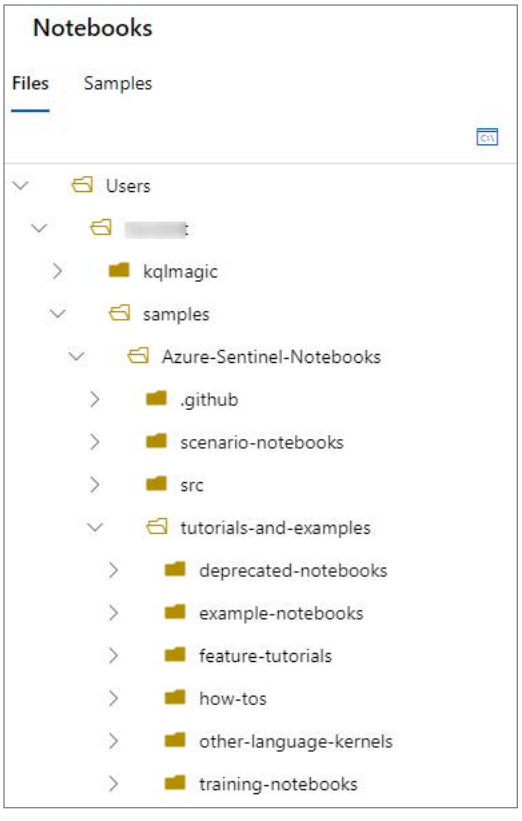


FIGURE 6-18 The Samples folder with the cloned Notebooks

NOTE **The `git clone` command that you just used is an easy way to clone the Notebooks, but it is not how you would typically maintain a synchronized GitHub repo for CI/CD purposes.**

Hunting and enrichment examples

This section provides a couple of Notebook-hunting examples to get you started. The following is based on the Notebook examples that you have cloned in the previous section and assumes that you have run through and completed the "A Getting Started Guide For Microsoft Sentinel ML Notebooks" Notebook.

Sign-ins that did not pass the MFA challenge

The `MSTICpy` library contains several useful pre-built queries that you can use. To see a list of queries and the syntax you can use, enter this query:

```
qry_prov.browse_queries()
```

When you use this query in a new cell and run it, it will show you the output, as shown in Figure 6-19, including the syntax and examples.

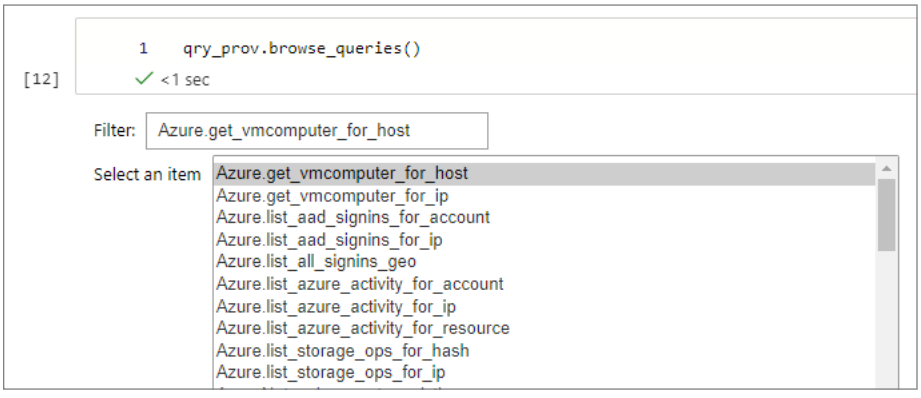


FIGURE 6-19 MSTICpy built-in query listing

Explore the `Azure.list_allsignins_geo` query because you will use it in the next example. In the example below, you are going to do the following:

1. Run a query against the `SiginLogs` table.
2. Apply a filter for the `ResultDescription` column.
3. Look for values that contain the `User did not pass the MFA challenge` string.
4. Look for unique `IP addresses` in the returned results.
5. Create a `Threat Intelligence lookup object`.

Check the IP addresses against `VirusTotal`.To follow along, create a new cell in the "A Getting Started Guide For Microsoft Sentinel ML Notebooks" Notebook. Ensure that you are authenticated and that you have executed all cells and their dependencies.

1. You are going to use the Python `Pandas` library, create a new cell, copy and paste the line below and execute the cell by pressing Shift+Enter:

```
import pandas as pd
```

2. Create a new cell and paste in the following:

```
#this query comes from the query provider:
signin_df = qry_prov.Azure.list_all_signins_geo()

#show the first 5 results:
signin_df.head()
```

3. Execute the cell.

4. You should see a similar output as shown in Figure 6-20.

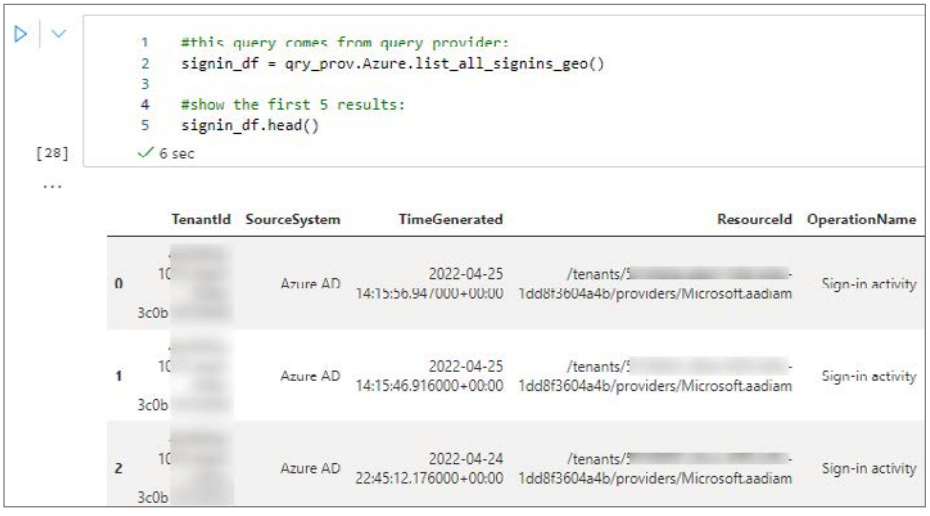


FIGURE 6-20 Query the signinlog table. (Some information has been intentionally blurred.)

5. Create a new cell and copy and paste the following into it:

```
#create a new pandas dataframe and filter for a specific string:
signin_mfa_df = signin_df[signin_df["ResultDescription"].str.contains("User did
not pass the MFA challenge")]

#look for unique IP addresses
signin_mfa_df = (pd.unique(signin_mfa_df['IPAddress']))

#show the array of unique IP addresses:
signin_mfa_df
```

6. That should return similar results as shown in Figure 6-21.

```
#create a new pandas dataframe and filter for a specific string:
signin_mfa_df = signin_df[signin_df["ResultDescription"].str.contains("User did not pass the MFA challenge")]

#look for unique IP addresses
signin_mfa_df = (pd.unique(signin_mfa_df['IPAddress']))

#show the array of unique IP addresses:
signin_mfa_df


[34]  ✓ <1 sec

...  array(['105.161.22.56', '46.210.49.160', '88.3.168.167', '130.44.170.146',
       '157.48.68.172', '167.220.205.116', '98.16.47.97'], dtype=object)
```

FIGURE 6-21 Results of the signinlog table with a filter condition

7. Now that you have values stored based on your filter condition, you are going to create a threat intelligence lookup object. Copy and paste the following into a new cell:

```
#create a Threat Intelligence object:
ti = TILookup()

#use the IPAddress column values and check these against VirusTotal:
ti.lookup_iocs(signin_mfa_df, obs_col="IPAddress", providers=["VirusTotal"])
```

8. That should return similar results as shown in Figure 6-22.

```
#create a Threat Intelligence object:
ti = TILookup()

#use the IPAddress column values and check these against VirusTotal:
ti.lookup_iocs(signin_mfa_df, obs_col="IPAddress", providers=["VirusTotal"])

✓ 1 sec
```

	Ioc	IocType	Safeloc	QuerySubtype	Provider	Result	Severity	Details
0	105.161.22.56	ipv4	105.161.22.56	None	VirusTotal	True	information	{'verbose_msg': 'Missing IP address', 'response_code': 0, 'positives': 0}
1	46.210.49.160	ipv4	46.210.49.160	None	VirusTotal	True	information	{'verbose_msg': 'Missing IP address', 'response_code': 0, 'positives': 0}
2	88.3.168.167	ipv4	88.3.168.167	None	VirusTotal	True	information	{'verbose_msg': 'IP address in dataset', 'response_code': 1, 'positives': 0, 'detected_urls': []}
3	130.44.170.146	ipv4	130.44.170.146	None	VirusTotal	True	information	{'verbose_msg': 'Missing IP address', 'response_code': 0, 'positives': 0}

FIGURE 6-22 Results of a VirusTotal lookup

9. If you want to filter on specific threat intelligence columns, such as a `Severity` value of warning or `Result == True`, you can adapt the query by using something like this:

```
#create a Threat Intelligence object:
ti = TILookup()

#use the IPAddress column values and check these against VirusTotal:
ti_warning = ti.lookup_iocs(signin_mfa_df, obs_col="IPAddress",
providers=["VirusTotal"])

#filter out the column "Result" for a value of "True"
ti_warning[ti_warning["Result"]==True]
```

10. That will return results similar to those shown in Figure 6-23.

```
#create a Threat Intelligence object:
ti = TILookup()

#use the IPAddress column values and check these against VirusTotal:
ti_warning = ti.lookup_iocs(signin_mfa_df, obs_col="IPAddress", providers=["VirusTotal"])

#filter out the column "Result" for a value of "True"
ti_warning[ti_warning["Result"]==True]
```

✓ 1 sec

	Ioc	IocType	Safeloc	QuerySubtype	Provider	Result	Severity	
0	105.161.22.56	ipv4	105.161.22.56	None	VirusTotal	True	information	{'verbose_msg': 'response_c
1	46.210.49.160	ipv4	46.210.49.160	None	VirusTotal	True	information	{'verbose_msg': 'response_c
2	88.3.168.167	ipv4	88.3.168.167	None	VirusTotal	True	information	{'verbose_msg': 'IP 'response_code': 1, 'positives':
3	130.44.170.146	ipv4	130.44.170.146	None	VirusTotal	True	information	{'verbose_msg': 'response_c

FIGURE 6-23 Results of a VirusTotal lookup based on a Result filter

11. You can easily use a visualization by taking the same data frame and plotting it, using something similar to this:

```
ti_warning["Result"].value_counts().plot(kind='pie')
```

See Figure 6-24.

```
ti_warning["Result"].value_counts().plot(kind='pie')
```

✓ <1 sec

```
<matplotlib.axes._subplots.AxesSubplot at 0x7f22f00c9220>
```

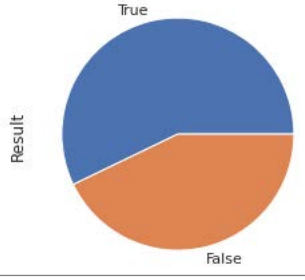


FIGURE 6-24 VirusTotal results visualized

Creating interactive cells

For triaging an incident, it is very common to use interactive cells that will prompt you for input. The example below is based on Maxmind's GeoLiteLookup for retrieving geo IP information. The installation and configuration are covered in the sample Notebook that you have been using so far.

1. Add a new cell in the Notebook that you have been working on.
2. Copy and paste the following into your new cell:

```
# Investigate an IP address, ask for user input:
IPaddress = input("Which IP address to investigate? ")
# use as sample: '90.156.201.97'

iplocation = GeoLiteLookup()
loc_result, ip_entity = iplocation.lookup_ip(ip_address=IPaddress)

print('Raw result')
display(loc_result)

print('IP Address Entity')
display(ip_entity[0])
```

3. Execute the cell. Your output should look like Figure 6-25.

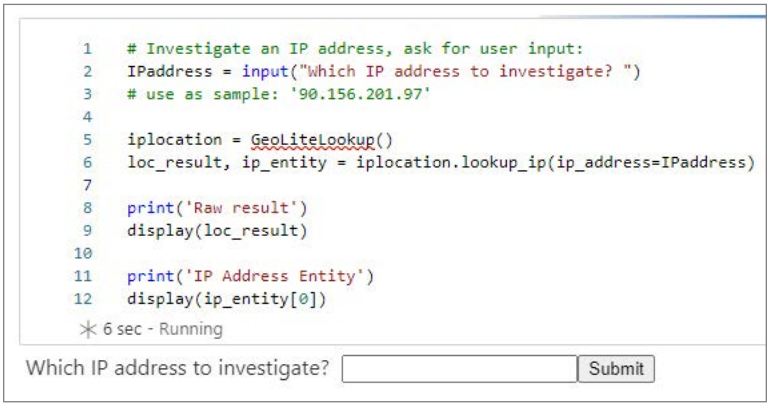


FIGURE 6-25 GeoIP look up

4. After pasting in an IP address and clicking **Submit**, you should see something similar to what's shown in Figure 6-26.

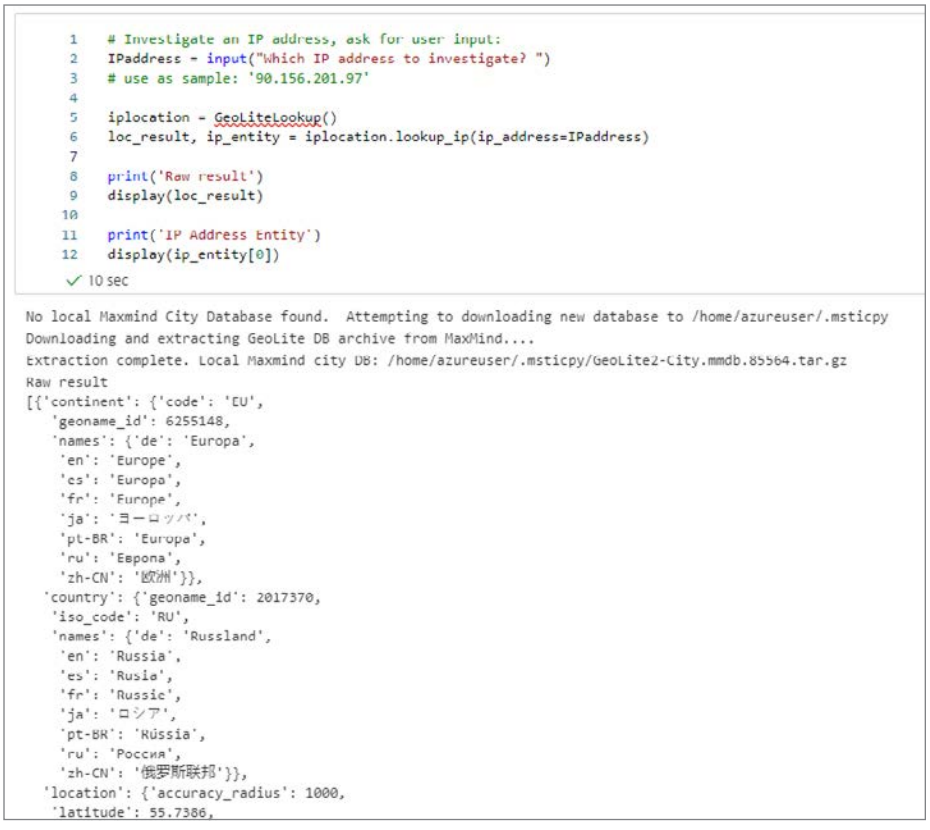


FIGURE 6-26 GeoIP look up results

The intention of this chapter was to provide you with an interactive introduction to Notebooks, with practical examples for you to try out. This chapter only scratched the tip of the iceberg. To continue your exploration and Notebook learning path, a copy of the next steps, as listed in the sample Notebook you have been using, is summarized below:

1. Run the Getting Started Notebook in Azure Sentinel. This will help you get your configuration set up.
2. Try the MSTICPy Lab at *https://aka.ms/msticpy-demo*.
3. Read the documentation at *https://msticpy.readthedocs.io/en/latest/GettingStarted.html*.
4. Learn more about Pandas at *https://pandas.pydata.org/docs/*.
5. Check out our other Notebooks for ideas! See *https://github.com/Azure/Azure-Sentinel-Notebooks*.

Chapter 7

Automating response

Security Orchestration, Automation and Response (SOAR) is defined as a solution stack of compatible software programs that allow an organization to collect data about security threats from multiple sources and respond to low-level security events without human assistance.

In Microsoft Sentinel, you can leverage automation rules and Playbooks—which is a direct integration with Logic Apps—to perform SOAR for Incidents that are created in your environment. Automation rules are a way to automate incident handling to perform simple tasks like assigning the incident to SOC personnel. Automation rules can also call Playbooks that provide the ability to build flows that can automate your investigations and respond to security alerts. Playbooks have hundreds of built-in connectors, making it easy to connect to systems, data, and apps and integrate and orchestrate for security response. If a connector doesn't exist, you can even create a custom connector.

TIP **This chapter will not concentrate on understanding Logic Apps. For more information on Logic Apps, see *https://aka.ms/ASB/LogicApps*.**

Microsoft Sentinel provides several ways to leverage Logic Apps. Real-time automation can be configured as part of the analytic to call a Playbook when the analytics are triggered. This will call the Playbook automatically when the incident is created. The second option allows you to call a Playbook from the incident on-demand. Lastly, you can use an automation rule to call the Playbook, which can apply across multiple incident types.

The importance of SOAR

In today's cyber landscape, the number of threats is increasing, which leads to an increasing number of alerts security teams need to respond to. SOAR can be used to enrich alerts with data from other sources, investigate entities for more context, orchestrate across the organization, and act on incidents. Using SOAR can reduce the time to resolution of security incidents and allow security teams to focus on the most important alerts. Security teams can automate the response actions for low-severity incidents, which can eliminate the security team's need to even be involved. They can enrich and investigate a medium-severity incident, again reducing the time needed to understand what happened and decide on a response

action. All this leaves more time to focus on high-severity incidents that have a greater impact on the organization.

Understanding automation rules

Automation rules allow the SOC to centrally manage the automations that occur for incident handling. Assigning Playbooks directly to Analytics Rules did not allow Playbooks to be easily configured for multiple rules. Automation rules can trigger on two options:

- **When Incident Is Created**—The automation rule runs when the incident is created from the analytic rule.
- **When Incident Is Updated**—The automation rule runs when changes are made to the incident.

Automation rules also provide conditions and simple actions that can be called to selectively apply the automation to incidents and handle some basic steps, such as assignment. Today, there are only five actions for an automation rule:

- **Change Status**—This can change the incident status to **New**, **Active**, or **Closed**.
- **Change Severity**—This can change the incident severity to **Informational**, **Low**, **Medium**, or **High**.
- **Assign Owner**—This can set the owner of the incident to an Azure Active Directory user or group.
- **Add Tags**—This can add an additional tag(s) to the incident.
- **Run Playbook**—This can call a Playbook to trigger the incident.

TIP **Only Playbooks with the incident trigger can be selected for automation rules, and Microsoft Sentinel must have explicit permissions to the Playbook.**

Automation rules can have multiple actions assigned in an "and then" fashion. The condition of the automation rule supports filtering to the Analytic Rule name, which can match either `contains` or `does not contain`. They can also be filtered by incident properties, such as the title, and entity properties, such as the account name. These conditions make the selective application of automation rules very powerful for SOAR.

Creating an automation rule

Now that we understand the purpose and capabilities of automation rules, let's create one. In this scenario, we will create an automation rule to raise the severity if the account is a high-value user.

1. Open the Azure portal and sign in as a user who has Microsoft Sentinel Contributor privileges.
2. In the search pane, type **Microsoft Sentinel** and click the Sentinel icon when it appears.
3. Select the workspace on which Microsoft Sentinel has been enabled.

4. In the left navigation pane, click **Automation**.
5. Click the **Create** button and select **Automation Rule**, as shown in Figure 7-1.

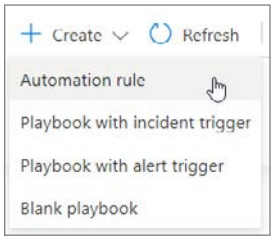


FIGURE 7-1 The Create Automation Rule button

6. Enter a name for the automation rule.
7. Click **+ Add Condition**.
8. From the dropdown menu that appears, select **Account Name**. Select **Equals** from the middle dropdown menu and enter a user name in the field box.
9. Click the **Actions** dropdown and select **Change Severity**. Select **High** from the dropdown menu. Figure 7-2 shows what the creation blade should look like.

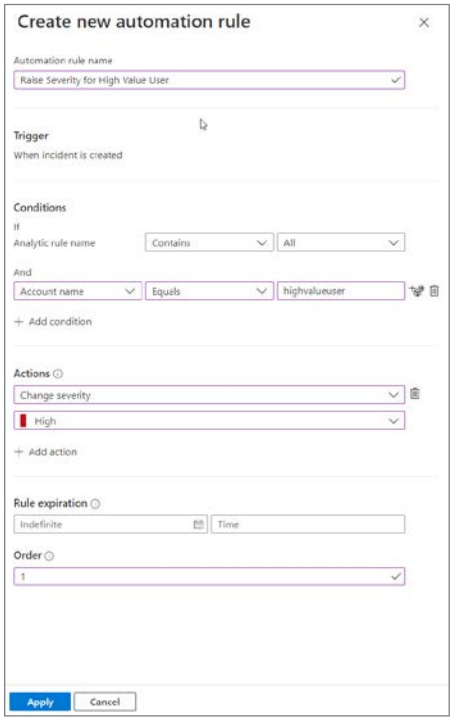


FIGURE 7-2 The Create New Automation Rule blade

10. Click **Apply**.

As you can see, creating automation rules is very easy. You can have many automation rules in the workspace to control and handle many response scenarios. Also, you can see incidents that have been modified by automation rules by searching the `SecurityIncident` table using the following query:

```
SecurityIncident
| where ModifiedBy contains "Automation"
```

Advanced automation with Playbooks

When you create an analytic or automation rule, you can define a Playbook to trigger. The Playbook is automatically run when the analytic or the automation rule is triggered, and it follows the Playbooks' steps as you have configured them. Use the following steps to create and configure a Playbook automation.

1. Open the Azure portal and sign in as a user who has Microsoft Sentinel Contributor privileges.
2. In the search pane, type **Microsoft Sentinel** and click the Sentinel icon when it appears.
3. Select the workspace on which Microsoft Sentinel has been enabled.
4. In the left navigation pane, click **Automation**.
5. Click the **+Create** dropdown and select **Playbook With Incident Trigger**, as shown in Figure 7-3.

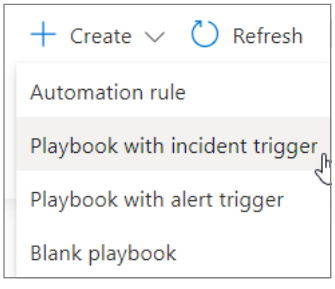


FIGURE 7-3 Create Playbook with the incident trigger

6. The **Logic App Create Playbook** blade appears, as shown in Figure 7-4. Enter **Prompt-User** in the Name field. Ensure that you have selected the correct subscription from the **Subscription** dropdown menu. Select **Use Existing** from the **Resource Group** menu. The Log Analytics option has two choices: On or Off. This option chooses whether to save diagnostic logs for Logic App in Log Analytics. Enabling this feature can help by providing richer debugging details.
7. Click the **Next: Connections** button.

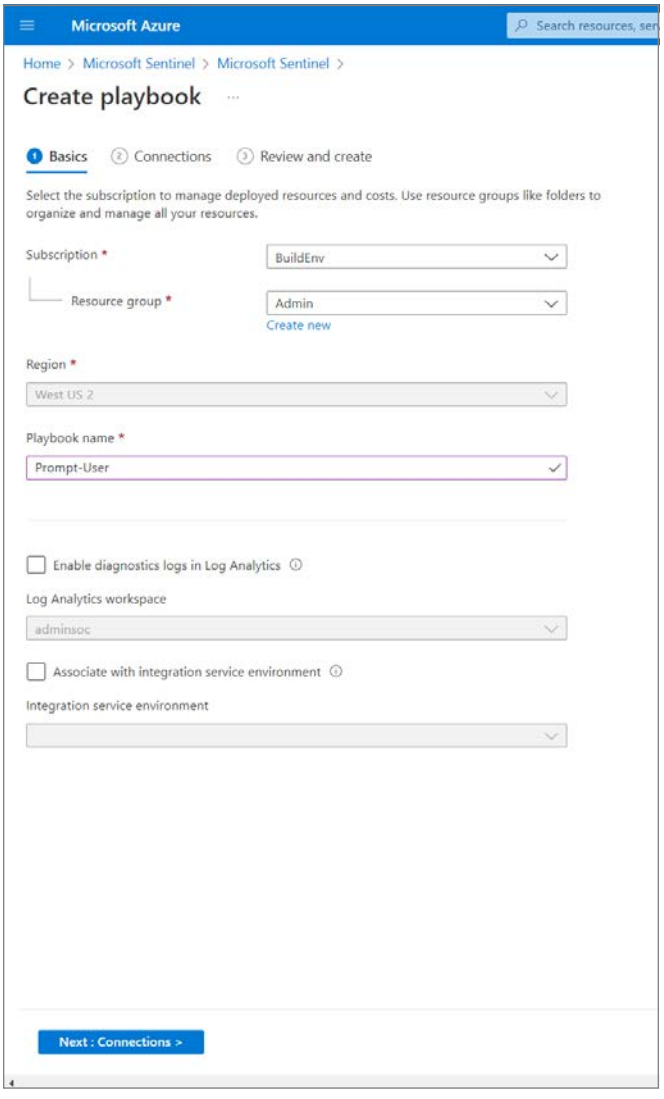


FIGURE 7-4 Logic App Create Playbook Blade

8. Next, the Connections are shown for the Playbook, as shown in Figure 7-5. Because the creation wizard was called from Microsoft Sentinel, it already has the Microsoft Sentinel connection selected and configured with the **Connect With Managed Identity** option.
9. Click **Next: Review And Create**.

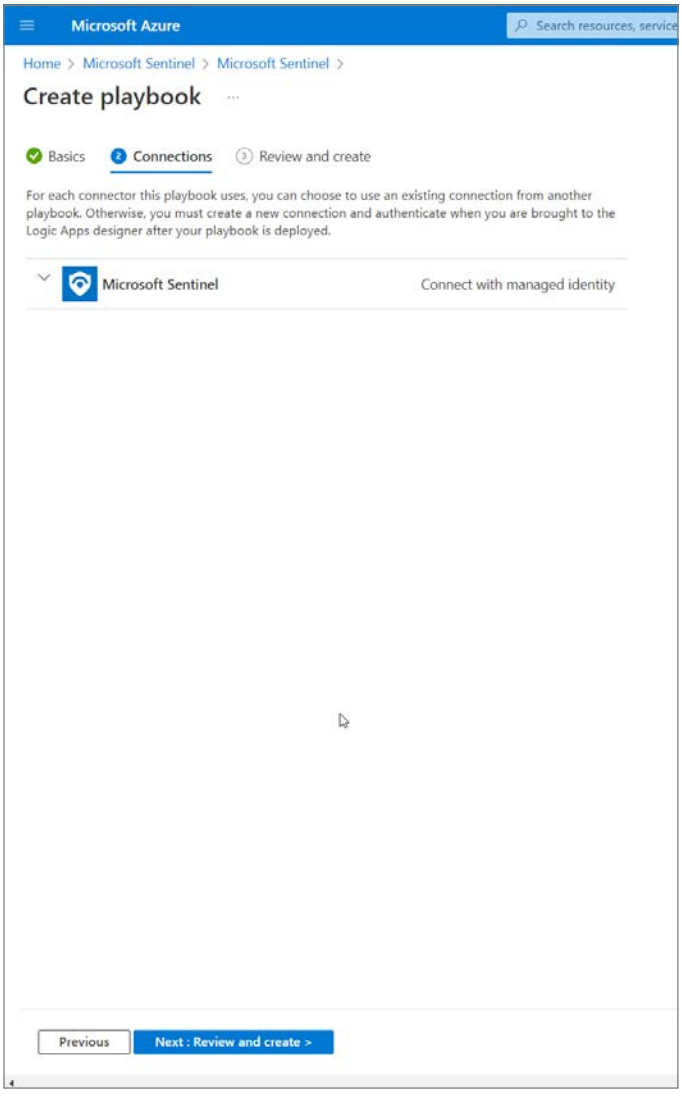


FIGURE 7-5 Logic App Create Playbook / Connections Options blade

10. Review the Playbook creation options and click **Create And Continue To Designer**.

11. Once created, the Logic Apps Designer blade appears, as shown in Figure 7-6. The basic structure has already been selected, which starts with the Microsoft Sentinel Incident trigger.

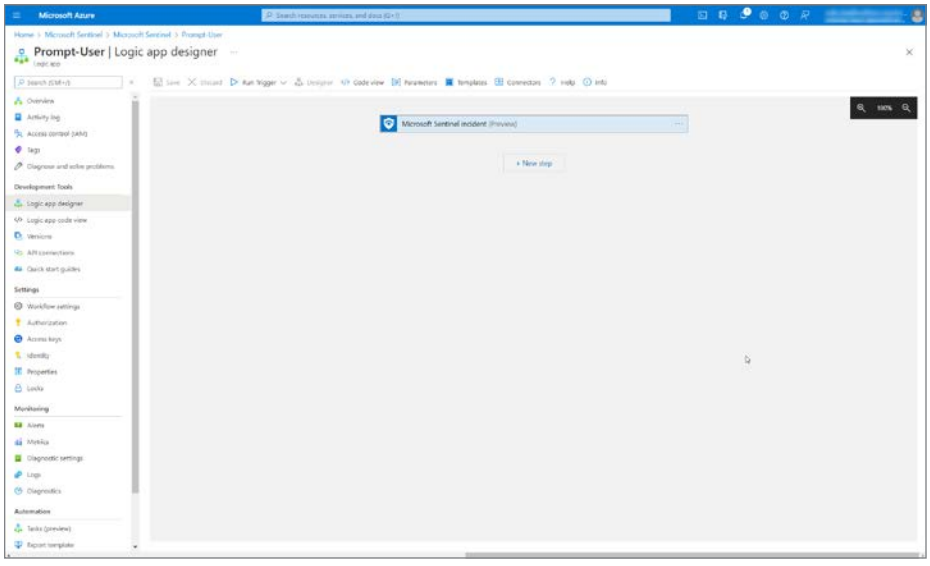


FIGURE 7-6 Logic App Designer blade

Each Logic App connector requires an application program interface (API) connection resource. These API connections store the variables and tokens needed to access the API for the connection, such as Office or Azure. Logic Apps make it easy by allowing you to sign in as you add new connectors and creating the API connection resource for you. The trigger already has an API connection created and configured from the creation wizard.

Each Playbook must start with a trigger. This is the action that starts the Playbook run. You can start adding actions after the trigger. Click the **New Step** button to add a step, as shown in Figure 7-7.

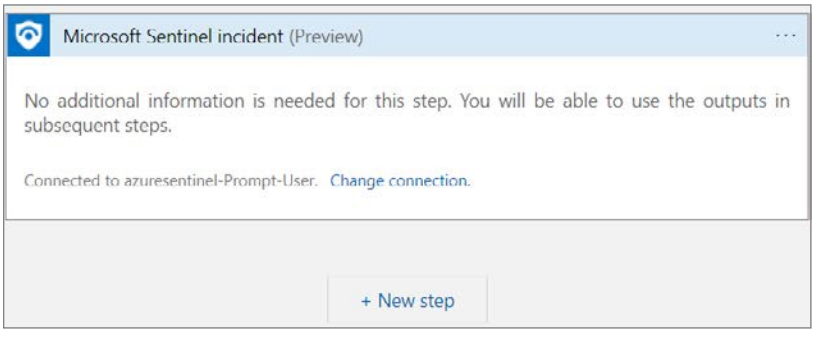


FIGURE 7-7 New Step button in Logic App Designer

In this Playbook, you are going to prompt the user to see if they had indeed taken the action that was part of the incident.

1. The first thing you need to do is get the user entity from the property of the incident. Search for Microsoft Sentinel and select **Entities–Get Accounts**, as shown in Figure 7-8.

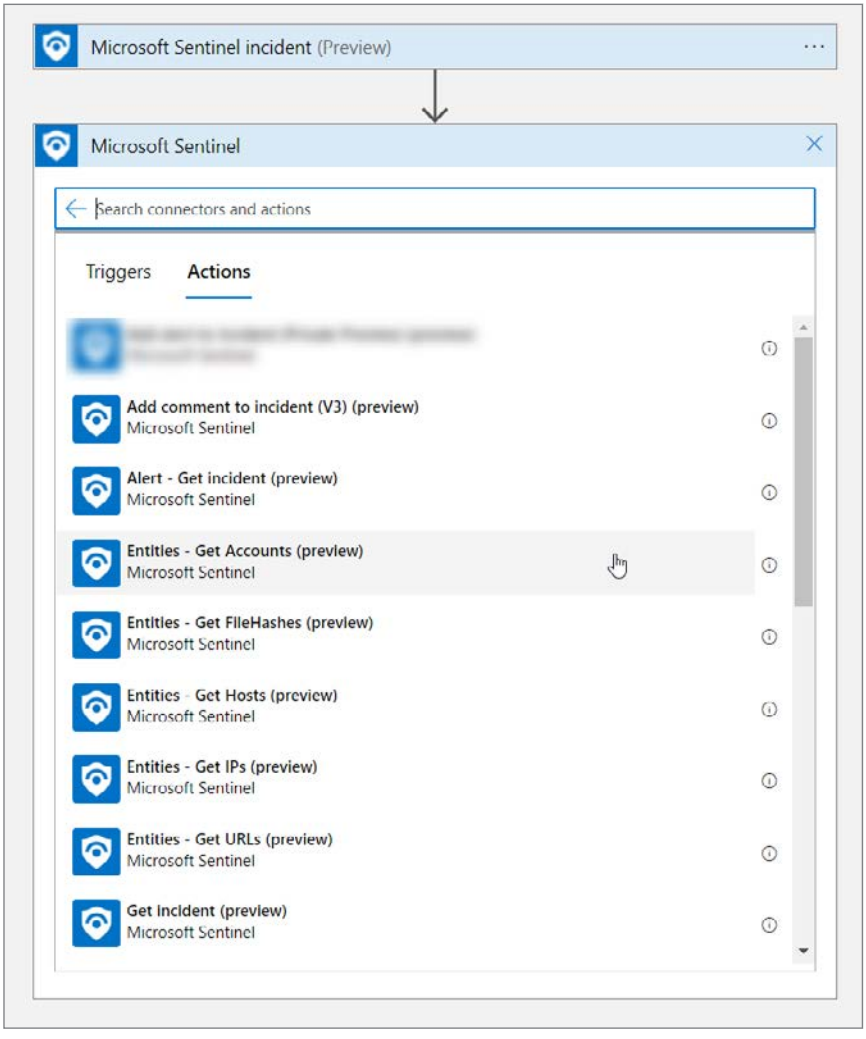


FIGURE 7-8 Adding an action to the Playbook. (Some information is intentionally blurred for security reasons.)

2. For the **Entities–Get Accounts** action, you are required to provide the list of entities from the Microsoft Sentinel incident or alert. The great thing about Logic Apps is that each step has inputs and outputs. Those outputs become Dynamic Properties that can be used in later steps. The trigger named **Microsoft Sentinel Incident** provides dynamic properties like Incident ARM ID**, Entities, Incident Title**, and the like.
3. Click the **Entities List** field, and the **Dynamic Content** flyout menu will appear. Select **Entities** from the **Dynamic Content** list, as shown in Figure 7-9.

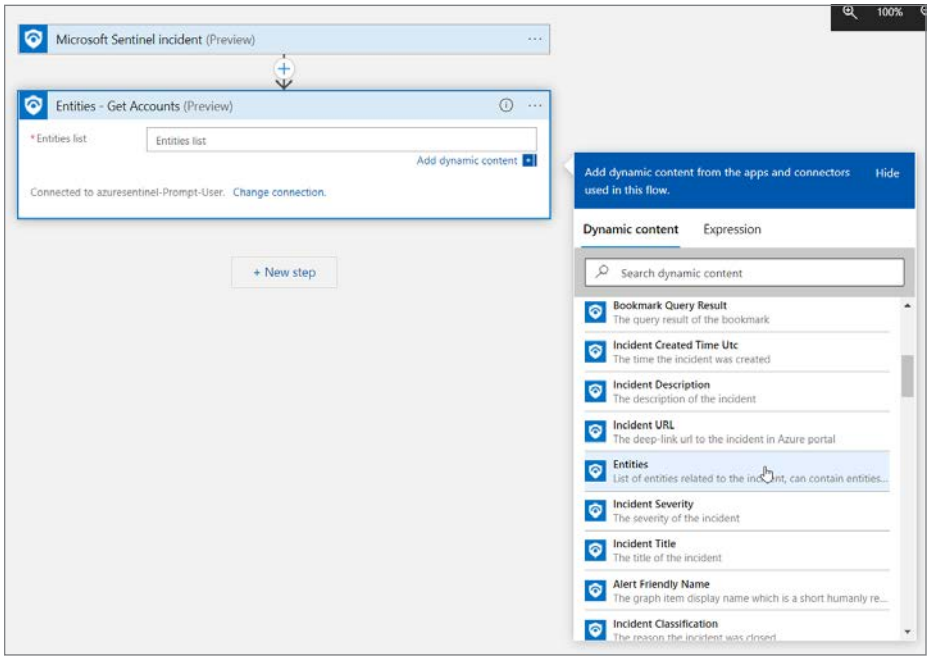


FIGURE 7-9 Adding Dynamic Content to the action

4. Click **New Step** and type **Azure AD**. Click **Azure AD** and select **Get User,** as shown in Figure 7-10. Click **Sign In** and provide credentials to create the Azure AD API connection.

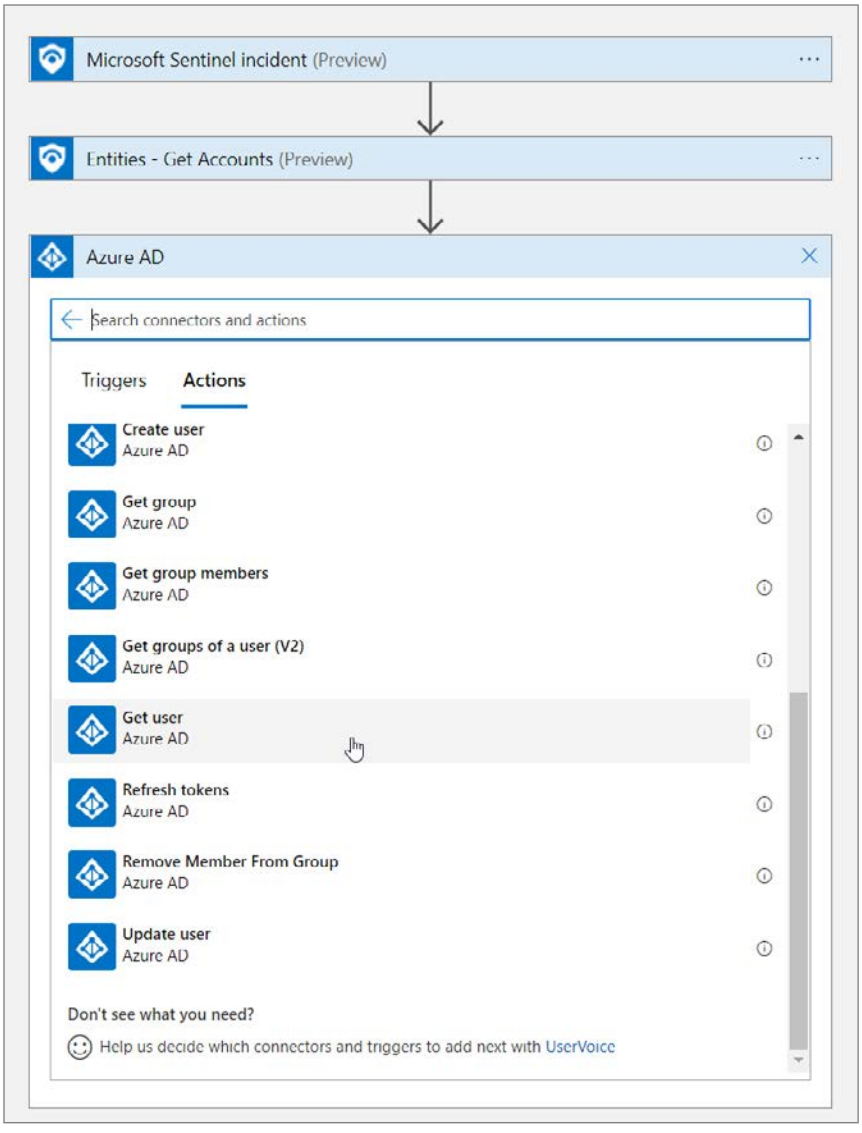


FIGURE 7-10 Adding another action to get the AAD user in Logic Apps

5. Click the **User ID Or Principal Name** text box. Select **Accounts AAD User ID** from the **Dynamic Content** flyout menu, as shown in Figure 7-11. Notice that once you click the **Accounts AAD User ID**, Logic Apps adds a **For Each** loop action. This is because the Accounts returned from the **Entities–Get Accounts** action is an array and could contain multiple accounts.

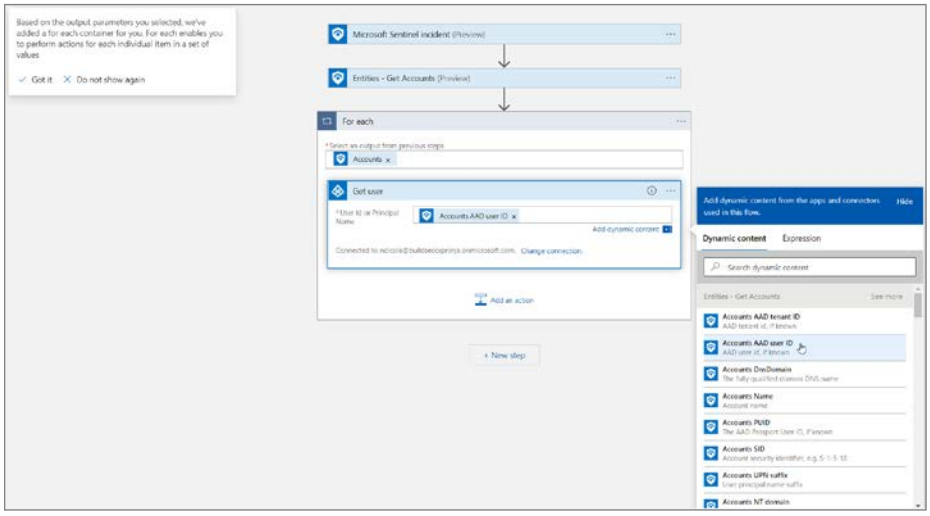


FIGURE 7-11 Adding the dynamic property to the AAD action

6. In the **For each** dialog, click **Add An Action** and type **Office 365**. Select **Office 365 Outlook,** scroll down, and select **Send Approval Email,** as shown in Figure 7-12.

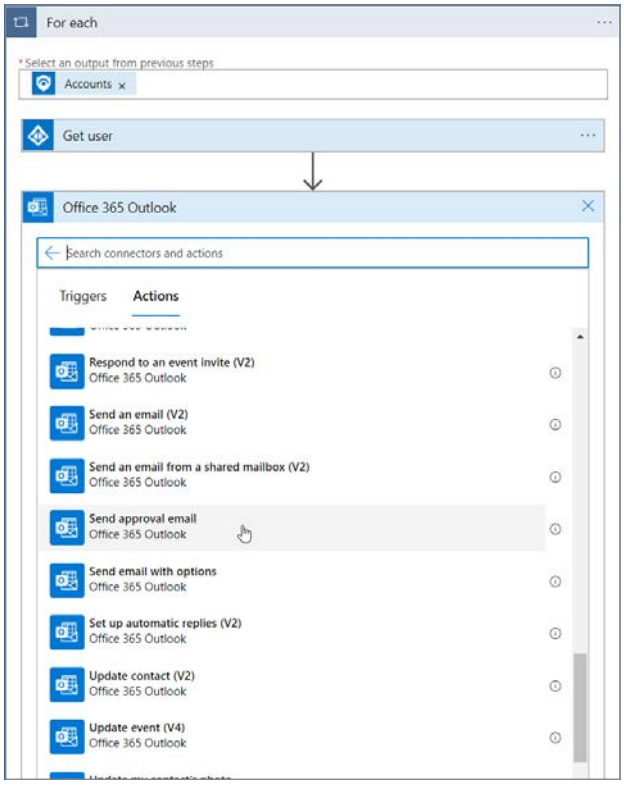


Figure 7-12 Adding an Office 365 action to send an email

TIP You can click the Information buttons next to actions and triggers to see what they do.

7. Click **Sign In**. Enter credentials to create the Office 365 Outlook API connection.
8. Click the **To** box. Select **Mail** from the **Get User** step in the **Dynamic Content** flyout menu.
9. Change the **Subject** text box to **Security Alert** and add the **Incident Title**.
10. In the **User Options** box, change the text to something like **This was me** or **This was not me**.
11. Change the **Importance** to **High**.
12. Click **Add New Parameter Option** and select **Body**. Click outside the dropdown menu to make it disappear.
13. For the **Body**, enter **New alert from Microsoft Sentinel. Please respond ASAP**.
14. In the **Dynamic Content** menu, under **Microsoft Sentinel Incident**, choose **Severity**.
15. In the **Dynamic Content** menu, under **Microsoft Sentinel Incident**, choose **Incident Description**.
16. The text is using text plus dynamic content from the previous actions. Figure 7-13 shows an example.

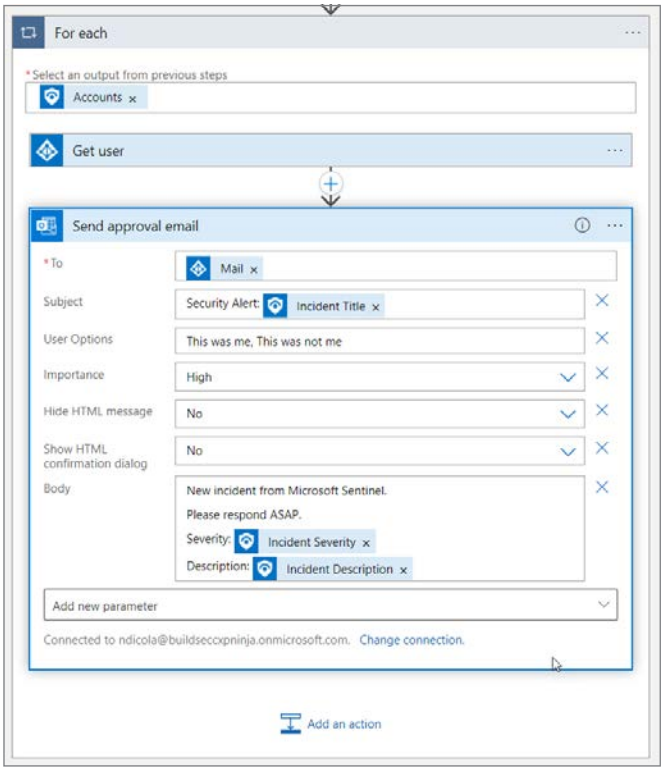


FIGURE 7-13 Setting the properties of the Send Approval email action

17. Click **Add An Action,** type **Control**, and Select **Control**.

18. Select **Condition**. Condition is an `If` operator, so we can use this to determine the action to take based on the response. In the **Condition** menu, select **Selected Option**. For the value, enter something like **This was me.** See Figure 7-14 for an example.

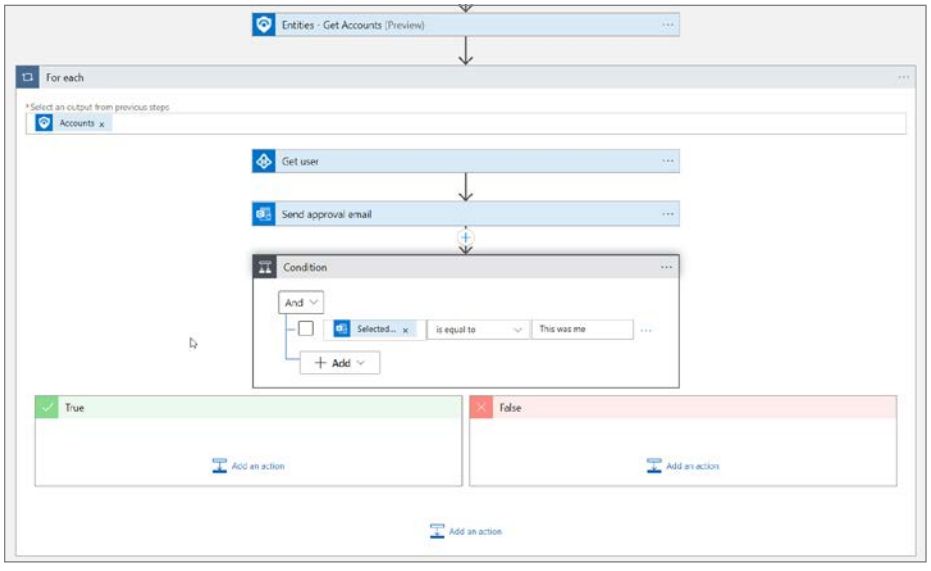


FIGURE 7-14 Adding a condition to evaluate

19. In the **True** condition box, click **Add An Action** and click **Microsoft Sentinel**. Then select **Update Incident**. Enter the **Incident ARM ID** using the dynamic properties.

20. Select **Closed** in the **Status** field and select **BenignPositive–SuspiciousButExpected** in the **Classification Reason** field.

21. Type **User confirms it was them** in the **Close Reason** text box. See Figure 7-15 for the action.

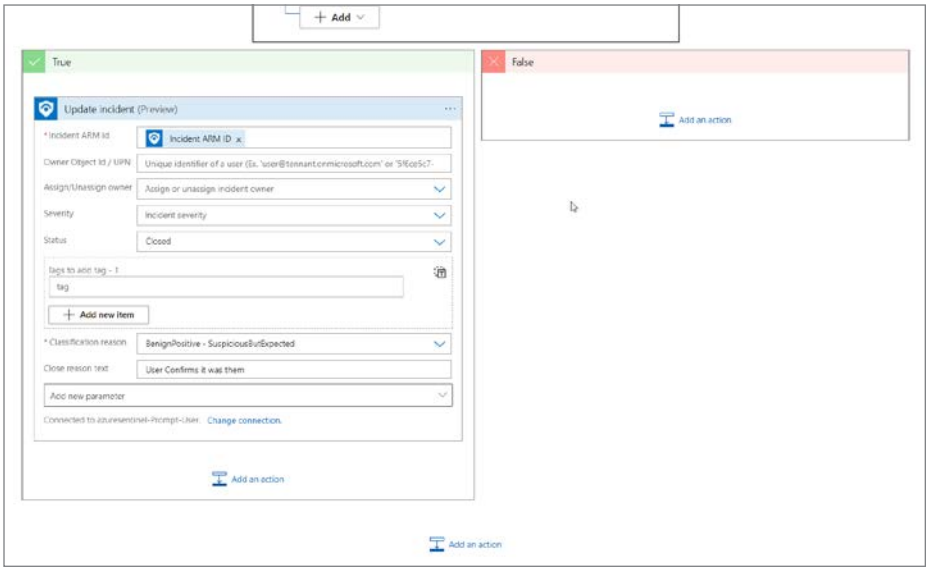


FIGURE 7-15 Adding steps in the If true action area

We have now configured the true side of the condition. We are checking with the user to see if they performed the action from the Microsoft Sentinel incident and if so, we are closing the incident as resolved. This means the security analyst does not need to go investigate the incident further.

1. In the **False** condition box, click **Add An Action** and click **Microsoft Sentinel**. Then select **Add Comment To Incident**. Enter the **Incident ARM ID** using the dynamic content from the trigger. Type **User confirms they did not complete the action. Further investigation is needed** in the **Incident Comment** box.
2. In the **If False** condition, click **Add An Action** and type **Microsoft Teams**. Click **Microsoft Teams** and select **Post A Message In A Chat Or Channel**. Click **Sign In** and use the pop-up menu to sign in.
3. Select **User** in the **Post As** field.
4. Select **Channel** in the **Post In** field.
5. Select your **Team** from the dropdown menu.
6. Select your **Channel** from the dropdown menu.
7. In the **Message** body, enter **New alert from Microsoft Sentinel. Please investigate ASAP**.

8. In the **Dynamic Content** menu, under **Microsoft Sentinel Incident**, choose **Incident Severity**.
9. In the **Dynamic Content** menu, under **Microsoft Sentinel Incident**, choose **Incident Description**.
10. In the **Dynamic Content** menu, under **Microsoft Sentinel Incident**, choose **Incident URL**.
11. Click **Add Parameter**, select **Subject**, and click outside the dropdown menu to make it disappear.
12. In the **Subject** box, enter **Security Alert:** and select **Incident Title** from the dynamic content. See Figure 7-16 for the completed action.

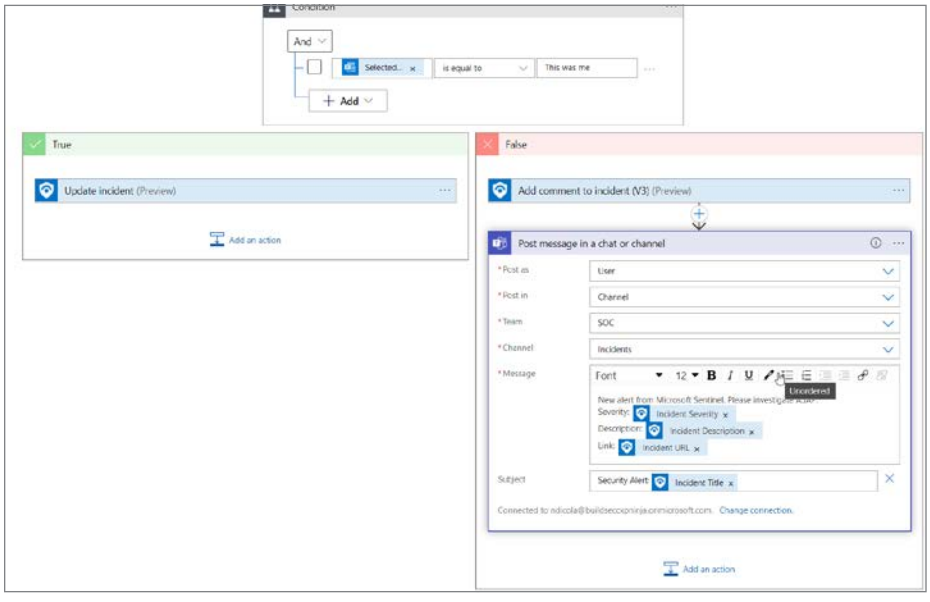


FIGURE 7-16 Adding a Microsoft Teams action to post a message

13. Click the **Save** button for the Logic App, as shown in Figure 7-17. Figure 7-18 shows the completed Playbook.

Figure 7-17 The Save button for Azure Logic Apps

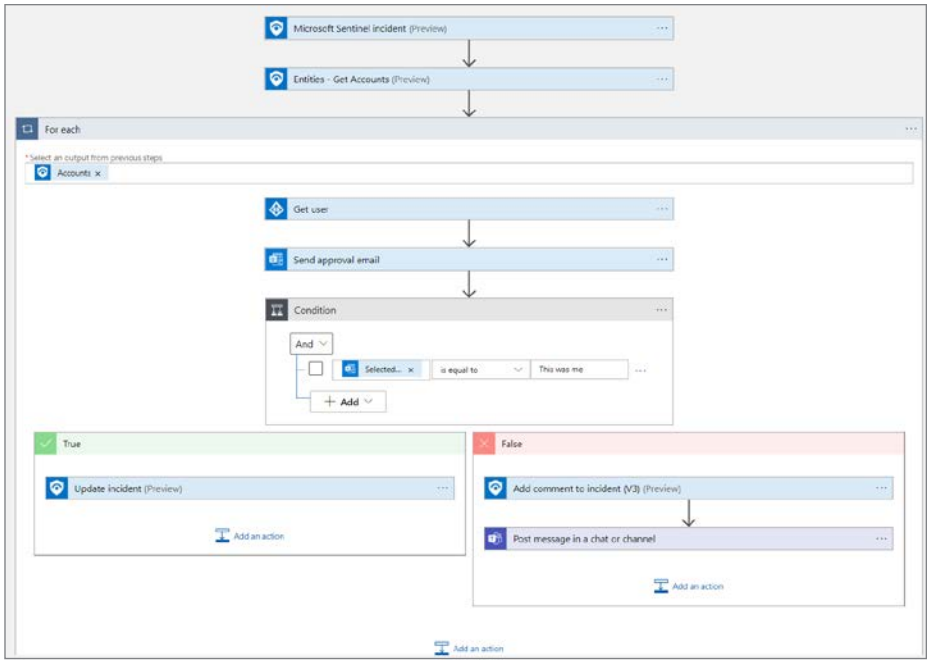


FIGURE 7-18 The completed Playbook

14. In the Azure navigation bar, click **Microsoft Sentinel** to go back to the Automation blade.
15. Click **+ Create** and select **Automation Rule**.
16. Enter **Prompt User** for the Automation Rule Name. Click the dropdown for the **If Analytic Rule Name** condition. Type **VM** and select the **Azure VM Deletion** rule created in Chapter 2.
17. Click **Manage Playbook Permissions**. Select the Resource Group where the Playbook was created. This will grant Microsoft Sentinel permissions to run the Playbook automatically.
18. Click the **Actions** dropdown and choose Run Playbook. In the Playbook selection dropdown, type **Prompt-User** and select the **Prompt-User Playbook**. Figure 7-19 shows the properties of the **Create New Automation Rule** blade.

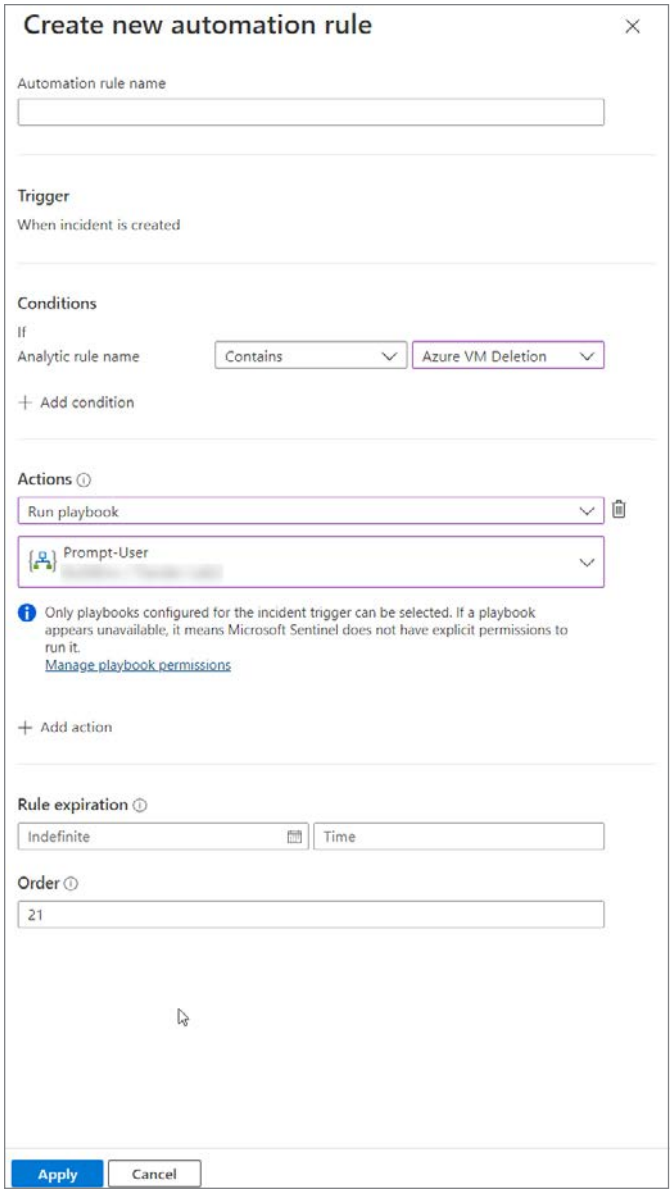


FIGURE 7-19 The completed automation rule

19. Click Apply.

Perform the following tasks to create a new VM to test the automation rule:

1. Create a new Virtual Machine with the following specifications:

- **Operating system:** Windows Server 2016.
- **Resource group:** Use the same resource group that you created for the workspace earlier in this chapter.

2. Once the VM is created, go to the resource group. Select the virtual machine and click **Delete**. It will take a few minutes for the activity logs to populate and for the analytic to trigger. Figure 7-20 shows the deletion logs in Azure Activity.

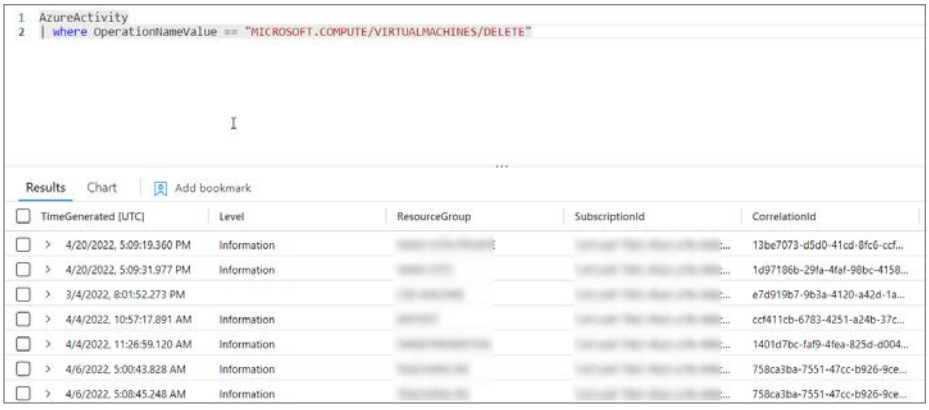


FIGURE 7-20 View of the Azure Activity Logs for Delete Virtual Machine

3. Now that the incident has been created, we can see in Figure 7-21 that the Playbook has run and is waiting for the user input.

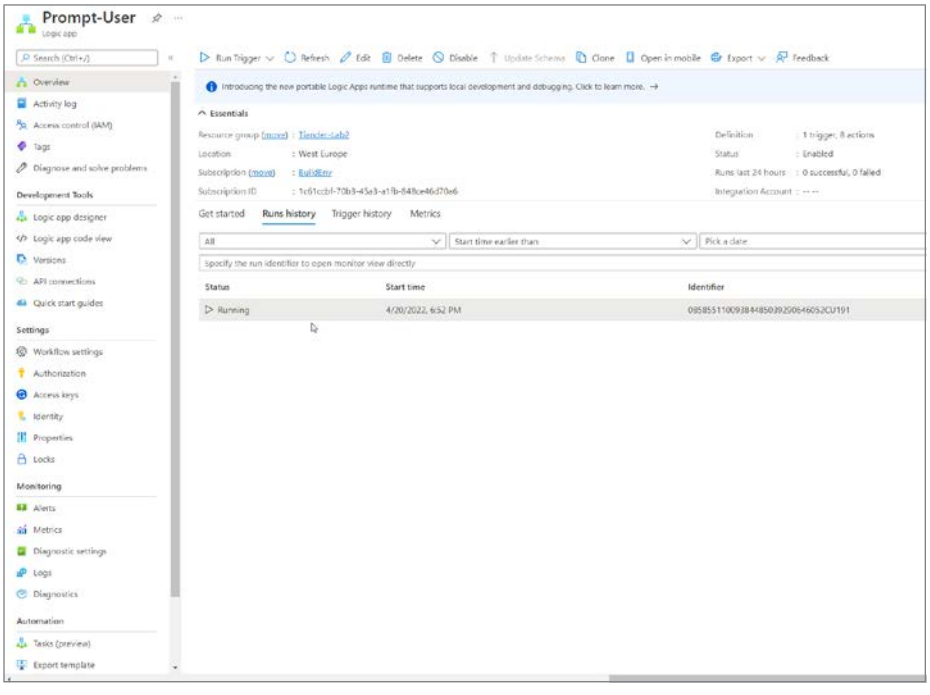


FIGURE 7-21 Logic App Playbook blade showing the run history

4. If we look in the user mailbox, we can see the email from our Playbook, as shown in Figure 7-22. Click **This Was Not Me**.

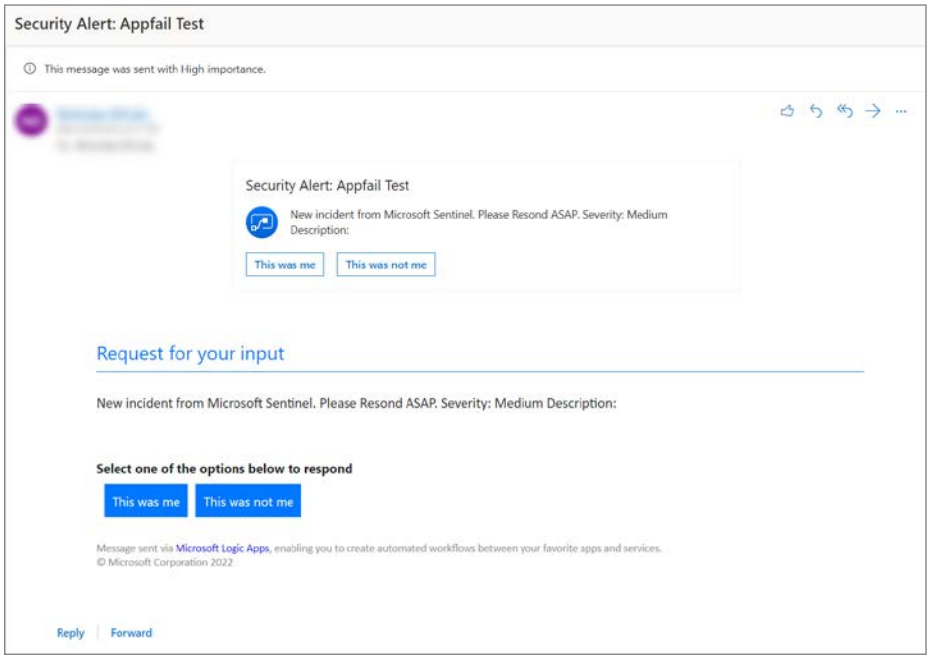


FIGURE 7-22 The Send Approval Email

5. Go to **Microsoft Sentinel**, and in the **Azure portal**, click **Incidents** and select the incident. After a few moments, you should see the comment that was added (see Figure 7-23).

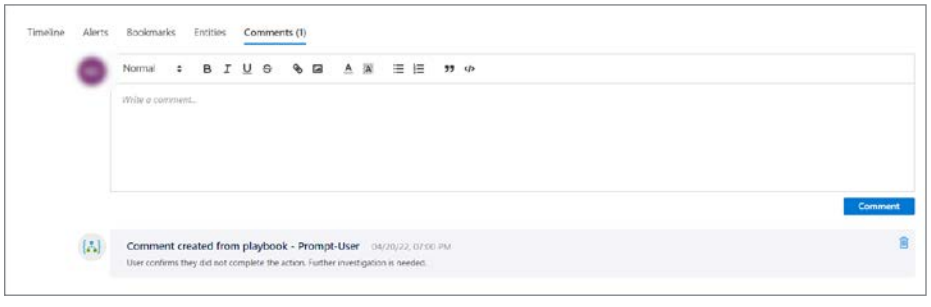


FIGURE 7-23 The Incident with automated comment

6. Figure 7-24 shows the Microsoft Teams message that was posted to the SOC Channel.

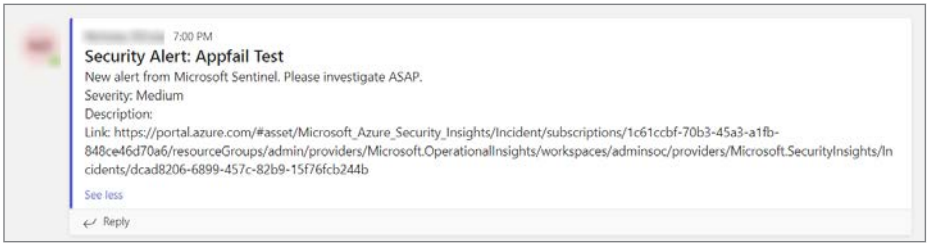


FIGURE 7-24 The message posted to Microsoft Teams

As you can see in this section, you can create some simple but powerful Playbooks to help reduce the work on security analysts so they can focus on creating new detections, improving existing detections, and investigating higher-severity alerts.

Post-incident automation

Not every incident can be automatically remediated using real-time automation as part of the analytic. This is because your SOC processes might not have a process defined, or the incident needs more investigation before executing the Playbook. In this section, we will cover the capability to run Playbooks on demand from the incident details.

You might want to use this to trigger steps as part of the investigation, like isolating a VM in the cloud. Or, you could use it to conduct some remediation action once you have completed your investigation to clean up the incident.

1. Open the **Azure portal** and sign in as a user who has Microsoft Sentinel Contributor privileges.
2. In the search pane, type **Microsoft Sentinel** and click the Sentinel icon when it appears.
3. Select the workspace on which **Microsoft Sentinel** has been enabled.
4. In the left navigation pane, click **Automation**.
5. Click the **Playbook Templates** tab. Microsoft Sentinel has a Playbook gallery, where you can deploy templated Playbooks for use in the environment.

6. Type **Reset-AAD** in the **Search By Name** box. Select the **Reset-AADUserPassword Playbook** with the Microsoft Sentinel Incident trigger. Click **Create Playbook**, as shown in Figure 7-25.

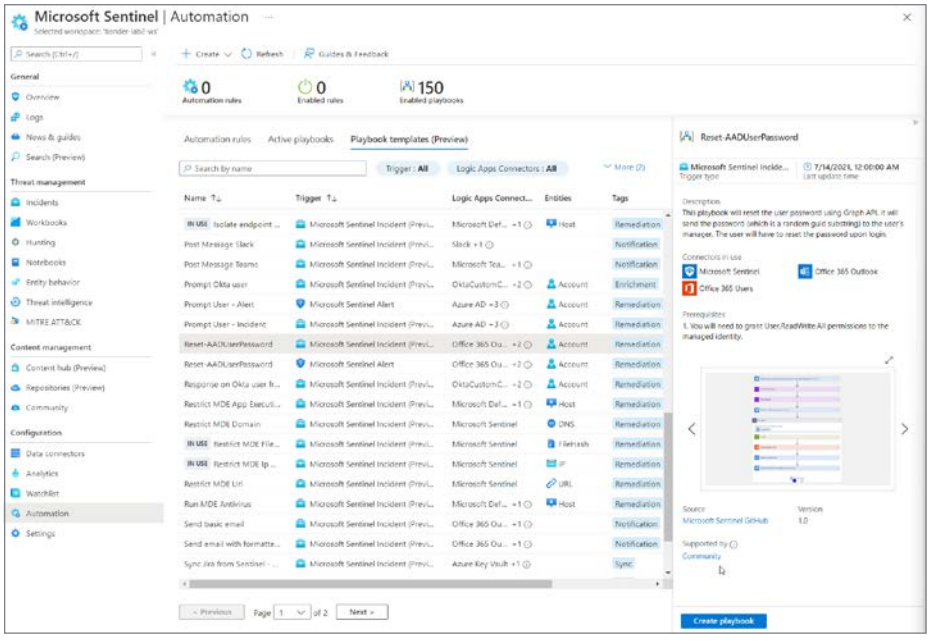


FIGURE 7-25 The Playbook gallery

7. Select a resource group to deploy to. Click **Next: Parameters**.
8. Enter a username@domain to use for the API connections. Click **Next: Connections**.
9. Review the connections and click **Next: Review And Create**.
10. Click **Create And Continue To Designer**.

11. When the Designer opens, as shown in Figure 7-26, you can see some of the connections are not authenticated.

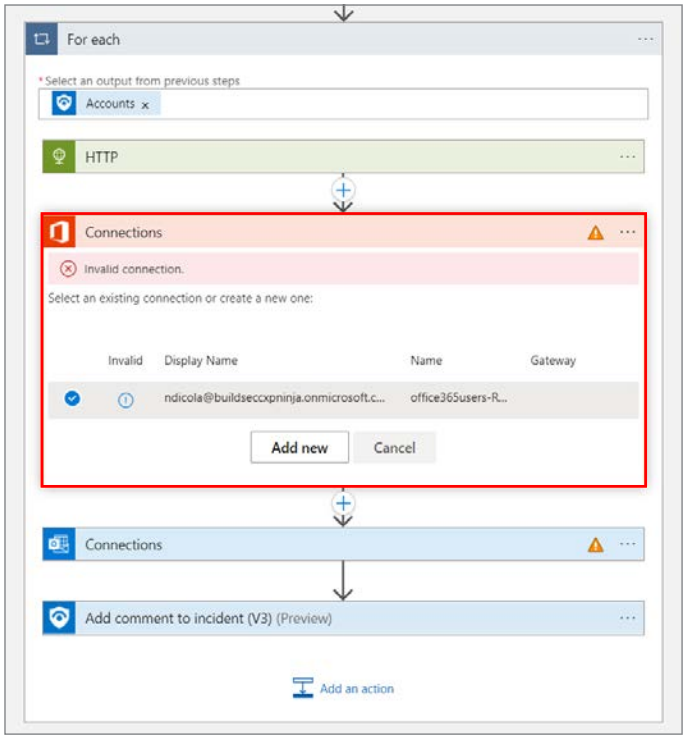


FIGURE 7-26 Logic App Designer

12. Click **API Connections** in the left menu. Select the Office 365 connection.
13. Click **Edit API Connection** in the left menu. Click **Authorize**, as shown in Figure 7-27.

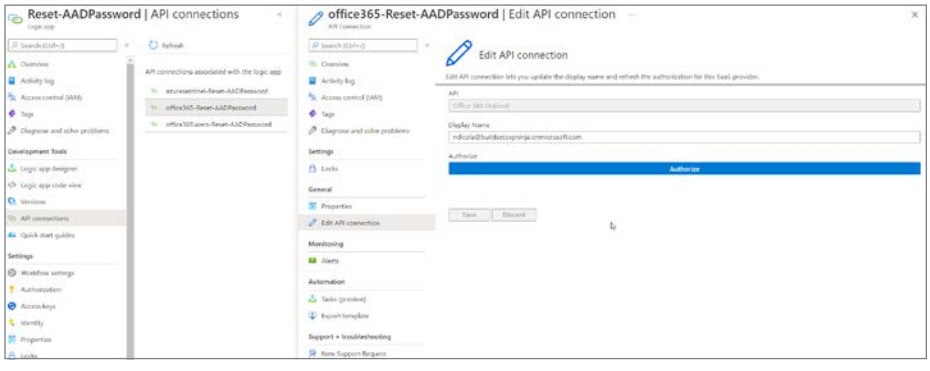


FIGURE 7-27 The Edit API Connection blade

14. A pop-up window will appear. Sign in to Azure. After the authentication is successful, click **Save**.

15. Repeat steps 12-14 for the Office 365 Users' connection.
16. In the search pane, type **Microsoft Sentinel** and click the Microsoft Sentinel item when it appears.
17. Select the workspace on which **Microsoft Sentinel** is enabled.
18. Click **Incidents**.
19. Click the **VM Deletion** incident we have been using.
20. Click **Actions**. Figure 7-28 shows the analyst could manually trigger the Playbook after investigation from here.

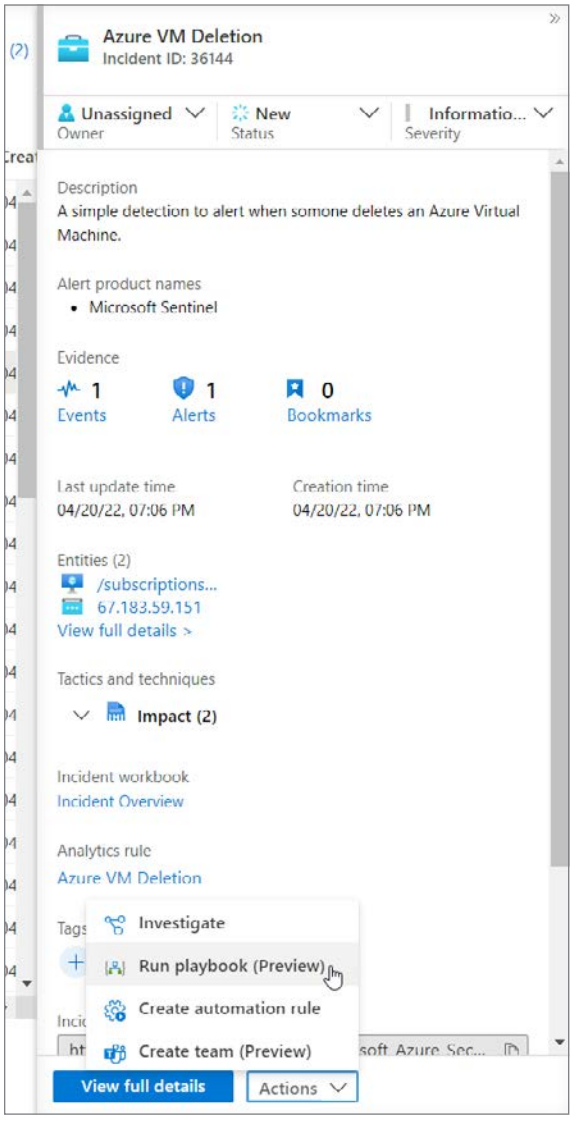


FIGURE 7-28 The Actions menu of an Incident

In this chapter, you learned how to create an automation rule and a Playbook from scratch, how to use automation rules for real-time response, and how to run Playbooks on-demand as part of your investigation. You can use these techniques to automate response actions and help speed up the investigation and triaging of incidents in Microsoft Sentinel.

Chapter 8

Data visualization

A great way to make sense of large volumes of data is to create graphic visualizations that make it easier for users or consumers of the data to understand what the data is telling them. Graphics can make spotting trends easier to identify, help clarify relationships between data elements, and improve the decision-making cycle.

Some of the most common data visualizations include time-series analysis (line charts), ranking (bar charts), ratio analysis (pie charts), frequency distribution, geospatial (maps), correlation (scatterplots), and cluster analysis. In this chapter, you will learn more about Microsoft Sentinel Workbooks, how to leverage the built-in Workbooks, and how to create your own Workbook.

Microsoft Sentinel Workbooks

Sentinel Workbooks provide interactive reports that can be used to visualize your security and compliance data. Workbooks combine text, queries, and parameters to make it easy for developers to create mature visualizations, and they provide advanced filtering, drill-down capabilities, advanced dashboard navigations, and more.

Also, Workbooks allow users of the dashboards to edit and customize the visualizations to meet their needs using simple dropdown menus. While you can create your own Workbook, it is important to review the library of templates to see if what you need is already there. To view the available Workbook templates, follow the steps below:

1. Open the **Azure portal** and sign in as a user who has either contributor or reader permissions on the resource group to which the Microsoft Sentinel workspace belongs.
2. In the search pane, type **Sentinel** and click the Microsoft Sentinel icon when it appears.
3. Select the workspace on which **Microsoft Sentinel** has been enabled.

4. In the left navigation pane, click **Workbooks**. By default, the **Templates** tab will appear, as shown in Figure 8-1.

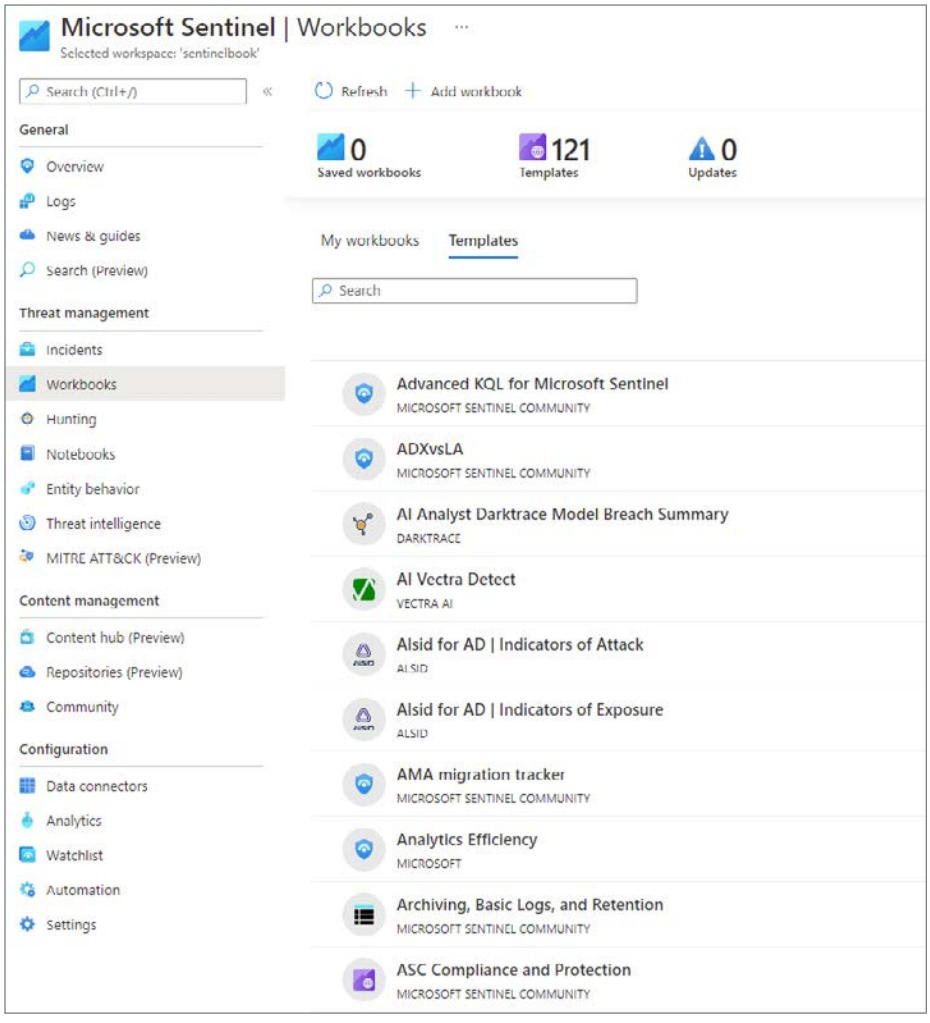


FIGURE 8-1 Sentinel Workbooks Templates tab

5. The **Templates** tab has a collection of Workbooks that were created based on customers' demands, and they were tailored for specific scenarios. It is very important to emphasize that if there is no data ingestion to feed the Workbook, there will be no data to show. For example, if you don't have the Office 365 Connector already working, nothing will appear when you load the Office 365 Workbook.

6. For this example, open the Azure Activity Log Workbook by typing **Activity** in the **Search** box and clicking the **Azure Activity** option. The **Azure Activity** blade appears, as shown in Figure 8-2.

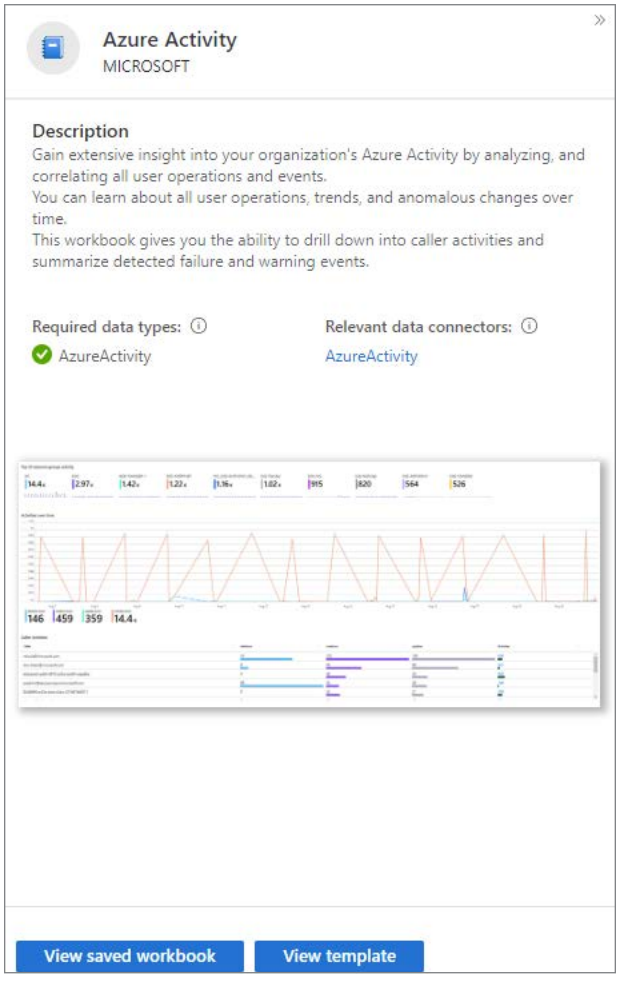


FIGURE 8-2 The Azure Activity blade

7. Click the **View Template** button to see what the Workbook looks like without the data. The structure of the Workbook appears as shown in Figure 8-3.

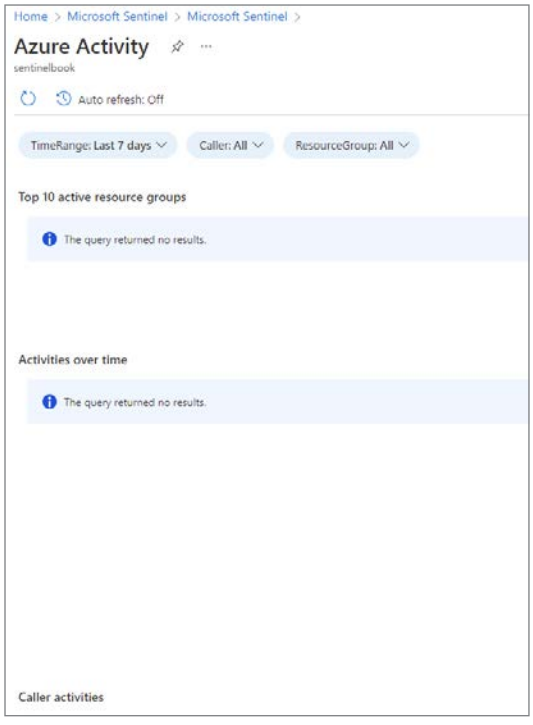


FIGURE 8-3 Workbook template without data

8. If your Workbook appears fully empty, as shown in Figure 8-3, it is because you don't have a data connector that is ingesting the data needed to feed this Workbook. When the connector is working properly, the Workbook template will look like the one shown in Figure 8-4.

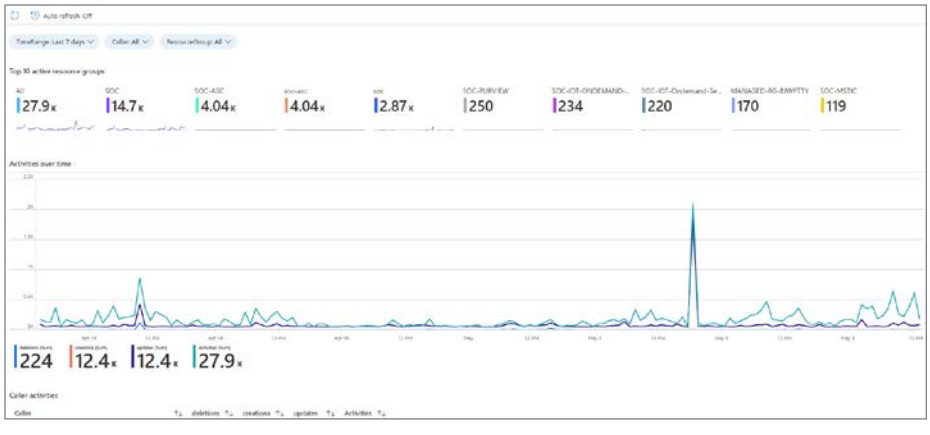


FIGURE 8-4 Workbook template with data

9. In the example shown in Figure 8-4, you have the Workbook fully populated with the data, which may not look exactly like the one you built since the data is different. After reviewing, you can click on the close icon (**X**) in the top-right corner of the Workbook.

To leverage a specific Workbook template, you must have at least Workbook reader or Workbook contributor permissions on the resource group of the Microsoft Sentinel workspace. The Workbooks that you can see in Microsoft Sentinel are saved within the Sentinel's workspace resource group and are tagged by the workspace in which they were created. Follow the steps below to view and make changes to the **Data Collection Health Monitoring** Workbook:

1. Open the **Azure portal** and sign in as a user who has either contributor or reader permissions on the resource group to which the Microsoft Sentinel workspace belongs.
2. In the search pane, type **Sentinel** and click the Microsoft Sentinel icon when it appears.
3. Select the workspace on which **Microsoft Sentinel** has been enabled.
4. In the left navigation pane, click **Workbooks**.
5. In the **Search** box, type **Data collection** and click the **Data Collection Health Monitoring** Workbook.
6. The **Data Collection Health Monitoring** blade appears on the right side. Click the **View Template** button, and the **Data Collection Health Monitoring** page appears, as shown in Figure 8-5.

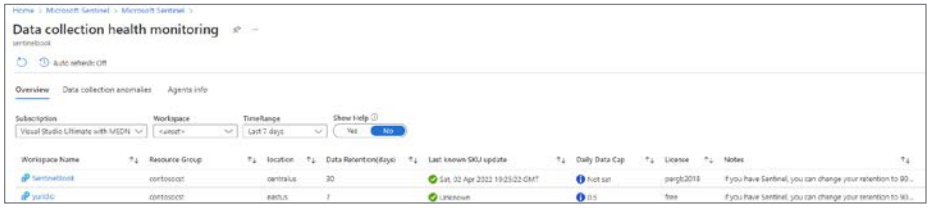


FIGURE 8-5 Insights from the Data Collection Health Monitoring Workbook

7. If your environment has multiple workspaces, this Workbook will retrieve information about the following workspace items:
   - Resource group
   - Geolocation
   - Data retention
   - Last update
   - Daily data cap
   - License
8. You can also use the **TimeRange** option to visualize more or less than 7 days (default selection), customize which **Subscription** you want to focus on, and select the individual workspace.

9. You can also use the **Data Collection Anomalies** tab to detect potential anomalies in the data collection process by table and data source. An example of the results in this tab is shown in Figure 8-6.

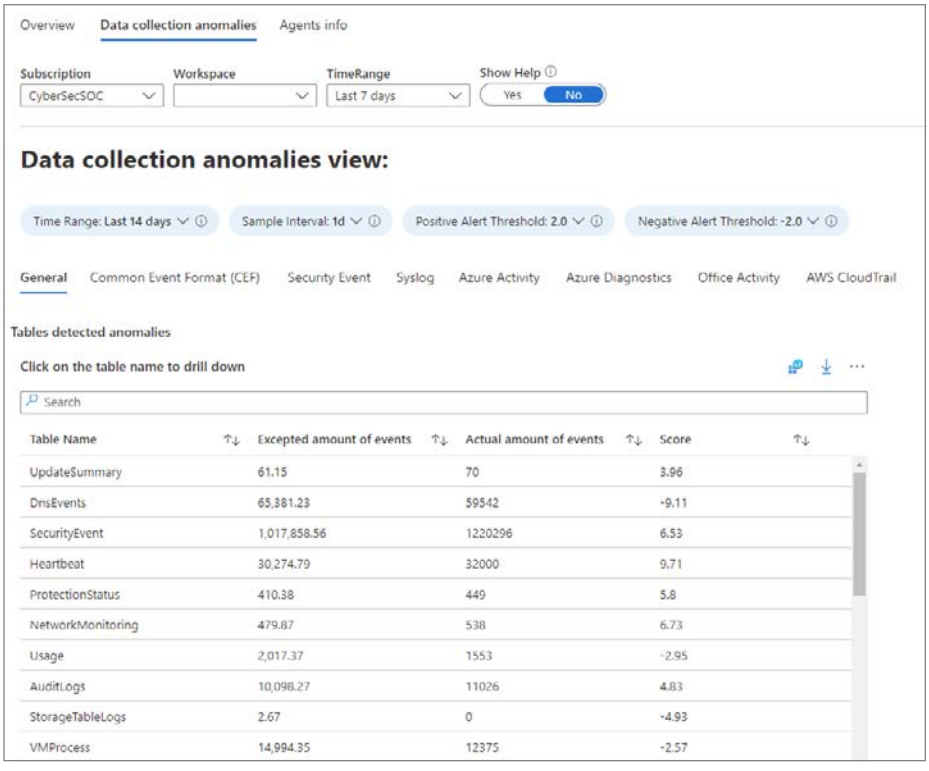


FIGURE 8-6 Data collection anomalies view

10. Each tab presents anomalies for a particular table (the **General** tab includes a collection of tables). The anomalies are calculated using the `series_decompose_anomalies()` function that returns an anomaly score.

> ***NOTE*** For more information about the `series_decompose_anomalies()` function, see *http://aka.ms/SWBAnomalies*.

Creating custom Workbooks

You can also create your own custom Workbooks if the pre-built templates are insufficient for your needs. You can combine text, analytic queries, Azure metrics, and parameters into highly interactive reports. Follow the steps below to create your own Workbook:

1. In the **Microsoft Sentinel** dashboard, select **Workbooks** and then select **Add Workbook** to create a new Workbook from scratch. You will be taken to the **New Workbook** screen, as shown in Figure 8-7.

FIGURE 8-7 New Workbook

2. To edit the Workbook, select **Edit**. In the top-right corner, select the **Edit** button to make changes to the text that was included with the **New Workbook** template. As shown in Figure 8-8, add the following text: **Workbook to Visualize changes in the volume and severity of Security Alerts**. Click the **Done Editing** button to finish.

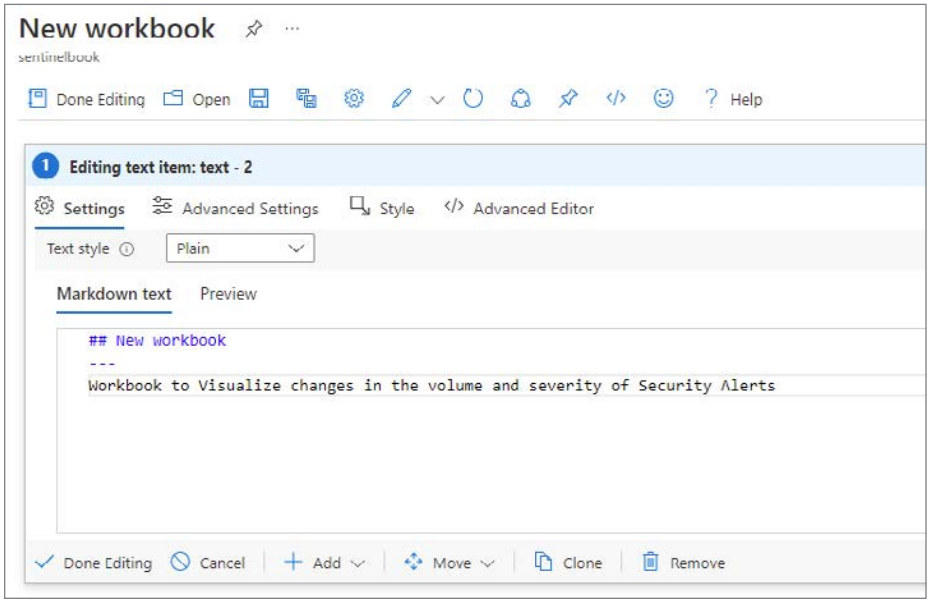


FIGURE 8-8 A view of the Markdown Text To Display screen

3. Now add a pie chart displaying the `Security Events` that have occurred over the last six months, sorted by severity. To do this, select **Edit** at the top of the Workbook. Now, scroll to the right of the screen and select the second **Edit** button. In the **Log Analytics Workspace Logs Query** section, add the following query:

```
SecurityAlert
| where TimeGenerated >= ago(180d)
| summarize Count=count() by AlertSeverity
| render piechart
```

4. You can now test by clicking the **Run Query** button. Although the final numbers in the pie chart might vary, the example shown in Figure 8-9 has the expected output.

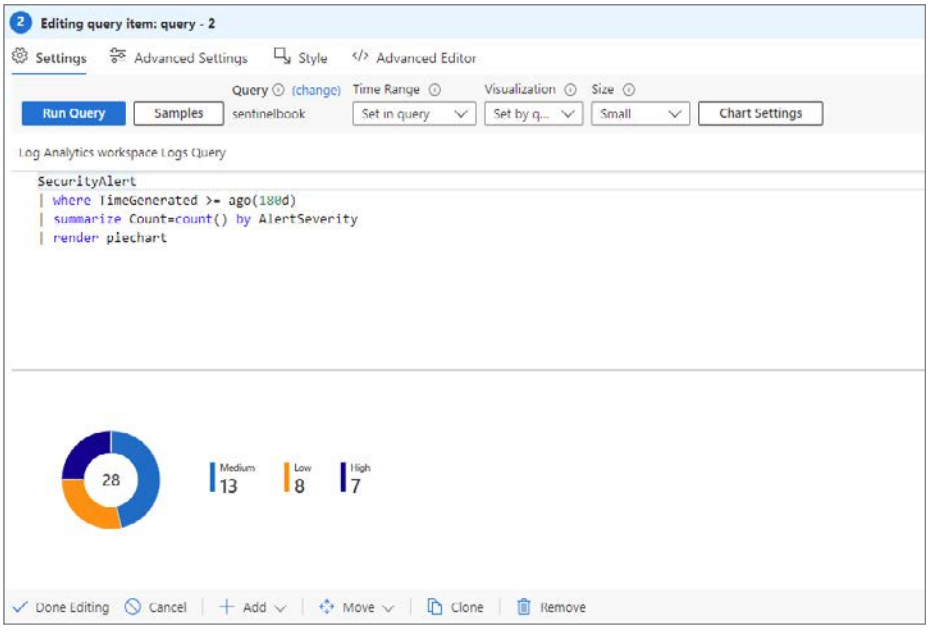


FIGURE 8-9 Changing the graphical representation of the query

5. Click the **Done Editing** button to finish.
6. Now create a new time chart displaying changes in the number of security alerts by severity over the last year. Click the **Edit** button, and then click the **Add** button, followed by the **Add Query** option.
7. In the **Edit Query** window, type the following query:

```
SecurityAlert
| where TimeGenerated >= ago(365d)
| summarize Count=count() by bin(TimeGenerated, 1d), AlertSeverity
```

8. From the **Visualization** dropdown menu, select **Time Chart** and click the **Run Query** button. Figure 8-10 shows an example of how this query will be represented in this graph format.

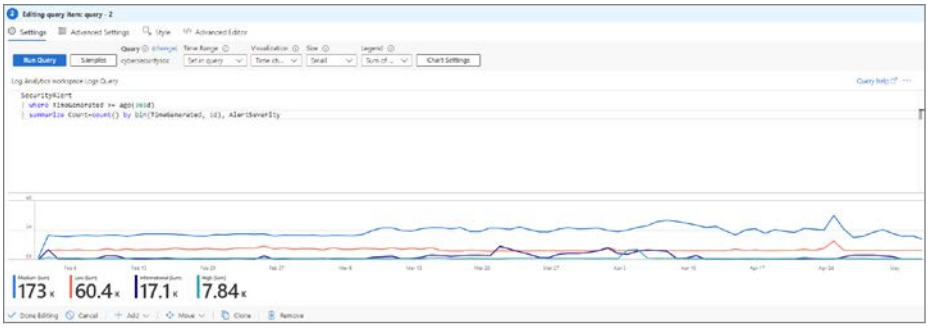


FIGURE 8-10 Changing the visualization for time chart

9. After visualizing, you can click the **Done Editing** button.
10. Now that you have created your new Workbook, save the Workbook by selecting the **Save** button at the top of the screen. You will then be presented with a set of text boxes and dropdown menus, including **Title**, **Save To**, **Subscription**, **Resource Group**, and **Location**. Ensure that you save the new Workbook under your Microsoft Sentinel workspace's subscription and resource group. If you want to let others in your organization use the Workbook, select **Shared Reports** from the **Save To** menu. If you want this Workbook to be available only to you, select **My Reports**, add a meaningful title for your Workbook, and then select **Save**.

Creating visualizations in Power BI and Excel

SOC leaders are often asked to provide metrics and report on their operations to executives and key business partners. Most likely, executives and business partners will not have access to Microsoft Sentinel; therefore, another method must be leveraged to provide them with the information they need.

Creating visualizations in Power BI

Log Analytics provides a native integration with Power BI. You can take any query used in Log Analytics and export it in Power Query language to create a Power BI Dataset. The architecture for exporting Microsoft Sentinel data in Power BI is shown in Figure 8-11.

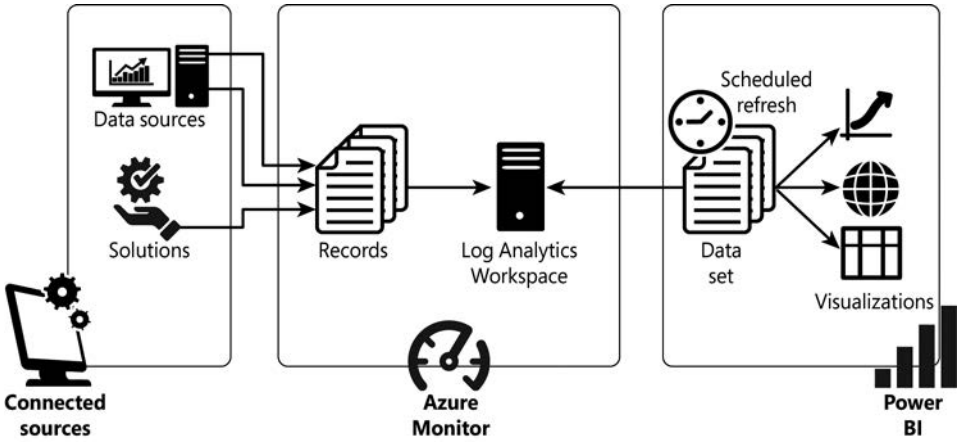


FIGURE 8-11 Architecture for exporting Microsoft Sentinel data to Power BI

To create visualizations in Power BI with Microsoft Sentinel data, you need to perform the following steps:

1. Ensure that you have Power BI Desktop installed on your computer.
2. Next, create a log query within Microsoft Sentinel that returns the data that you want to populate a Power BI dataset. To do this, open the Azure portal and sign in as a user who has either contributor or reader permissions on the resource group to which the Microsoft Sentinel workspace belongs.
3. In the search pane, type **Sentinel** and click the Microsoft Sentinel icon when it appears.
4. Select the workspace on which Microsoft Sentinel has been enabled.
5. Click **Logs** in the left navigation pane and enter the query to retrieve the data you want to share. For example, enter the following query to retrieve all Azure Active Directory audit logs for the last six months:

```
AuditLogs
| where TimeGenerated >= ago(120d)
```

6. Click **Export** at the top of the Query window and then select **Export To Power BI (M Query)**, as shown in Figure 8-12.

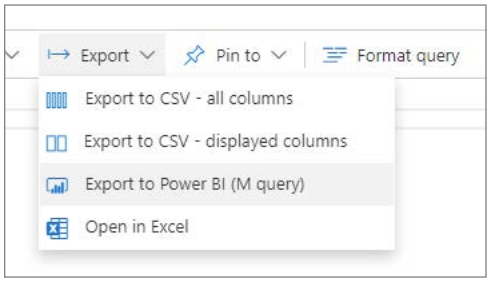


FIGURE 8-12 PowerBI Export menu

7. You will be prompted to open or save the Power BI M query. For demonstration purposes, click **Open**. A Notepad file will open with the M query.
8. Open Power BI Desktop, click **Get Data** > **Blank Query,** and then select **Advanced Editor,** as shown in Figure 8-13. Paste the contents from the exported file into the query window and click **Done**.

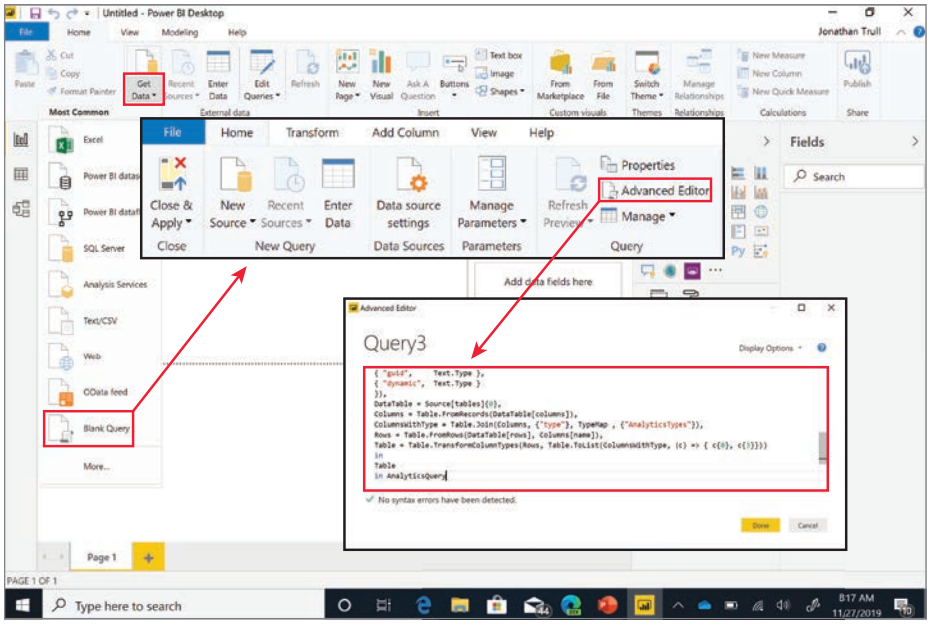


FIGURE 8-13 Power BI Desktop app navigation to the Advanced Editor

9. Click **Close & Apply**.
10. Microsoft Sentinel data is now available within Power BI, and you can create custom reports and share those reports with others within your organization.

Exporting data to Microsoft Excel

You can also easily export your Microsoft Sentinel data to Microsoft Excel to create visualizations and share information. You can use this approach if you need to create custom, one-time reports for individuals. Follow these steps:

1. Open the **Azure portal** and sign in as a user who has either contributor or reader permissions on the resource group to which the Microsoft Sentinel workspace belongs.
2. In the search pane, type **Sentinel** and click the Microsoft Sentinel icon when it appears.
3. Select the workspace in which **Microsoft Sentinel** has been enabled.
4. Select **Logs** and enter the query to retrieve the data you want to share. For example, enter the following query to retrieve all Security Events that have occurred over the last six months and display the alert name, severity level, and whether it was identified as an incident:

```
SecurityAlert
| where TimeGenerated >= ago(120d)
| project AlertName, AlertSeverity, IsIncident
```

5. Select **Run**.
6. Select **Export** at the top of the window, and select **Export To CSV – All Columns**.

Now you can open, save, or share the CSV file and work with the data as needed to create additional reports and visualizations.

Chapter 9

Data connectors

A key requirement for every SIEM is the capability to ingest and process massive amounts of data from various sources—data to analyze, data to run detections on, data to hunt for indicators of compromise, and more. As a cloud-born SIEM, one of Microsoft Sentinel's strengths is to handle terabytes of data with ease, without any scaling or sizing issues for you to worry about. Data can be ingested in several ways, such as by leveraging the following:

- Data connectors, including service-to-service connections
- Rest API endpoints
- Agents, including forwarders and plug-ins, like the output plug-in for Logstash

Data connectors are Microsoft Sentinel's primary tool for ingesting and processing data and should be your first option for data ingestion. This chapter focuses on data connectors and solutions that package content in a single offering.

Understanding data connectors

Microsoft Sentinel comes with many out-of-the-box data connectors, the majority of which can be enabled and configured with a couple of clicks. A key aspect of a data connector is that the data is normalized and comes with content like detection rules, queries, and, optionally, Workbooks.

> ***NOTE*** *Normalization* **is the process of transforming data collected from various sources into a uniform presentation as defined by a standardized schema.**

Follow the steps below to see a full list of out-of-the-box connectors:

1. Open the **Azure portal** and sign in as a user who has Microsoft Sentinel Reader privileges.
2. Navigate to the **Microsoft Sentinel** page.
3. Under **Configuration**, click **Data Connectors**. Doing so shows a list of data connectors, as shown in Figure 9-1.

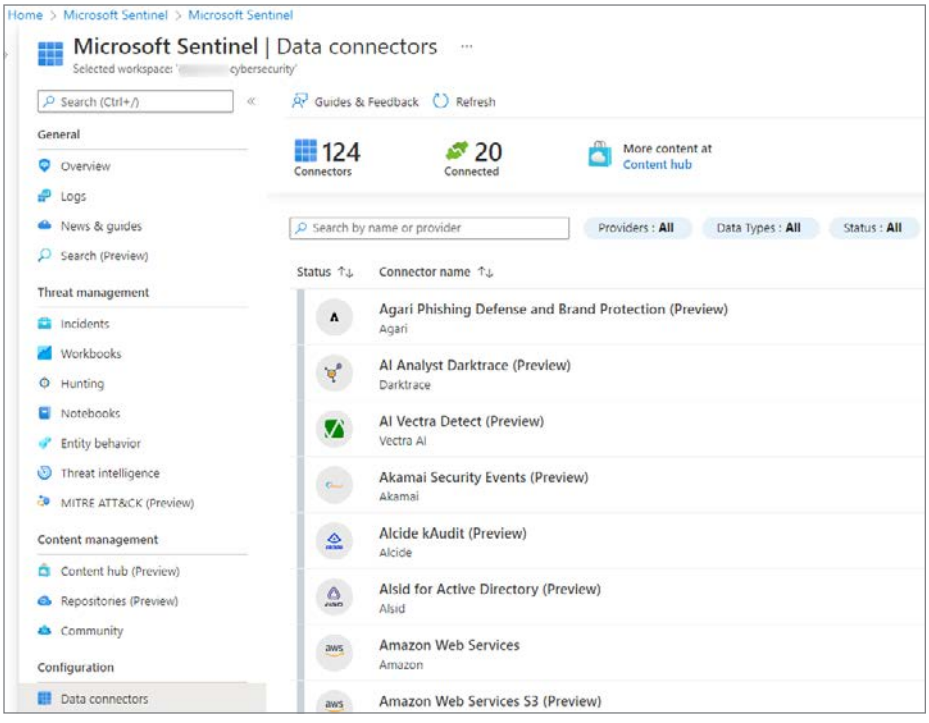


FIGURE 9-1 List of available data connectors

4. In the top of the middle pane, notice the number of available connectors (124, at the time this book was written). Twenty (20) of the connectors are connected, and you can filter by **Providers**, **Data Types**, and **Status.**

5. Each data connector has its own specific permission requirements and configuration steps, which will be shown when you click **Open Connector Page**, as shown in Figure 9-2.

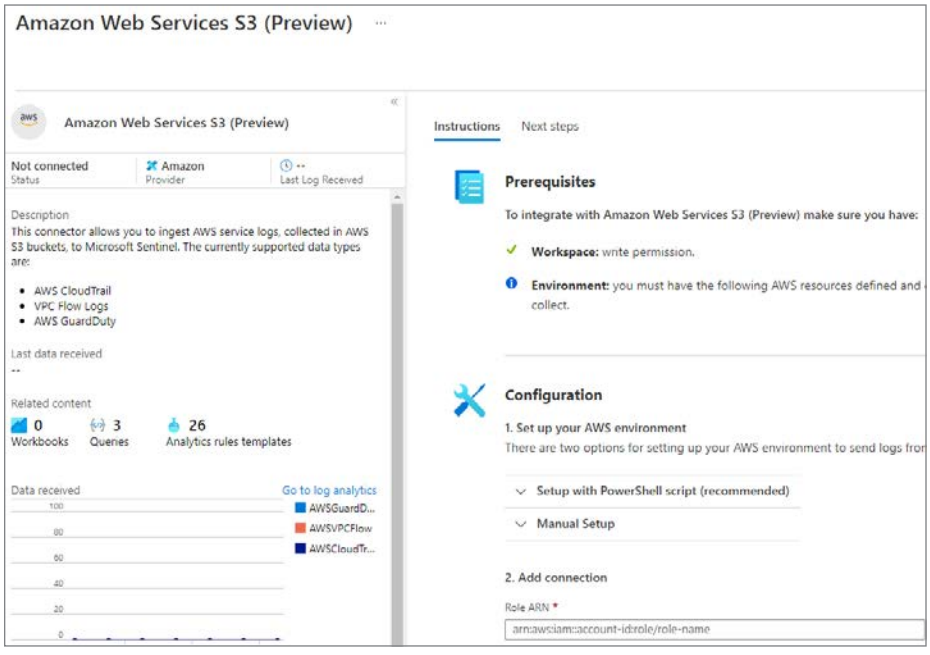


FIGURE 9-2 Amazon Web Services S3 data connector

Ingestion methods

Microsoft Sentinel supports different types of ingestion. Table 9-1 lists the different ingestion methods:

TABLE 9-1 Different types of data connectors by ingestion method

INGESTION METHOD	DATA SOURCES
Azure service-to-service integration	Built-in Azure foundational support, which supports: ■ API-based connections ■ Diagnostics settings connections ■ Agent-based connections
Common Event Format (CEF) over Syslog	Linux machine with the agent installed as a forwarder
Azure functions and REST API	Serverless connector to connect to the REST API endpoints
Syslog	Agent for Linux (`rsyslog/syslog-ng`)
Custom logs	Agent-based API based

The Codeless Connector Platform

Besides the out-of-the-box data connectors and agents, Microsoft Sentinel offers the option to create your own custom connector through the **Codeless Connector Platform** (CCP). Connectors created using CCP are Software as a Service (SaaS)–based, which means that Microsoft runs this connector for you in Azure without any requirements for service installations. Another benefit is that it automatically provides health monitoring for your connector.

In practice, CCP connectors will most likely be authored by managed security service providers (MSSPs), independent software vendors (ISVs), or enterprises that act as MSSPs because there's an authoring effort involved. In its current release, CCP enables you to connect to any data source that exposes a public REST API endpoint. In future releases, you can expect additional features, such as support for more authentication models, pagination types, and more.

The benefits of using a CCP-based connector include:

- Minimal development effort required to connect with publicly exposed REST APIs
- Scalable built-in Poller as a service
- Configurable UI components for your connector
- Health monitoring integration

Preparing for a new data connector

Enabling and configuring a new data connector requires planning and preparation. It starts with validating whether you have the appropriate authorization and authentication information. For example, some connectors require Azure Global Administrator permissions, while others require authorization and authentication configuration steps at the source. Because of new data being ingested, you need to be aware of the potential increases in ingestion and retention costs.

For example, enabling the collection of the Non-Interactive User Sign-In Logs (part of the Azure Active Directory connector) rapidly increases the amount of data being ingested. Although it can be very useful and insightful data, this only makes sense if detection rules for this data type are enabled, and your SOC team is trained to triage and investigate these types of incidents.

> ***TIP*** **Before enabling a connector, explore the potential amount of data the connector generates (such as by leveraging a test environment). Also, determine which Log Analytics table data is being written to and what kind of content comes with the connector. Having detection rules that generate incidents based on the new connector is key.**

Ideally, enabling a data connector should be followed by having the following:

- Detection rules enabled
- Prepared hunting queries

- Workbooks to visualize data with drill-down capabilities
- Relevant SOAR solutions, such as automation rules and Playbooks

Together with training for your SOC teams, this should be part of your strategy for onboarding new data sources.

Enabling and configuring a data connector

The easiest way to enable and configure a data connector is through the Azure portal. Several connectors, such as the Azure resource-based connectors (Azure Active Directory or Azure Activity), can be deployed through PowerShell, Azure Command Line Interface (CLI), APIs, or through Azure Resource Manager (ARM) templates. For example, the following PowerShell script enables the Office 365 data connector for Exchange, SharePoint, and Teams:

```
New-AzSentinelDataConnector -ResourceGroupName $resourceGroupName -WorkspaceName
$workspaceName -Office365 -Exchange Enabled -SharePoint Enabled -Teams Enabled
```

> ***NOTE*** **To configure Microsoft Sentinel with PowerShell, you need to install the Az.SecurityInsights module, which can be installed from *https://www.powershellgallery.com*.**

If you would like to use an ARM template for the same connector, you can leverage the ARM template in Listing 9-1.

LISTING 9-1 ARM template example for the Office 365 data connector

```
{
  "$schema": "https://schema.management.azure.com/schemas/2019-04-01/deploymentTemplate.
json#",
  "contentVersion": "1.0.0.0",
  "parameters": {
    "workspaceName": {
      "type": "string",
      "metadata": {
        "description": "Sentinel workspace name"
      }
    },
    "exchangeState": {
      "type": "string",
      "defaultValue": "enabled",
      "allowedValues": [
        "enabled",
        "disabled"
      ],
      "metadata": {
        "description": "Collect Exchange data (enabled or disabled)."
      }
    },
    "sharePointState": {
      "type": "string",
```

```
        "defaultValue": "enabled",
        "allowedValues": [
          "enabled",
          "disabled"
        ],
        "metadata": {
          "description": "Collect SharePoint data (enabled or disabled)."
        }
      },
      "teamsState": {
        "type": "string",
        "defaultValue": "enabled",
        "allowedValues": [
          "enabled",
          "disabled"
        ],
        "metadata": {
          "description": "Collect Teams data (enabled or disabled)."
        }
      },
      "location": {
        "type": "string",
        "defaultValue": "[resourceGroup().location]"
      },
      "connectorId": {
        "type": "string",
        "defaultValue": "[newGuid()]",
        "metadata": {
          "description": "New autogenerated GUID for the data connector"
        }
      }
    },
    "resources": [
      {
        "type": "Microsoft.OperationalInsights/workspaces/providers/dataConnectors",
        "apiVersion": "2020-01-01",
        "location": "[parameters('location')]",
        "name": "[concat(parameters('workspaceName'),'/Microsoft.SecurityInsights/',parame
ters('connectorId'))]",
        "kind": "Office365",
        "properties": {
          "tenantId": "[subscription().tenantId]",
          "dataTypes": {
            "exchange": {
              "state": "[parameters('exchangeState')]"
            },
            "sharePoint": {
              "state": "[parameters('sharePointState')]"
            },
            "teams": {
              "state": "[parameters('teamsState')]"
            }
          }
        }
      }
    ],
```

```
  "outputs": {
    "connectorId": {
      "type": "string",
      "value": "[parameters('connectorId')]"
    }
  }
}
```

To use the Azure portal instead of PowerShell or an ARM template, follow these steps to enable the Office 365 connector:

1. Open the **Azure portal** and sign in as a user who has Microsoft Sentinel **Contributor** and **Global Administrator** or **Security Administrator** privileges.
2. Navigate to the **Microsoft Sentinel** page.
3. Under **Configuration**, click **Data Connectors**.
4. In the search box, enter **Office**.
5. Select **Office 365** and click **Open Connector Page**.
6. Select **Exchange**, **SharePoint**, and **Teams** and click **Apply Changes**, as shown in Figure 9-3.

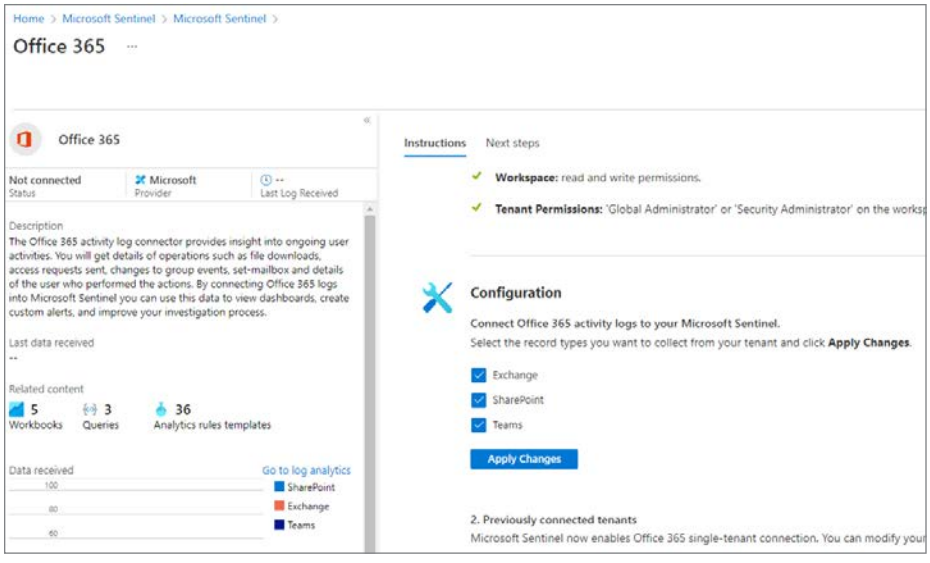


FIGURE 9-3 Office 365 data connector configuration instructions

7. After validation, the connector will change the status to **Connected**; data will start flowing into your workspace shortly.

The Microsoft 365 Defender connector

The new Microsoft 365 Defender connector unifies and natively integrates the following Defender solutions:

- Microsoft Defender for Endpoint
- Microsoft Defender for Identity
- Microsoft Defender for Office 365
- Microsoft Defender for Cloud Apps
- Microsoft Defender Alert Evidence

Eventually, the release of the Microsoft 365 Defender connector (in preview at the time this book was written) will replace all the Defender connectors mentioned in the previous bulleted list. Also, it will allow the connection of incidents and alerts with a single click, as shown in Figure 9-4.

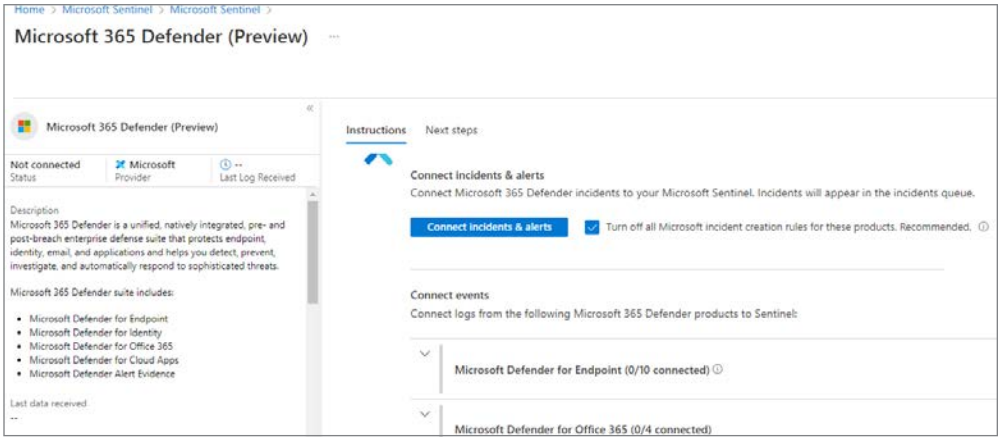


FIGURE 9-4 The new Microsoft 365 Defender connector

The onboarding of most Microsoft connectors, including several third-party connectors, is completed with a couple of clicks. For some connectors—such as the Amazon Web Services S3 connector—additional configuration steps are necessary at the source.

Understanding the Amazon Web Services S3 connector

Microsoft Sentinel's strategy is to provide cross-platform support, which includes applications and services, operating systems, and multi-cloud. Support for Google Cloud Platform (GCP) and Amazon Web Services (AWS) is a key deliverable.

In addition to agent-based collection capabilities—regardless of the cloud platform vendor—Microsoft Sentinel offers a new, improved AWS data connector. By providing access to your AWS service logs, AWS service logs can be ingested into Microsoft Sentinel.

> ***TIP*** **Elaborated step-by-step guidance on how to configure the AWS S3 connector can be found at *https://aka.ms/SentinelAWSconnector*.**

At the time this book was written, two connectors were available for AWS. The new Amazon Web Services S3 connector, as shown in Figure 9-5, will be the eventual successor.

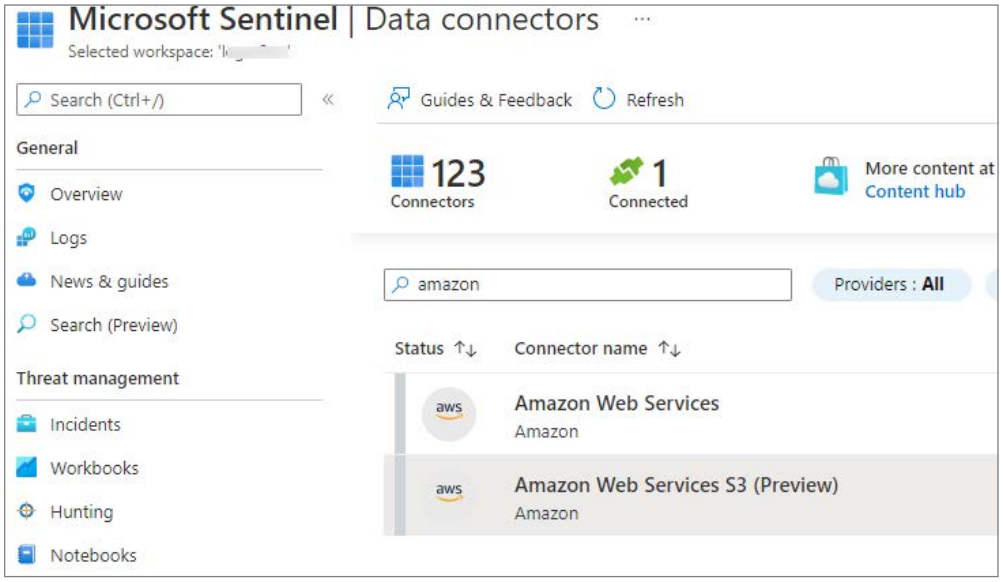


FIGURE 9-5 The new Amazon Web Services S3 connector

This new connector is not affected by the built-in limitations of the `LookupEvents` API. The connector performs better and, therefore, it is the recommended connector going forward.

The AWS S3 connector allows log ingestion into an S3 bucket for the following services:

- **Amazon Virtual Private Cloud (VPC)**—VPC Flow Logs
- **Amazon GuardDuty**—Findings
- **AWS CloudTrail**—Management and data events

The AWS S3 configuration process

The Microsoft Sentinel AWS S3 connector retrieves data from an S3 bucket using a pull mechanism, which requires your AWS services to be configured to send logs to this S3 bucket. When new logs arrive, the following steps occur:

1. The Simple Queue Service (SQS) sends a notification containing the full path to the log files.
2. The Microsoft Sentinel account—part of an assumed role that has been granted permissions to access your AWS resources—regularly polls the SQS and discovers a new notification.
3. Based on the full path, which is contained in the notification, the Microsoft Sentinel account pulls out the data from the S3 bucket and ingests it into your workspace.

The diagram shown in Figure 9-6 summarizes the log collection process.

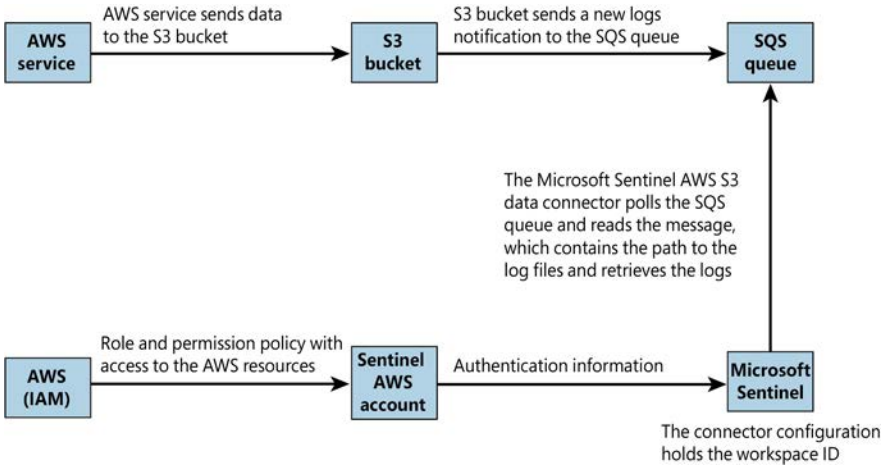


FIGURE 9-6 The AWS S3 connector log collection process

To quickly and efficiently configure the AWS S3 connector, PowerShell scripts are available to configure the necessary AWS services.

> ***NOTE*** **The location of the scripts is linked in the instructions section of the configuration guide at *https://aka.ms/SentinelAWSconnector*. If you want full control of the changes, the same link also provides a walk-through of the manual steps.**

> ***TIP*** **If you are using the PowerShell scripts to configure the AWS S3 data connector (recommended), make sure that you understand what those scripts are executing by opening them in an editor such as VS Code.**

Data connector health monitoring

Any Security Operations Center (SOC) relies on the right data, the quality and depth of the data, and an assurance that the data stream is not interrupted or intermittent. Microsoft Sentinel provides an out-of-the-box Workbook called **Data Collection Health Monitoring,** which provides insights into the data ingested by table size, number of events, events per second (EPS), last time received, and more, as shown in Figure 9-7.

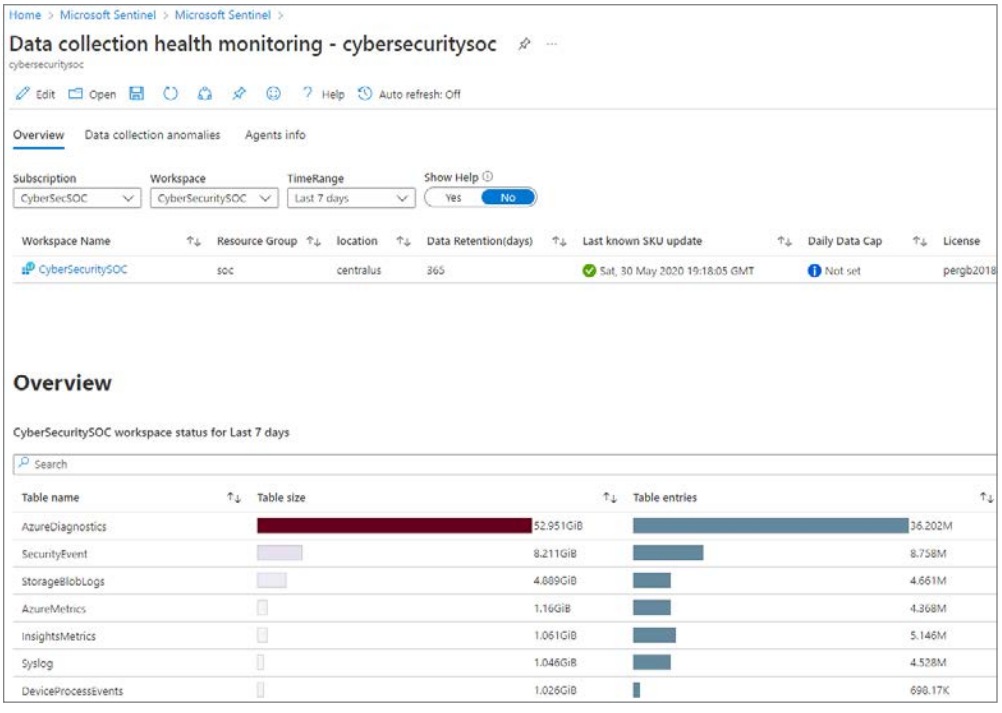


FIGURE 9-7 The Data Collection Health Monitoring Workbook

The **Data Collection Anomalies** tab is very useful for viewing the number of expected events and the actual number of events. This could indicate data connector issues, which can be filtered for all tables (the **General** tab) or per specific tables, as shown in Figure 9-8.

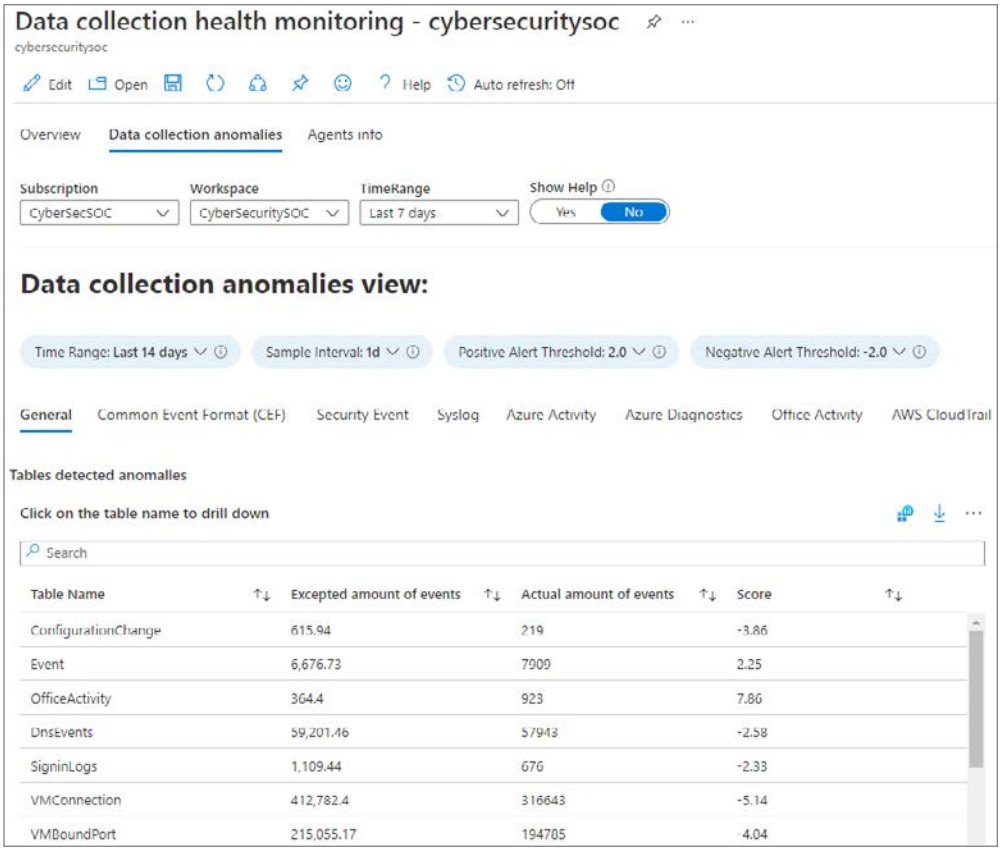


FIGURE 9-8 Data Collection Anomalies View

The **Agents Info** tab shows heartbeat information, which can be helpful in monitoring and troubleshooting the agent connectivity, as well as the heartbeat history, as shown in Figure 9-9.

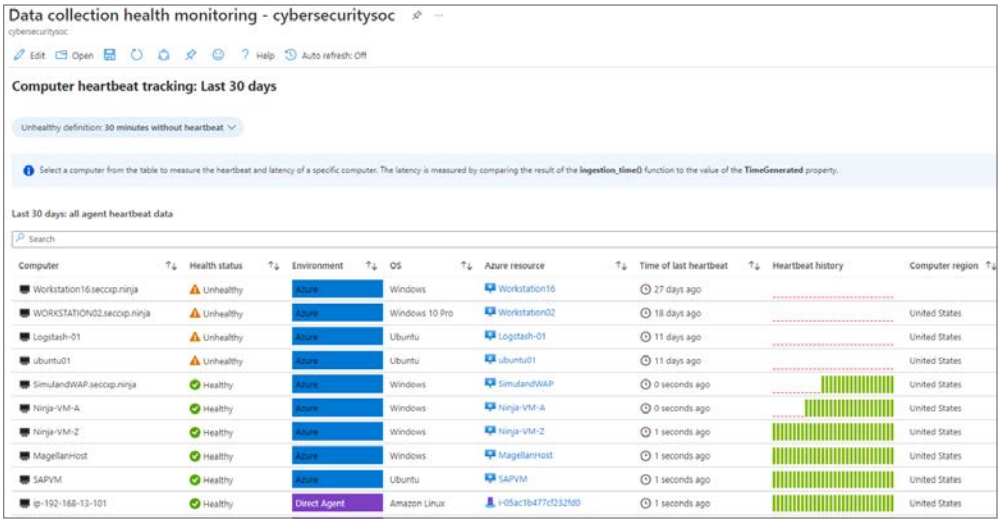


FIGURE 9-9 Agent Info tab, showing the Computer Heartbeat Tracking

NOTE **An agent sends a heartbeat every 60 seconds.**

The Microsoft SentinelHealth table

The new **SentinelHealth** table helps you monitor your connector health, providing insights on health drifts, such as the latest failure events per connector or connector state changes from success to failure. This allows you to create incidents based on data connector failures.

Health insights for out-of-the-box connectors are currently supported for the following tables:

- Amazon Web Services (CloudTrail and S3)
- Dynamics 365
- Office 365
- Office ATP
- Threat Intelligence—TAXII
- Threat Intelligence Platforms

The following query can be used to show data connector failures:

```
SentinelHealth
| where TimeGenerated > ago(3d)
| where OperationName == 'Data fetch status change'
| where Status in ('Success', 'Failure')
| summarize TimeGenerated = arg_max(TimeGenerated,*) by SentinelResourceName,
SentinelResourceId
| where Status == 'Failure'
```

A sample of this query and its result is shown in Figure 9-10.

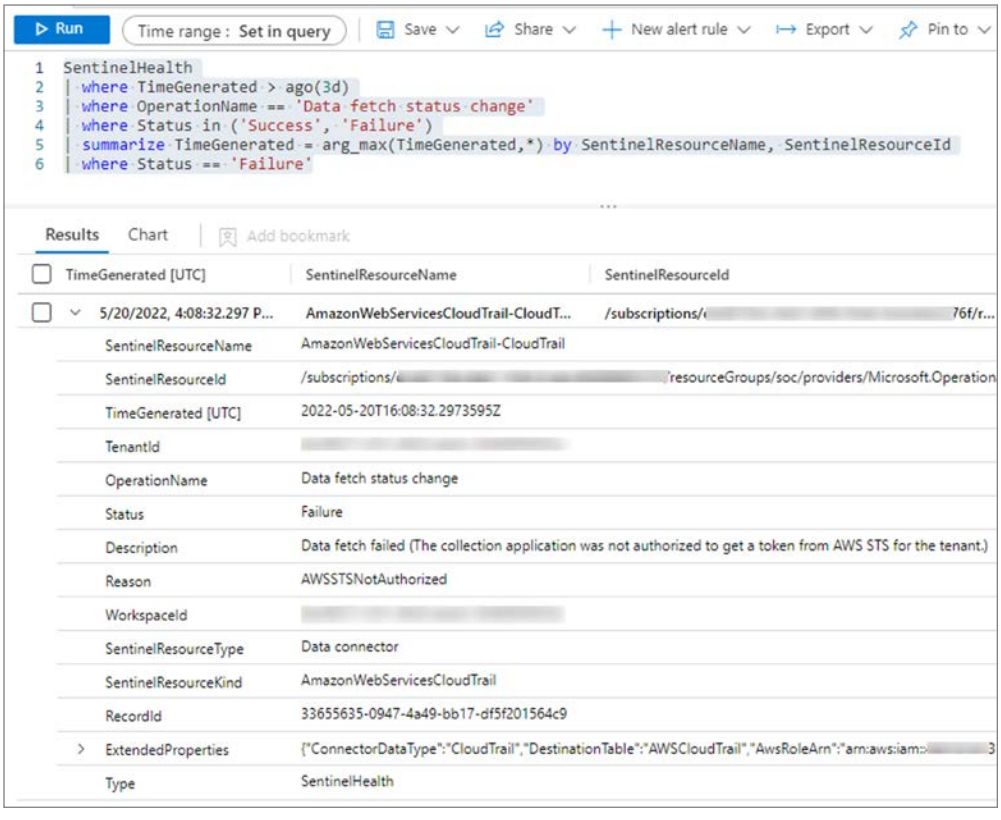


FIGURE 9-10 SentinelHealth table query

The Content Hub

When Microsoft Sentinel was released in March 2019, the out-of-the-box data connectors helped customers quickly and efficiently onboard their data. However, with the rapidly increasing number of data connectors and custom content, challenges surfaced regarding *h*ow to manage and centrally deploy content. This is why repositories were initially developed.

The repositories feature is aimed at supporting a CI/CD source control strategy for GitHub and Azure DevOps for the following content types:

- Analytics rules
- Hunting queries
- Playbooks
- Automation rules
- Parsers
- Workbooks

When content is modified or added to your repository, a workflow automatically deploys the update to your connected workspaces. Figure 9-11 shows a configured repository connection for GitHub.

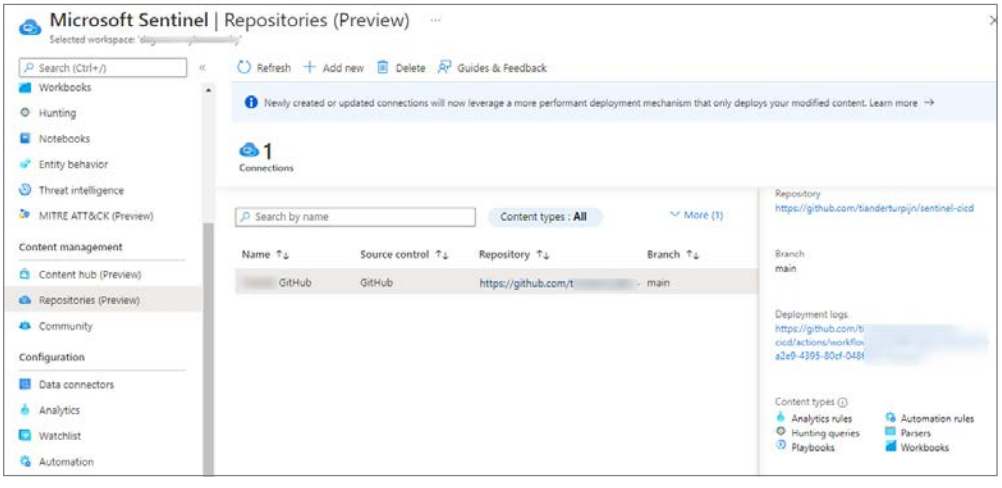


FIGURE 9-11 Repositories connection for GitHub

> ***NOTE*** **Repositories connections created after April 20, 2022, will, by default, use a back-end capability called *Smart deployments*, which only deploys modified content. Prior to April 20, 2022, all content was redeployed with each update. See *https://aka.ms/SentinelRepositories* for more information.**

Although the repositories feature addresses the source control challenge, the Log4J vulnerability made it very clear that better a solution, based on a specific scenario or use case, was necessary. Although Microsoft released Log4J hunting queries and detection rules in less than two days of the exploit discovery, Microsoft Sentinel customers struggled to find the content. The new Content Hub aims to address this gap.

> ***NOTE*** **Microsoft Sentinel solutions are packages of Microsoft Sentinel content or Microsoft Sentinel API integrations that fulfill an end-to-end product, domain, or industry vertical scenario in Microsoft Sentinel.**

Although the Content Hub supports more content types, the main differences between the data connectors page are the search and filter capabilities. The Content Hub also improves how you apply updates to already installed content. Table 9-2 provides a listing of the Content Hub types and what each provides.

TABLE 9-2 Content Hub types

CONTENT TYPE	PROVIDES
Data connectors	Log ingestion from various sources
Parsers	Log formatting/transformation into Advanced Security Information Model (ASIM) formats
Workbooks	Monitoring, visualization, and interactivity
Analytics rules	Creation of incidents
Hunting queries	Proactive queries to hunt for threats
Notebooks	Advanced hunting features in Jupyter and Azure Notebooks
Watchlists	Support for the ingestion of specific data for enhanced threat detection and reduced alert fatigue
Playbooks and Azure Logic Apps custom connectors	Features for automated investigations, remediations, and response scenarios

> ***NOTE*** **More information about the Advanced Security Information Model (ASIM) can be found at *https://aka.ms/sentinelasim*.**

Aside from search capability, the Content Hub provides several useful filtering capabilities, which are shown in Figures 9-12 to 9-17.

The **Status** filter now also shows **New**, **Preview,** and **Update Available** content, as shown in Figure 9-12.

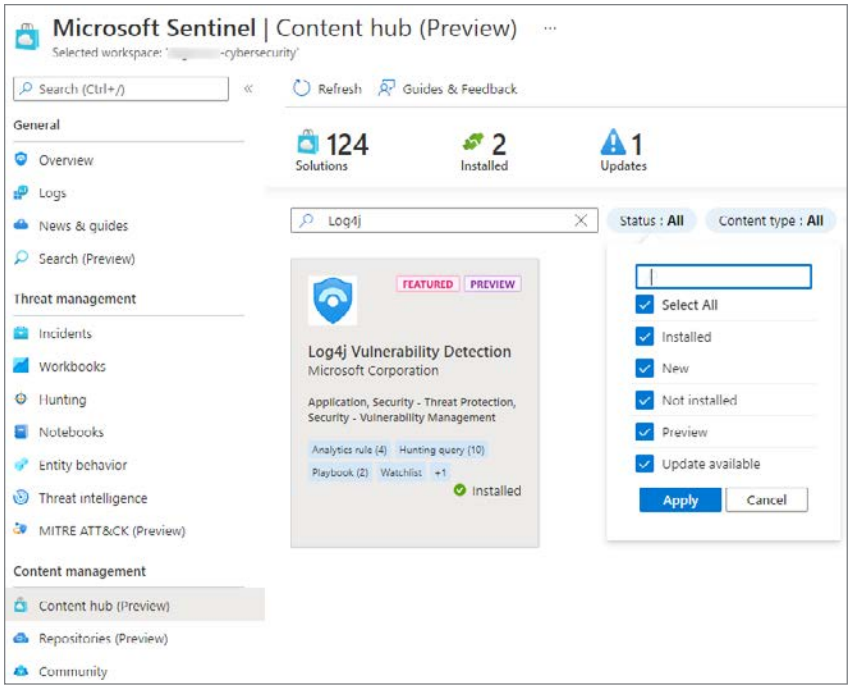


FIGURE 9-12 Content Hub search and status filters

The **Content Type** filter allows you to select different types of content, as shown in Figure 9-13.

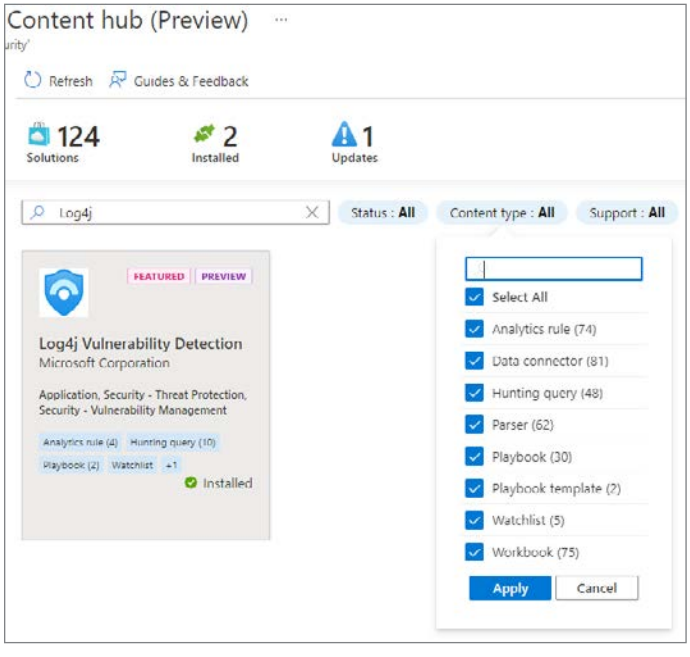


FIGURE 9-13 Content Hub Content Type filters

With the integration of partner and community solutions, being clear on supportability is important if you need to raise a support ticket. Figure 9-14 shows the **Support** filter.

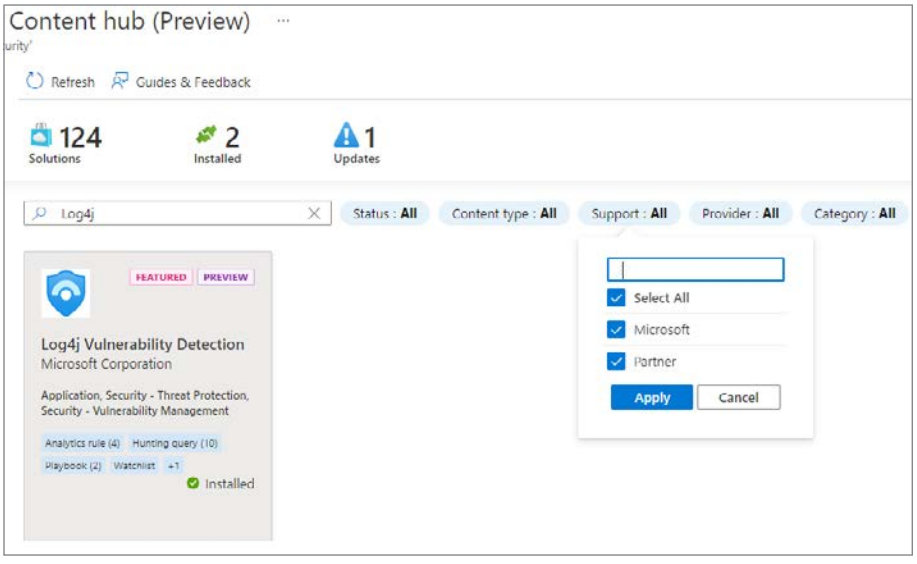


FIGURE 9-14 Content Hub Support filters

The **Provider** filter shows who the solution provider is, as shown in Figure 9-15.

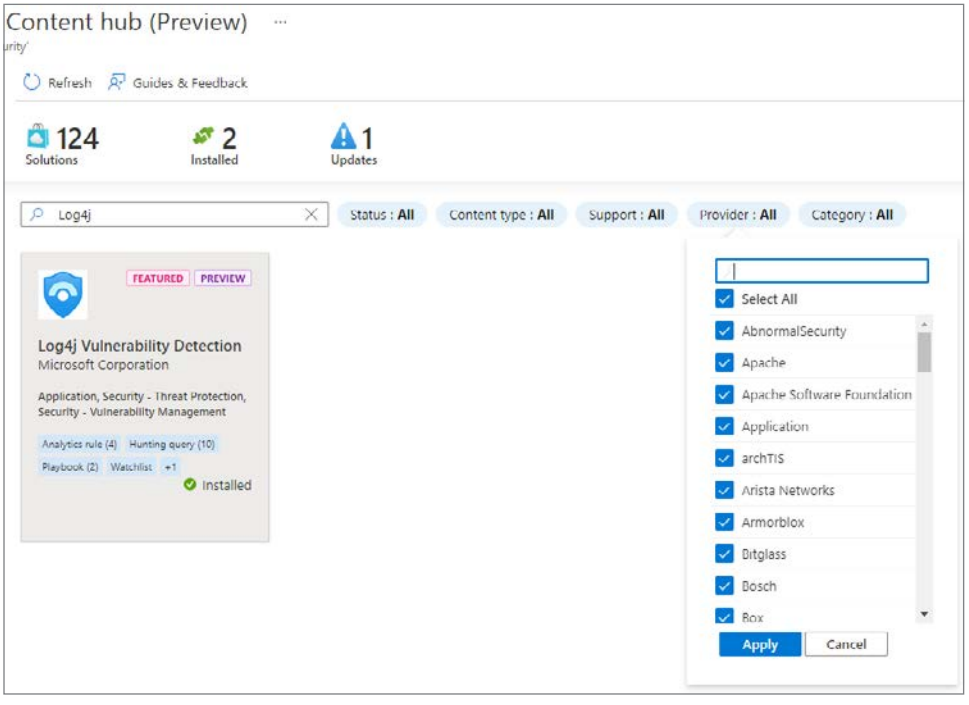


FIGURE 9-15 Content Hub Provider filters

The **Category** filter, as shown in Figure 9-16, is used to filter content per domain.

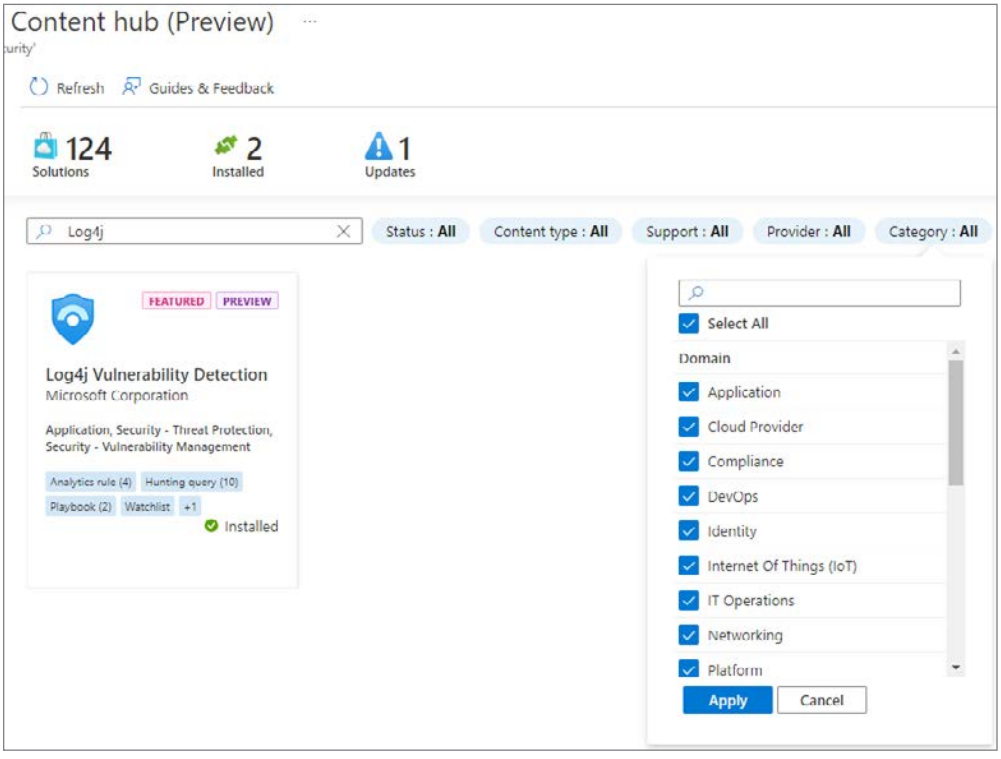


FIGURE 9-16 Content Hub Category filter

When a solution has been previously installed, the solution tile indicates when an update is available. Also, the top icons will show the number of updates available. Using the **Actions** button, you can update the solution, as shown in Figure 9-17.

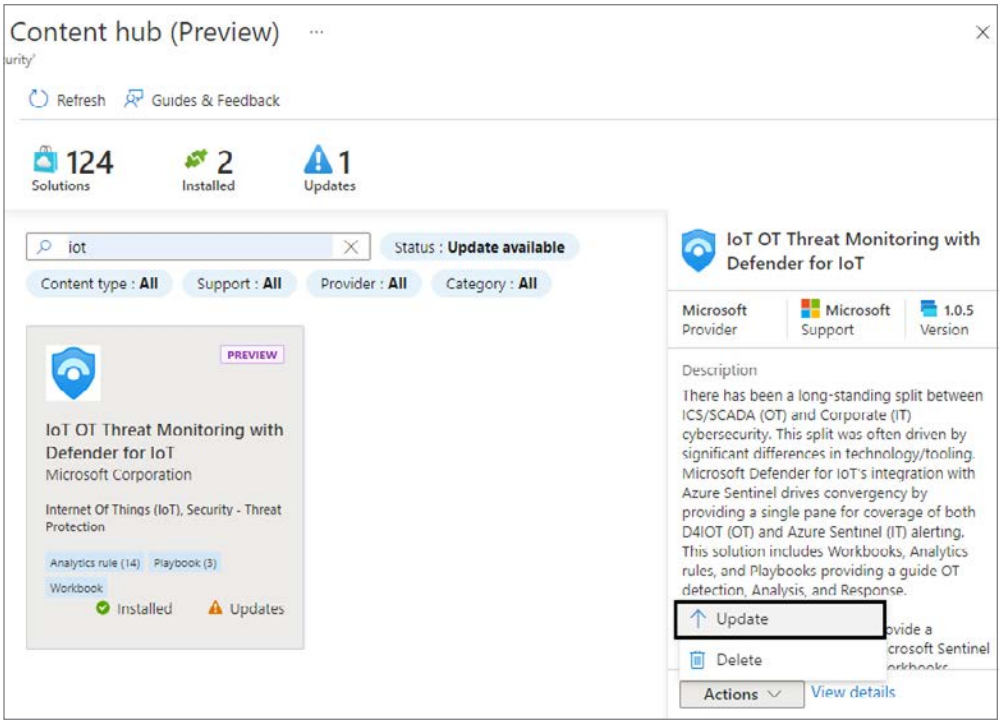


FIGURE 9-17 Content Hub solution update

Over time, the Content Hub will continue to evolve, offering more solutions, more manageability features, and better content integration.

Appendix A

Introduction to Kusto Query Language

By Mike Kassis,
Senior Program Manager
Microsoft CxE Security

The Kusto Query Language, referred to as KQL in this book, is the language you will use to work with and manipulate your data consumed by Microsoft Sentinel. The logs you feed into your workspace aren't worth much if you can't visualize and analyze the important data therein. The best part of KQL is that the power and flexibility of the language is matched by its simplicity. If you have a background in scripting or working with databases, much of what I cover here will feel very familiar. If not, don't worry, you will walk away from this appendix ready to start writing your own queries and driving value for your organization.

This appendix introduces many of the foundational concepts of KQL without getting too bogged down in the details. I will cover some of the most used functions and operators, which should address 75 to 80 percent of the queries you will write day to day. While KQL basics are rather simple, there are times when you will need to run more advanced queries, so I encourage you to carry your learning to more comprehensive resources, such as the official KQL documentation and online courses.

The KQL query structure

A good place to start learning KQL is to develop an understanding of the overall query structure and how it compares to a few other common languages. I have always found that KQL feels like a hybrid of SQL and PowerShell. The former is a mainstay for database administrators, while the latter is the scripting tool of choice for IT operations teams in Windows-heavy environments. Let's start by taking a quick look at SQL.

SQL

Let's start by taking a quick look at SQL where we make use of keywords to structure the query:

1. *SELECT TOP(5)*
2. *Country,*
3. *Count(Country) as CountryCount*
4. *FROM contact*
5. *WHERE Country IS NOT NULL*
6. *GROUP BY Country*
7. *ORDER BY CountryCount DESC*

The *SELECT* and *FROM* keywords let us detail which variables we want returned, how many records we want returned, and from what table they should be taken. The *WHERE* keyword on line 5 lets us filter the dataset based on one or more variables. We use the *GROUP* keyword to say that we want to summarize our data in some way. In this case, we used the *count()* function on line 3, so we are summarizing the count of records associated with each country. Finally, we can sort our data by using the *ORDER* keyword.

In the case of SQL, the structure of the query is largely determined by the keywords and the text included with the keywords. Notice that some things seem to happen in a non-intuitive order. For example, we specified we wanted the top 5 results in line 1, but SQL won't use that information until the very end of the query where it will only keep 5 records. Wouldn't it make more sense to specify *TOP(5)* at the end of the query?

Also, another minor annoyance about SQL's structure is that we had to specify how we wanted to summarize our data in two places. On line 3, we needed our aggregation function, and on line 6, we had to specify what value we wanted that function to summarize *by*. In KQL, we can do all of this in one line, as we'll see in a moment.

PowerShell

Let's look at PowerShell now, which is not a DBA-centric language, but it still serves an important purpose for retrieving and manipulating data.

1. *Get-Process* | `
2. *Where CPU -gt 100* | `
3. *Group ProcessName* | `
4. *Sort Count -descending* | `
5. *Select Count, Name -first 5*

I broke this query into multiple lines (using the backtick character) for readability, but think for a moment how this example varies from the SQL example. The first thing that I notice is the use of the pipe symbol (|). The structure of a PowerShell command is one where you pass your data across a "pipeline," and each step provides some level of processing. At the end of the pipeline, you will get your final result. In effect, this is our pipeline:

```
Get Data | Filter | Summarize | Sort | Select
```

I would argue that this concept of passing data down the pipeline for further processing is a more intuitive structure than what we saw with SQL because it is easier to create a mental picture of your data at each step. We know that on line 1, our pipeline contains every process running on the system. We know that at line 2, we are only keeping processes that have a CPU time that is more than 100 seconds. On line 3, we know that we are summarizing our data to show the count of processes by the process name. Finally, on lines 5 and 6, we know that the data has been sorted, and we only kept the rows we want.

Obviously, SQL and PowerShell serve two very different purposes, but as we look at KQL's query structure, you should notice how it seamlessly combines much of the best components of each language into something that is simplistic, flexible and, most importantly, intuitive.

Here is a look at a KQL query, which looks at Azure Active Directory (AAD) sign-in logs. As you read through each line, you should start to see the SQL and PowerShell similarities quite clearly.

1. *SigninLogs*
2. *| evaluate bag_unpack(LocationDetails) //Don't worry about this line for now.*
3. *| where RiskLevelDuringSignIn == 'none'*
4. *and TimeGenerated >= ago(7d)*
5. *| summarize Count = count() by city*
6. *| order by Count desc*
7. *| take 5*

The use of the pipe symbol between each step works much the same way as we saw with PowerShell. We are passing our set of data down the "pipeline," and at each step, we have a keyword, like SQL, in which we can specify the type of processing we want done. One of the best parts of KQL is that within reason, you can make the steps happen in any order you choose. The pipeline for our above example looks like this:

```
Get Data | Filter | Summarize | Sort | Select

        Get Data = Line 1
        Filter = Lines 3 and 4
        Summarize = Line 5
        Sort = Line 6
        Select = Line 7
```

Like most languages, however, the more flexible the language is, the more prone to mistakes and performance issues it can be; KQL is no exception. The order of the steps we used above can easily be rearranged, but depending on the order, you may get better or worse query performance. A good rule of thumb is to filter your data early, so you are only passing relevant data down the pipeline. This will drastically increase performance and ensure that you aren't accidentally including irrelevant data in summarization steps.

Hopefully, you now have an appreciation for the overall *structure* of a KQL query. Now let's look at the actual KQL operators themselves, which are used to create a KQL query.

NOTE KQL has both tabular and scalar operators. In the remainder of this appendix, if you simply see the word "operator," you can assume it means *tabular operator,* unless otherwise noted.

Data types

Before we get into the actual KQL operators, let's first touch on data types. As in most languages, the data type determines what calculations and manipulations can be run against a value. For example, if you have a value that is of type *string*, you won't be able to perform arithmetic calculations against it.

In KQL, most of the data types follow traditional names you are used to seeing, but there are a few that you might not have seen before such as *dynamic* and *timespan*. Table A-1 provides a look at the full list:

TABLE A-1 Data Type Table

TYPE	ADDITIONAL NAME(S)	EQUIVALENT .NET TYPE
bool	Boolean	System.Boolean
datetime	Date	System.DateTime
dynamic		System.Object
guid	uuid, uniqueid	System.Guid
int		System.Int32
long		System.Int64
real	Double	System.Double
string		System.String
timespan	Time	System.TimeSpan
decimal		System.Data.SqlTypes.SqlDecimal

While most of the data types are standard, *dynamic*, *timespan*, and *guid* are less commonly seen.

Dynamic has a structure very similar to JSON (Javascript Object Notation) with one key difference: It can store KQL-specific data types that traditional JSON cannot, such as a nested dynamic value or timespan. Here's an example of a dynamic type:

```
{
"countryOrRegion":"US",
"geoCoordinates": {
"longitude":-122.12094116210936,
"latitude":47.68050003051758
},
"state":"Washington",
"city":"Redmond"
}
```

Timespan is a data type that refers to a measure of time such as hours, days, or seconds. Do not confuse *timespan* with *datetime*, which is an actual date and time, not a measure of time. Table A-2 shows a list of timespan suffixes.

Table A-2 Timespan suffixes

FUNCTION	DESCRIPTION
`D`	`days`
`H`	`hours`
`M`	`minutes`
`S`	`seconds`
`Ms`	`milliseconds`
`Microsecond`	`microseconds`
`Tick`	`nanoseconds`

Guid is a datatype representing a 128-bit, globally-unique identifier, which follows the standard format of *[8]-[4]-[4]-[4]-[12]*, where each *[number]* represents the number of characters and each character can range from 0-9 or a-f.

Getting, limiting, sorting, and filtering data

When learning any new language, we want to start with a solid foundation. For KQL, this foundation is a collection of operators that will let you start to filter and sort your data. What's great about KQL is that these handful of commands and operators will make up about 75 percent of the querying you will ever need to do. The remaining 25 percent will be stretching the language to meet your more advanced needs. Let's expand a bit on some of the commands we used in our above example and look at *take*, *order*, and *where*.

For each operator, we will examine its use in our previous *SigninLogs* example. Additionally, for each operator, I'll provide either a useful tip or a best practice.

Getting data

The first line of any basic query in KQL specifies which table you want to work with. In the case of Microsoft Sentinel, this will likely be the name of a log type in your workspace, such as *SigninLogs*, *SecurityAlert*, or *ThreatIntelligenceIndicator*. For example:

```
SigninLogs
```

Note that log names are case sensitive, which is true about KQL in general, so *SigninLogs* and *signinLogs* will be interpreted differently. Take care when choosing names for your custom logs, so they are easily identifiable and are not too similar to another log.

Limiting data: *take*

The *take* operator is used to limit your results by the number of rows returned. It accepts an integer to determine the number of rows returned. Typically, it is used at the end of a query after you have determined your sort order.

Using *take* earlier in the query can be useful for limiting large datasets for testing; however, you run the risk of unintentionally excluding records from your dataset if you have not determined the sort order for your data, so take care. Here's an example of using *take*:

```
SigninLogs
      | take 5
```

> ***TIP*** **When working on a brand-new query where you may not know what the query will look like, it can be useful to put a *take* statement at the beginning to artificially limit your dataset for faster processing and experimentation. Once you are happy with the full query, you can remove the initial *take* step.**

Sorting data: *order*

The *order* operator is used to sort your data by a specified column. For example, here we ordered the results by *TimeGenerated* and we set the order direction to descending (*desc*), which will place the highest values first; the inverse being ascending, which is denoted as *asc*.

```
SigninLogs
| order by TimeGenerated desc
| take 5
```

Note that we put the *order* operator before the *take* operator. We need to sort first to make sure we get the appropriate five records.

In cases where two or more records have the same value in the column you are sorting by, you can be explicit in how the query handles these situations by adding a comma-separated list of variables after the *by* keyword, but before the sort order keyword (*desc*), like so:

```
SigninLogs
| order by TimeGenerated, Identity desc
| take 5
```

Now, if *TimeGenerated* is the same between multiple records, it will then try to sort by the value in the *Identity* column.

Filtering data: *where*

The *where* operator is arguably the most important operator because it is key to making sure you are only working with the subset of data that is valuable to your use case. You should do your best to filter your data as early in the query as possible because doing so will improve query performance by reducing the amount of data that needs to be processed in subsequent steps; it also ensures that you are only performing calculations on the desired data. See this example:

```
SigninLogs
| where TimeGenerated >= ago(7d)
| order by TimeGenerated, Identity desc
| take 5
```

The *where* operator accepts the name of a variable, a comparison (*scalar*) operator, and a value. In our case, we used >= to denote that the value in the *TimeGenerated* column needs to be greater than or equal to (later than) seven days ago.

There are two types of comparison operators in KQL: string and numerical. Table A-3 shows the full list of numerical operators:

Table A-3 Numerical operators

OPERATOR	DESCRIPTION
+	Add
-	Subtract
*	Multiply
/	Divide
%	Modulo
<	Less
>	Greater
==	Equals
!=	Not equals
<=	Less or Equal
>=	Greater or Equal
in	Equals to one of the elements
!in	Not equals to any of the elements

However, the list of string operators is a much longer list because it has permutations for case sensitivity, substring locations, prefixes, suffixes, and much more. Note, the == operator is both a numeric and string operator, meaning it can be used for both numbers and text. For example, both of the following statements would be valid *where* statements:

```
| where ResultType == 0
| where Category == 'SignInLogs'
```

Best Practice: Almost certainly, you will want to filter your data by more than one column or filter the same column in more than one way. In these instances, there are two best practices you should keep in mind.

1. You can combine multiple *where* statements into a single step by using the *and* keyword. For example

```
SigninLogs
| where Resource == ResourceGroup
    and TimeGenerated >= ago(7d)
```

2. When you have multiple *where* clauses joined with the *and* keyword, like above, you will get better performance by putting clauses that only reference a single column first. So, a better way to write the above query would be:

```
SigninLogs
| where TimeGenerated >= ago(7d)
    and Resource == ResourceGroup
```

Summarizing data

Summarizing is one of the most important tabular operators in KQL, but it also is one of the more complex operators to learn if you are new to query languages in general. The job of *summarize* is to take in a table of data and output a *new table* that is aggregated by one or more columns.

Structure of the summarize statement

The basic structure of a *summarize* statement is as follows:

```
| summarize <aggregation> by <column>
```

For example, the following would return the count of records for each *CounterName* value in the *Perf* table:

```
Perf
| summarize count() by CounterName
```

Because the output of *summarize* is a *new* table, any columns not explicitly specified in the *summarize* statement will not be passed down the pipeline. To illustrate this concept, consider this example:

```
Perf
| project ObjectName, CounterValue , CounterName
| summarize count() by CounterName
| order by ObjectName asc
```

On the second line, we are specifying that we only care about the columns *ObjectName*, *CounterValue*, and *CounterName*. We then summarized to get the record count by *CounterName* and finally, we attempt to sort the data in ascending order based on the *ObjectName* column. Unfortunately, this query will fail with an error indicating that the *ObjectName* is unknown. This is because when we summarized, we only included the *Count* and *CounterName* columns in our new table. To fix this, we can simply add *ObjectName* to the end of our summarize step, like this:

```
Perf
| project ObjectName, CounterValue , CounterName
| summarize count() by CounterName, ObjectName
| order by ObjectName asc
```

The way to read the *summarize* line in your head would be: "summarize the count of records by *CounterName*, and group by *ObjectName*". You can continue adding comma-separated columns to the end of the *summarize* statement.

Building on the previous example, if we want to aggregate multiple columns at the same time, we can achieve this by adding a comma-separated list of aggregations. In the example below, we are getting a sum of the *CounterValue* column in addition to getting a count of records:

```
Perf
| project ObjectName, CounterValue , CounterName
| summarize count(), sum(CounterValue) by CounterName, ObjectName
| order by ObjectName asc
```

This seems like a good time to talk about column names for these aggregated columns. At the start of this section, we said the *summarize* operator takes in a table of data and produces a new table, and only the columns you specify in the *summarize* statement will continue down the pipeline. Therefore, if you were to run the above example, the resulting columns for our aggregation would be *count_* and *sum_CounterValue*.

The KQL engine will automatically create a column name without us having to be explicit, but often, you will find that you will prefer your new column have a friendlier name. To do this, you can easily name your column in the *summarize* statement, like so:

```
Perf
| project ObjectName, CounterValue , CounterName
| summarize Count = count(), CounterSum = sum(CounterValue) by CounterName, ObjectName
| order by ObjectName asc
```

Now, our summarized columns will be named *Count* and *CounterSum*.

There is much more to the *summarize* operator than we can cover in this short section, but I encourage you to invest the time to learn it because it is a key component to any data analysis you plan to perform on your Microsoft Sentinel data.

Aggregation reference

The are many aggregation functions, but some of the most commonly used are *sum()*, *count()*, and *avg()*. Table A-4 shows the full list.

Table A-4 Aggregation Functions

FUNCTION	DESCRIPTION
any()	Returns random non-empty value for the group
arg_max()	Returns one or more expressions when argument is maximized
arg_min()	Returns one or more expressions when argument is minimized
avg()	Returns average value across the group
buildschema()	Returns the minimal schema that admits all values of the dynamic input
count()	Returns count of the group
countif()	Returns count with the predicate of the group
dcount()	Returns approximate distinct count of the group elements
make_bag()	Returns a property bag of dynamic values within the group
make_list()	Returns a list of all the values within the group
make_set()	Returns a set of distinct values within the group
max()	Returns the maximum value across the group
min()	Returns the minimum value across the group
percentiles()	Returns the percentile approximate of the group
stdev()	Returns the standard deviation across the group
sum()	Returns the sum of the elements withing the group
variance()	Returns the variance across the group

Adding and removing columns

As you start working more with KQL, you will find that you either have more columns than you need from a table, or you need to add a new calculated column. Let's look at a few of the key operators for column manipulation.

Project and project-away

Project is roughly equivalent to many languages' *select* statements. It allows you to choose which columns to keep. The order of the columns returned will match the order of the columns you list in your project statement, as shown in this example:

```
Perf
| project ObjectName, CounterValue , CounterName
```

As you can imagine, when you are working with very wide datasets, you may have lots of columns you want to keep, and specifying them all by name would require a lot of typing. For those cases, you have *project-away*, which lets you specify which columns to remove, rather than which ones to keep, like so:

```
Perf
| project-away MG, _ResourceId, Type
```

> ***TIP*** **It can be useful to use *project* in two locations in your queries, both at the beginning as well as the end. Using *project* early in your query can provide you with performance improvements by stripping away large chunks of data you don't need to pass down the pipeline. Using it at the end lets you strip away any columns that may have been created in previous steps and you do not need in your final output.**

Extend

Extend is used to create a new calculated column. This can be useful when you want to perform a calculation against existing columns and see the output for every row. Let's look at a simple example where we calculate a new column called *Kbytes*, which we can calculate by multiplying the MB value by 1,024.

```
Usage
| where QuantityUnit == 'MBytes'
| extend KBytes = Quantity * 1024
| project ResourceUri, MBytes=Quantity, KBytes
```

On the final line in our *project* statement, we renamed the *Quantity* column to *Mbytes*, so we can easily tell which unit of measure is relevant to each column. It is worth noting that *extend* also works with previously calculated columns. For example, we can add one more column called *Bytes* that is calculated from *Kbytes*:

```
Usage
| where QuantityUnit == 'MBytes'
| extend KBytes = Quantity * 1024
| extend Bytes = KBytes * 1024
| project ResourceUri, MBytes=Quantity, KBytes, Bytes
```

Joining tables

Much of your work in Microsoft Sentinel can be carried out by using a single log type, but there are times when you will want to correlate data together or perform a lookup against another set of data. Like most query languages, KQL offers a few operators used to perform various types of joins. In this section, we will look at the most-used operators, *union* and *join*.

Union

Union simply takes two or more tables and returns all the rows. For example:

```
OfficeActivity
| union SecurityEvent
```

This would return all rows from both the *OfficeActivity* and *SecurityEvent* tables. *Union* offers a few parameters that can be used to adjust how the union behaves. Two of the most useful are *withsource* and *kind*:

```
OfficeActivity
| union withsource = SourceTable kind = inner SecurityEvent
```

The parameter *withsource* lets you specify the name of a new column whose value will be the name of the source table from which the row came. In the example above, we named the column *SourceTable*, and depending on the row, the value will either be *OfficeActivity* or *SecurityEvent*.

The other parameter we specified was *kind*, which has two options: *inner* or *outer*. In the example we specified *inner*, which means the only columns that will be kept during the union are those that exist in both tables. Alternatively, if we had specified *outer* (which is the default value), then all columns from both tables would be returned.

Join

Join works similarly to *union*, except instead of joining tables to make a new table, we are joining *rows* to make a new table. Like most database languages, there are multiple types of *joins* you can perform. The general syntax for a *join* is:

```
T1
| join kind = <join type>
(
                T2
) on $left.<T1Column> == $right.<T2Column>
```

After the *join* operator, we specify the *kind* of join we want to perform followed by an open parenthesis. Within the parentheses is where you specify the table you want to join as well as any other query statements you wish to add. After the closing parenthesis, we use the *on* keyword followed by our left (*$left*) and right (*$right*) columns separated with a ==. Here's an example of an inner *join*:

```
OfficeActivity
| where TimeGenerated >= ago(1d)
    and LogonUserSid != ''
| join kind = inner (
    SecurityEvent
    | where TimeGenerated >= ago(1d)
        and SubjectUserSid != ''
) on $left.LogonUserSid == $right.SubjectUserSid
```

NOTE **If both tables have the same name for the columns on which you are performing a *join*, you don't need to use *$left* and *$right*; instead, you can just specify the column name. Using *$left* and *$right*, however, is more explicit and generally considered to be a good practice.**

For your reference, Table A-5 shows a list of available types of *joins*.

Table A-5 Types of Joins

JOIN TYPE	DESCRIPTION
`inner`	One row returned for each combination of matching rows.
`innerunique`	Inner join with left side deduplication. (Default)
`leftouter/rightouter`	For a leftouter join, this would return matched records from left table and all records from right, matching or not. Unmatched values will be null.
`fullouter`	Returns all records from both left and right tables, matching or not. Unmatched values will be null.
`leftanti/rightanti`	For a leftanti join, this would return records that did not have a match in the right table. Only columns from the left table will be returned.
`leftsemi/rightsemi`	For a leftanit join, this would return records that had a match in the right table. Only columns from the left table will be returned.

TIP **It is best practice to have your smallest table on the left. In some cases, following this rule will give you huge performance benefits, depending on the types of joins you are performing and the size of the tables.**

Evaluate

You may remember that in the first KQL example, I used the *evaluate* operator on one of the lines. The *evaluate* operator is less commonly used than the ones we have touched on previously. However, knowing how the *evaluate* operator works is well worth your time. Once more, here is that first query, where you will see *evaluate* on the second line.

```
SigninLogs
| evaluate bag_unpack(LocationDetails)
| where RiskLevelDuringSignIn == 'none'
   and TimeGenerated >= ago(7d)
| summarize Count = count() by city
| order by Count desc
| take 5
```

This operator allows you to invoke available plug-ins (essentially service-side functions). Many of these plug-ins are focused around data science, such as *autocluster*, *diffpatterns*, and *sequence_detect*. Some plug-ins, like *R* and *python,* allow you to run scripts in those languages within your queries.

The plug-in used in the above example was called *bag_unpack*, and it makes it very easy to take a chunk of dynamic data and convert it to columns. Remember, dynamic data is a data type that looks very similar to JSON, as shown in this example:

```
{
“countryOrRegion”:”US”,
“geoCoordinates”: {
“longitude”:-122.12094116210936,
“latitude”:47.68050003051758
},
“state”:”Washington”,
“city”:”Redmond”
}
```

In this case, I wanted to summarize the data by city, but *city* is contained as a property within the *LocationDetails* column. To use the city property in my query, I had to first convert it to a column using *bag_unpack*.

Let statements

Now that we have covered many of the major KQL operators and data types, let’s wrap up with the *let* statement, which is a great way to make your queries easier to read, edit, and maintain.

If you are familiar with programming languages and setting variables, *let* works much the same way. *Let* allows you to bind a name to an expression, which could be a single value or a whole query. Here is a simple example:

```
let daysAgo = ago(7d);
SigninLogs
| where TimeGenerated >= daysAgo
```

Here, we specified a name of *daysAgo* and set it to be equal to the output of a *timespan* function, which returns a *datetime* value. We then terminate the *let* statement with a semicolon to denote that we are finished setting our *let* statement. Now we have a new variable called *daysAgo* that can be used anywhere in our query.

As mentioned earlier, you can wrap a whole query into a *let* statement as well. Here’s a slight modification on our earlier example:

```
let daysAgo = ago(7d);
let getSignins = SigninLogs
| where TimeGenerated >= daysAgo;
getSignins
```

In this case, we created a second *let* statement, where we wrapped our whole query into a new variable called *getSignins*. Just like before, we terminate the second *let* statement with a semicolon and call the variable on the final line, which will run the query. Notice that we were able to use *daysAgo* in the second *let* statement. This was because we specified it on the previous line; if we were to swap the *let* statements so that *getSignins* came first, we would get an error.

Let statements are very easy to use, and they make it much easier to organize your queries. They truly come in handy when you are organizing more complex queries that may be doing multiple joins.

Suggested learning resources

As you can probably tell, we only scratched the surface of KQL, but the goal here was simply to demystify the basics of the language. In order to keep building your expertise around KQL, we recommend taking an online course and reading through the formal documentation.

The following list of resources is by no means an exhaustive list. However, the information here will help you create your own custom Microsoft Sentinel notebooks.

https://aka.ms/KQLDocs [Official documentation for KQL]

https://aka.ms/KQLFromScratch [Pluralsight Course: KQL from Scratch]

https://aka.ms/KQLCheatSheet [KQL Cheat Sheet made by Marcus Bakker]

Appendix B

Microsoft Sentinel for managed security service providers

By Javier Soriano,
Senior Program Manager
Microsoft CxE Security

Managed security service providers (MSSPs) play a key role in monitoring and managing security devices and systems for their customers. As part of their services, MSSPs might include multiple tasks related to Microsoft Sentinel, like architecture and design, implementation, management, or security incident handling.

Automation is another important pillar that MSSPs must put special effort into. MSSPs must operate at great scales, and therefore streamlining things like customer onboarding is critical to their success.

In this appendix, we focus on how MSSPs can manage and operate multiple Microsoft Sentinel customers, with a focus on automation and efficiency.

Accessing the customer environment

Before the MSSP can start managing and operating a customer environment, they need to have access to it. As you have seen earlier in this book, Microsoft Sentinel is a resource type inside Azure. Therefore, it lives within an Azure Active Directory (AAD) tenant, which belongs to the customer. On the other side, MSSP identities live in a separate AAD tenant, so there must be a way to connect those two identity providers. There are actually two methods: Azure Lighthouse and Azure AD B2B.

Azure Lighthouse

Azure Lighthouse enables cross-tenant management, allowing for higher automation, scalability, and enhanced governance across resources and tenants. This is the preferred method to access your customer environment because it allows you to manage customer resources as if they were in your own Azure AD tenant.

Azure Lighthouse is based on delegations. Each delegation contains three components: Identities, Roles, and Scope.

- **Identities** These are the identities (normally from the MSSP tenant) that will have access to customer resources. You can specify users, groups, or service principals as the recipients of a delegation.
- **Roles** These are the permissions that the identities will have when accessing customer resources. The roles that can be used here are all Azure built-in roles, with three exceptions. Custom roles are currently not supported. Also, you cannot grant Azure AD roles.

> ***NOTE*** **See the differences between Azure and Azure AD roles at *http://aka.ms/azureadazurerbacroles*.**

- **Scope** This indicates where the delegation will apply; valid scopes are subscription and resource group.

In the context of Microsoft Sentinel, Azure Lighthouse can be used to manage the service across multiple customers. Figure B-1 shows a high-level view of the setup.

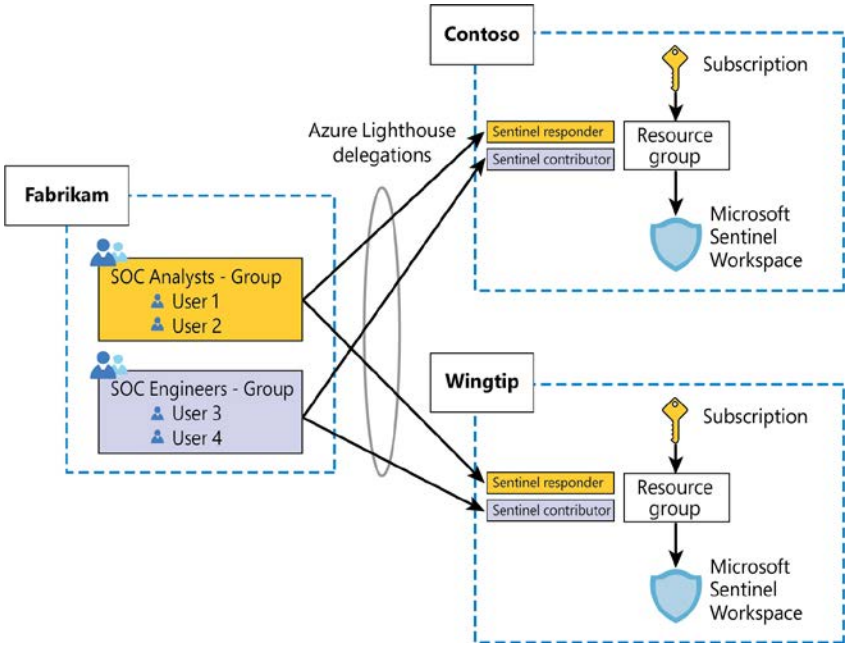


FIGURE B-1 Azure Lighthouse delegation from an MSSP to two customers

As you can see, in this case, the MSSP (Fabrikam) has two delegations for each customer—one for engineers with the Microsoft Sentinel Contributor role and one for analysts with the Microsoft Sentinel Responder role—and all have delegated access at the resource group level where Microsoft Sentinel is located. This will effectively provide them with access to the whole resource group with the permissions included in the granted role.

What can't you do through Lighthouse?

As we have explained before, the MSSP access to the customer's Microsoft Sentinel environment utilizes Azure Lighthouse. However, there are things that you won't be able to do with just Azure Lighthouse:

- You won't be able to onboard some connectors that require Security Admin or Global Admin permissions in the customer Azure AD tenant. Several Microsoft first-party connectors, like Office 365, Azure AD, or Microsoft 365 Defender, require one of these permissions to be enabled, and these roles can't be granted through Azure Lighthouse.
- You cannot assign incidents to a user in the customer's Azure AD tenant. Therefore, as you manage incidents in the customer workspaces, you will only be able to assign them to users in your own tenant.

Later in this appendix, we will review Azure AD B2B invites, which enable these scenarios.

Azure Lighthouse onboarding

As already mentioned, there are two options—an ARM template or an Azure Marketplace offer—the latter being preferred because it provides a very easy experience for customers.

> ***NOTE*** **There are some requirements before an MSSP can publish into the Azure Marketplace. The MSSP must have a silver or gold cloud platform competency level or be an Azure Expert MSP.**

Marketplace offers have an additional concept called a *plan*. A plan defines the service that you will provide to your customer. For example, you can have a marketplace offer to provide managed services for your customers, and within that offer, several plans with different flavors, including monitoring, backup and recovery, compliance, and fully-managed service.

In the context of Microsoft Sentinel, you could have an Azure Marketplace offer like the one shown in Figure B-2.

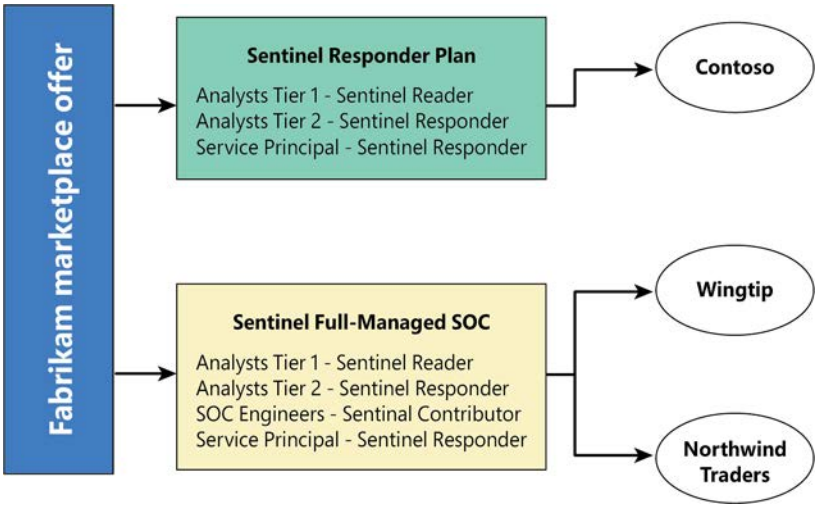


FIGURE B-2 Sample Azure Marketplace offer with two plans

As you can see, inside each plan, you define the groups of users from your tenant who will have access to the customer environment and the permissions that will apply. The plan also includes the scope (resource group or subscription), although we've omitted it in Figure B-2 for simplicity. Contoso, Wingtip, and Northwind Traders are customers that "purchase" specific plans from the Fabrikam offer.

You can make these plans public, so everyone in Azure can see them, or you can make them private if you only want a subset of customers to have access. This would allow you to create plans targeted just to specific audiences, such as a particular customer or a vertical.

Azure Lighthouse integration with Azure AD Privileged Identity Management (PIM)

Azure Lighthouse also can integrate with Azure AD Privileged Identity Management (PIM). This lets you grant delegated permissions to customer tenants on a just-in-time basis so that users only have those permissions for a set duration.

This can greatly reduce risks because it allows you to limit the number of permanent assignments of users to privileged roles. Because this feature relies on Azure AD PIM, it requires the MSSP Azure AD tenant to have licenses (such as Azure AD Premium P2) for Lighthouse. Also, Lighthouse can assign approvers who will be responsible for granting the requested permissions by the analysts.

Azure Lighthouse is a very important service for getting access to customer Azure resources, but it doesn't work for other workloads outside Azure, such as Office 365 or Microsoft 365 security services.

Azure Active Directory B2B

Azure Active Directory (Azure AD) business-to-business (B2B) collaboration is a feature within External Identities that lets you invite guest users to collaborate with your organization. MSSP users can be "invited" to the customer tenant to perform management activities in that tenant. MSSP users will appear as guest users in the customer tenant and can then be granted roles within. The main difference with Lighthouse is that the guest users can be granted any Azure (even custom ones) or Azure AD roles. (Remember, Azure Lighthouse can only grant Azure built-in roles.)

The ability to grant Azure AD roles opens new possibilities, such as managing Office 365 and Microsoft 365 services. However, you still need Azure Lighthouse because it provides two important capabilities not available with Azure B2B:

- **No cross-tenant management or visibility** As you are invited into a customer tenant, you must log in to the customer tenant in order to see its resources. This blocks your cross-tenant visibility because you cannot query multiple tenants simultaneously.
- **No ability to invite groups** Azure B2B is done on a user-by-user basis, meaning you cannot invite an entire group. This is challenging because you need to manage the lifecycle of each account in multiple places. (This limitation can be removed by using Azure Entitlement Management, which we review below.)

Taking these disadvantages into account, if you, as an MSSP, need to manage both Azure and Office 365/Microsoft 365 workloads, the best approach is to use a combination of Azure Lighthouse and Azure AD B2B invites. Figure B-3 depicts the combination.

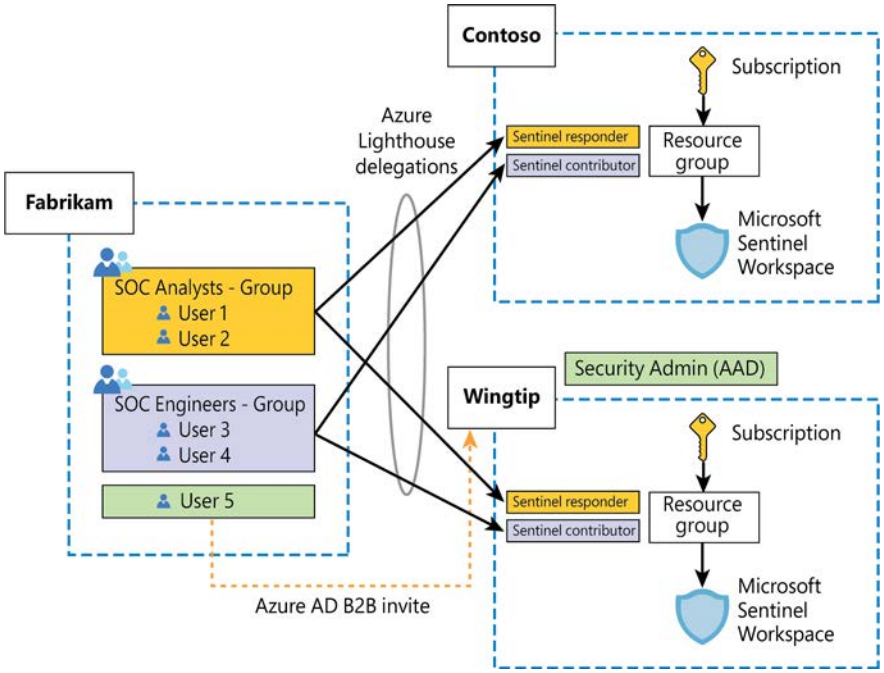


FIGURE B-3 Azure Lighthouse combined with Azure B2B

In Figure B-3, a new user from Fabrikam (MSSP) has been invited to Wingtip's Azure AD and is now a guest user in that AAD. Also, this user has been granted the Security Admin role. Notice that Security Admin is an Azure AD role, so it can be granted to guest users, but it cannot be granted to users who have delegated access via Azure Lighthouse. (Remember, Azure Lighthouse can only grant Azure roles.) Although not shown in Figure B-3, the same user can have Azure roles delegated through Azure Lighthouse and can be invited as a guest and be granted Azure AD roles like Security Admin or Global Admin.

Azure B2B invites provide a solution for the need to manage Office 365 or Microsoft 365 environments, but they can be difficult to automate.

Azure AD entitlement management

Azure Active Directory (Azure AD) entitlement management is an identity governance feature that enables organizations to manage identity and access lifecycle at scale by automating access request workflows, access assignments, reviews, and expiration. *This feature requires an Azure AD P2 license.*

This feature can also be used to manage access from external Azure AD organizations, so it's a perfect fit for the MSSP access needs when it comes to Microsoft 365 Defender workloads. Using this feature, MSSP users can be automatically invited into the customer tenant (after the appropriate approvals) to manage customer services. You can also assign which specific roles will be granted to those users; these roles should be specially crafted to manage Microsoft 365 Defender workloads.

> ***NOTE*** **For an in-depth explanation of setting up entitlement management for an MSSP to access customer environments, see *http://aka.ms/grantmsspaccess*.**

Remember, this is only needed to manage the Microsoft 365 Defender part of the customer environment. As explained above, the Azure Sentinel part will be managed through Azure Lighthouse.

Cross-workspace features

Now that we have reviewed how to access the customer environment, let's see how we can manage multiple Microsoft Sentinel customers in parallel.

At a high level, we will use the MSSP tenant as the single pane to look at multiple Microsoft Sentinel workspaces across Azure AD tenants. Figure B-4 shows this setup.

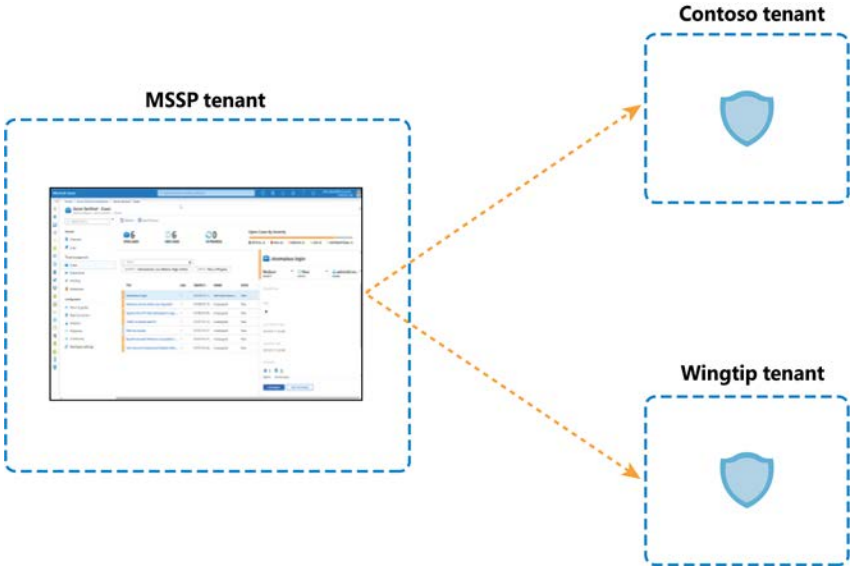


FIGURE B-4 Microsoft Sentinel multi-tenant management

Although this is the generic architecture, each Microsoft Sentinel feature has different characteristics and implementation details.

KQL Queries

Microsoft Sentinel supports querying multiple workspaces in a single query, allowing you to search and correlate data from multiple workspaces in a single query. This is done by leveraging the `workspace()` operator, which allows you to reference a table in a different workspace.

This can be very useful for SOC analysts when trying to analyze data from multiple customers in parallel. You can even create an alias, so your commands are easier to write and use. For example, if you want to look for failed AAD logins across two workspaces (A and B), you could create a function (alias) called `FailedLoginsAB` that contains the following code:

```
union isfuzzy=true workspace("workspaceA").SigninLogs, workspace("workspaceB").
SigninLogs
| where ResultType !in ("0", "50125", "50140", "70044", "70043")
```

Instead of having to write the KQL code above, an analyst can just use the `FailedLoginsAB` alias, which will return the aggregated results from both workspaces.

> ***IMPORTANT*** **The number of Log Analytics workspaces that you can include in a single query is limited to 100.**

Analytics rules

Scheduled Analytics rules also support the use of the `workspace` operator to reference tables in workspaces other than where the rule is being created. Following are important things to remember about cross-workspace rules:

- All workspaces referenced in the query must be onboarded into Microsoft Sentinel.
- A maximum of 20 workspaces can be used in a single analytic rule.
- Alerts and incidents will only be created in the workspace where the cross-workspace rule is created.
- There might be a performance impact if the same query contains workspaces in different regions.
- Investigation graph functionality for incidents and alerts coming from cross-workspace rules is limited. For example, expansion queries that can be executed on entities won't work properly.
- Cross-workspace rules are only possible with scheduled analytics rules. Other types of rules are not supported in this mode.

Because of these limitations, the use of cross-workspace analytics rules is only recommended in two scenarios:

- The intellectual property of the MSSP needs to be protected, and therefore the rule must be created in the MSSP tenant. (Artifacts created in the customer tenant will always be visible to the customer.)
- There is a need to correlate data coming from multiple customers (rarely needed in the MSSP scenario).

Figure B-5 shows how an MSSP can protect the intellectual property in an analytic rule by creating it in its tenant but pointing to the customer workspace.

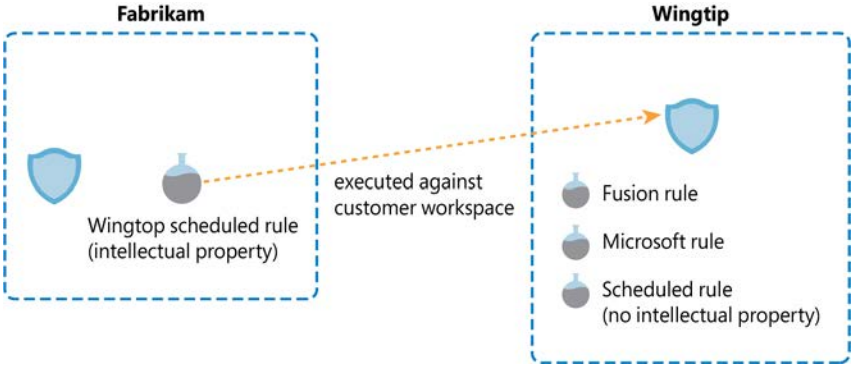


FIGURE B-5 Cross-tenant analytic rule

If cross-workspace rules are needed, the following best practices are recommended:

- Unless correlation is needed, don't mix customer workspaces into a single rule to avoid performance issues and poor manageability. Create one rule per customer.
- If you are managing many customers, consider whether you might hit the current limit of rules per workspace (512). If that's a possibility, create one workspace per customer in the MSSP tenant and place them in separate resource groups. Figure B-6 shows this scenario.

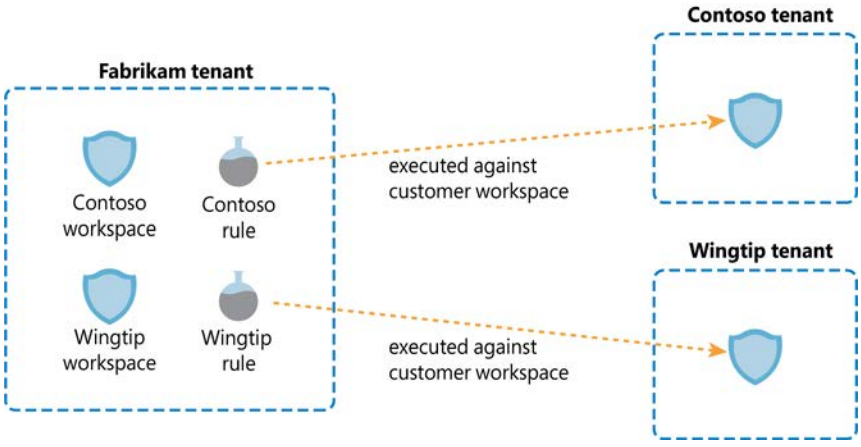


FIGURE B-6 Cross-tenant rule with separate workspace in MSSP tenant

Hunting

Threat hunting is a key function for MSSPs because it allows you to sweep through multiple customer environments in parallel to look for evidence of malicious activity. Microsoft Sentinel has several features related to threat hunting: hunting queries, Notebooks, and Watchlists.

Hunting queries

Like Analytics Rules, hunting queries can also use cross-workspace queries in KQL by utilizing the `workspace()` operator. As with analytics rules, all workspaces that are part of the query need to be onboarded into Microsoft Sentinel.

Although not strictly related to MSSPs, hunting queries also permit the use of the `adx()` operator, which can be used to reference data sitting on an Azure Data Explorer (ADX) cluster. In some scenarios, this can be useful for correlating with other data sitting in ADX.

Notebooks

In Chapter 6, you saw many of the great things you can do with Notebooks. In the context of MSSPs, Notebooks can be a very versatile investigation and threat hunting tool. For example, you can have a Notebook that looks for evidence of Log4j exploitation that looks at all your customer workspaces in parallel without looking at each one of them individually.

The first thing you should do to use Notebooks in a multi-tenant setup is to add all your workspaces to your `msticpyconfig.yaml` file. This allows you to reference whichever workspace you need, depending on the query. Following is an example of what a `msticpyconfig.yaml` file with three workspaces would look like:

```
AzureSentinel:
  Workspaces:
    Default:
      ResourceGroup: mssp_sentinel
      SubscriptionId: xxx-xxx-xxx-xxx
      TenantId: xxx-xxx-xxx-xxx
      WorkspaceId: xxx-xxx-xxx-xxx
    customerA:
      ResourceGroup: customerA
      SubscriptionId: yyy-yyy-yyy-yyy
      TenantId: yyy-yyy-yyy-yyy
      WorkspaceId: yyy-yyy-yyy-yyy
    customerB:
      ResourceGroup: customerB
      SubscriptionId: zzz-zzz-zzz-zzz
      TenantId: zzz-zzz-zzz-zzz
      WorkspaceId: zzz-zzz-zzz-zzz
```

Once you've done that, there are several ways in which you can utilize these workspaces within your Notebooks:

- Use a cross-workspace query (utilizing the `workspace()` operator) that will result in a table that will include records from all the specified workspaces. You can then split it into multiple data frames. (This option doesn't use the definitions in the `msticpyconfig.yaml` file, but it is a good practice to add them there.)
- Create multiple connections and query each, one by one. You can have multiple queries in one `%kql` cell. Separate each query with an empty line and assign the result of each query to a different Python variable. You can then aggregate results using the `append()` function.
- Write Python code that iterates over the different workspaces in your `msticpyconfig.yaml` file and use `%kql` for each.

All the options above are valid, and choosing the right one will greatly depend on the specific needs of the MSSP and the type of Notebook you need to create.

Watchlists

Watchlists are another important tool for MSSPs. Besides delivering all the normal functions, they can be used to provide additional context to the MSSP. For example, if the customer is using cross-tenant content like a Workbook, it might be difficult for a customer to identify from which customer the data is coming because the data doesn't contain the customer's name per se. By using a Watchlist, we can build a mapping table that correlates the workspace ID (stored in the `TenantID` field present in every table) with a friendly customer name.

It's important to remember that Watchlists only work in the context of the workspace where they are defined, and we can't reference a Watchlist in a remote workspace in a KQL query—not even by using the `workspace()` operator.

Incident management

Microsoft Sentinel allows you to view incidents coming from multiple workspaces in a single consolidated view. For better navigation, this cross-workspace incident view includes new columns indicating the workspace and the directory (Azure AD tenant) the incident is coming from. This view is extremely useful for MSSPs because it allows an analyst to oversee multiple customers from a single pane.

At the time this book was written, the cross-workspace incident view has a limit of 100 workspaces that can be monitored in parallel. Figure B-7 shows the Incidents view with incidents coming from multiple workspaces and tenants.

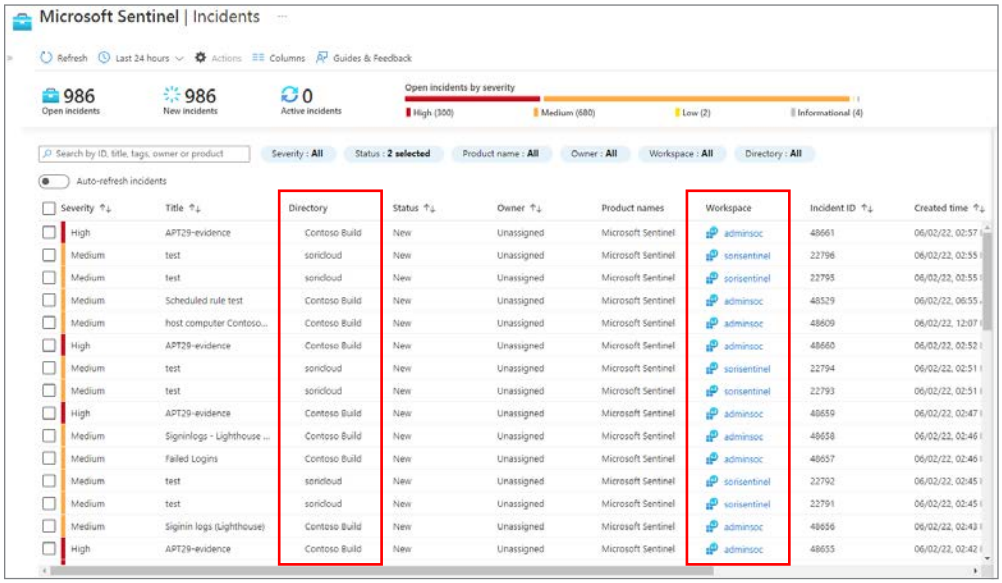


FIGURE B-7 Multi-tenant Incidents view

Once the user drills down into an incident, the user will be redirected to the appropriate workspace where that incident was created.

As we mentioned previously in this appendix, MSSP users managing incidents will only be able to assign them to other users in the MSSP Azure AD tenant. If there's a requirement to assign to users in the customer tenant, Azure B2B must be used in addition to Azure Lighthouse.

Automation/SOAR

As explained in other chapters in this book, there are two main SOAR components in Microsoft Sentinel: automation rules and Playbooks. In general, if there is no need to protect MSSP intellectual property, we recommend that you create both artifact types in the context of the customer's workspace. This simplifies the management of credentials used inside Playbooks and allows for the use of managed identities where possible.

> ***NOTE*** For implementation details of this option, see *http://aka.ms/automationrulesmssp*.

However, if protection of the MSSP's intellectual property is a requirement, Playbooks can be saved in the MSSP tenant, and the automation rules in the customer workspace can use them. This is the same if the Playbook is referenced directly from an analytics rule (see Figure B-8).

FIGURE B-8 The automation rule referencing the Playbook in another tenant

There are a couple of additional considerations for this model:

- The cost of the Logic App execution is charged to the MSSP tenant.
- If the Playbook needs to perform some sort of remediation activity in the customer environment, it will need the appropriate credentials. For example, if the Playbook needs to block Azure AD accounts, we will need to provision a service principal in the customer tenant that has the relevant permissions and then use those credentials in the Logic App.

Workbooks

Workbooks can also be modified to query data from multiple workspaces. This can be very useful if you need to see an aggregated view of data coming from multiple customers (see Figure B-9).

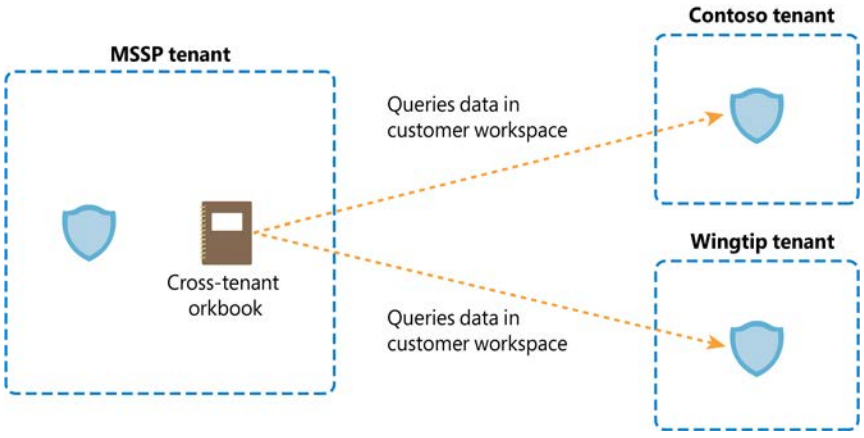


FIGURE B-9 Workbooks can reference data in customer workspaces

> ***TIP*** **One of the ways to implement this is by adding a workspace selector in your existing Workbook. See *http://aka.ms/crossworkspaceWorkbooks*.**

Like with other artifact types, hosting your Workbooks in the MSSP tenant can also be used as a way to protect intellectual property inside that Workbook. However, there are situations where the customer also needs to see and interact with that Workbook.

> ***TIP*** **For those cases, we recommend using PowerBI (see *http://aka.ms/loganalyticspowerbi*).**

Additional benefits of PowerBI include the following:

- **Easier to share** You can just send a link to the PowerBI dashboard, and the user will be able to see the report. There is no need to have Azure access permissions.
- **Scheduling** You can configure PowerBI to send an email on a given schedule that will contain a snapshot of the dashboard.

Security content management

Automation and DevOps practices are crucial components for a successful managed security practice. These are some of the key benefits:

- Reduction of human error
- Much faster deployment and configurations
- Improved change management because changes are tracked in source code control
- Enhanced security as consistency is guaranteed
- Time savings to allow employees to focus on adding value to our customers

When multiple customers are managed in parallel, there will always be content that is deployed to many or even all your customers. If a modification is needed for that content, we must have a way to make that modification only once and then have an automatic process that verifies if the change is valid and that updates any copies of that same content across all our customers. This is achieved via continuous integration and continuous deployment, or CI/CD.

How to adopt CI/CD?

There are several steps you should follow to adopt and implement CI/CD in your content management process:

- Turn your security content (detections, dashboards, Playbooks, queries, and so on) into code that can be interpreted by a machine. The most common formats are YAML and JSON.
- Host your code in a source code repository, such as Git, GitHub, Azure DevOps Repos, or BitBucket.
- Build continuous integration (content validation) and continuous deployment (content implementation) pipelines.
- Choose and configure a DevOps tool that orchestrates it all (such as Azure DevOps, or GitHub).

Luckily for us, Azure is very much built with DevOps in mind and already offers a great way to codify and automatically validate and deploy our content: Azure Resource Manager (ARM) templates.

> ***NOTE*** **You can see all the features already built into the ARM templates at *http://aka.ms/armtemplatefeatures*.**

As with any other Azure resource, Microsoft Sentinel and its associated content can benefit from using ARM templates, so we already have a great way to codify our security content that comes with tools to check the validity of the content and deploy it to Azure environment.

NOTE **You can see the reference on how to codify each Microsoft Sentinel component as an ARM template at *http://aka.ms/sentinelarmreference*.**

TIP **Building your own ARM templates can be a daunting task, though, especially if you're not used to JSON. To ease this process, we provide ways for users to easily generate ARM templates from some of their existing artifacts. For example, for analytics rules, you can export them to ARM templates from the Microsoft Sentinel portal. See *http://aka.ms/exportanalyticsrules*. There's also a script that can do this for more content types at *http://aka.ms/exportsentinelcontent*.**

Microsoft Sentinel repositories

Once you have your security content in ARM template format and hosted in a source code repository, you still need to configure a DevOps tool to create your CI/CD pipelines. To make this process even easier, Microsoft Sentinel offers Repositories, a feature that seamlessly integrates with GitHub and Azure DevOps, automating the following steps:

- Automatically creates a service principal (SPN) in Azure AD
- Grants that SPN permissions to deploy content to the Microsoft Sentinel workspace
- Creates a connection from either GitHub or Azure DevOps to the Azure environment
- Places a PowerShell script in the source code repository that can deploy the ARM content in the repository to Microsoft Sentinel
- Creates a CD pipeline that uses the PowerShell script

All the steps above are completely transparent to the user, who just needs to authenticate to the DevOps platform for the setup to be successful.

The repositories feature also allows you to select which content types should be deployed with this feature. Your choices are Analytics Rules, Hunting Queries, Workbooks, Playbooks, Parsers, and Automation Rules.

In the context of MSSPs, repositories can be a very useful tool to automate content deployment to customer workspaces. For example, you can set up a code repository in Azure DevOps where you store your security content. Then you create connections from each of the customer's workspaces to that repository. Whenever you make a change to your code, the updates will be automatically deployed to all your customers in parallel. If there's specific content that you only want to deploy to a subset of your customers, you can organize your content in branches or folders, as shown in Figure B-10.

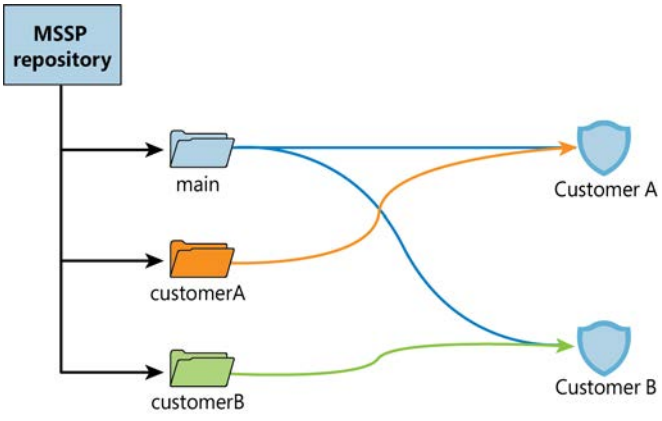


FIGURE B-10 MSSP Repositories architecture

As you have seen in this appendix, Microsoft Sentinel is fully adapted to work in an environment where there are multiple Azure AD tenants involved, as is the case for MSSPs. Additionally, Sentinel uses a cloud-native and API-driven approach, making it ideal for automating at scale so MSSPs can speed up their operations and focus on serving their customers.

Index

SYMBOLS

A

B

C

D

E

F

G

H

I

J

K

L

M

N

O

P

Q

R

S

T

U

V

W

Y