03 January 2025

Exploring Azure Sentinel: A Hands-On Lab for Cybersecurity

Description

This project provides a comprehensive walkthrough of configuring and utilizing Azure Sentinel for security monitoring, threat detection, and incident response. It covers the deployment of Azure resources, including virtual machines, the setup of data connectors, and the integration of Windows Security Event logs. By leveraging the MITRE ATT&CK framework, the project demonstrates how to identify and investigate potential threats using advanced query techniques within Azure Sentinel. This hands-on lab not only highlights the practical applications of SIEM tools but also emphasizes best practices for enhancing security posture in a modern cloud environment.

Irfan Ahmed

Introduction

The increasing sophistication of cyber threats has highlighted the critical need for robust Security Information and Event Management (SIEM) systems to protect organizational assets. Azure Sentinel, a cutting-edge cloud-native SIEM and Security Orchestration, Automation, and Response (SOAR) solution by Microsoft, provides powerful tools for detecting, investigating, and responding to security threats in real-time.

This project aims to demonstrate my cybersecurity expertise by exploring and utilizing Azure Sentinel in a hands-on lab environment. Through this project, I will deploy and configure essential Azure resources, implement security best practices, and analyse security event data to detect and respond to potential threats. The lab will also showcase my ability to leverage advanced analytics, custom rules, and the MITRE ATT&CK framework to enhance threat detection and mitigation.

This documentation captures the step-by-step process, key findings, and practical insights gained during the project, serving as a testament to my technical proficiency and problem-solving skills in a real-world cybersecurity context.

Project Goals

The primary objectives of this project are:

1. **Showcase Skills in Cybersecurity**:
    a. Demonstrate hands-on expertise in configuring and managing a cloud-native SIEM solution.
    b. Highlight proficiency in log analysis, custom rule creation, and real-time threat monitoring.
2. **Deploy and Configure Azure Resources**:
    a. Set up a secure and scalable environment using Azure services, including Log Analytics Workspace, Virtual Machines, and Azure Sentinel.
3. **Implement Security Best Practices**:
    a. Apply best practices for securing network and endpoint resources to mitigate vulnerabilities.
4. **Explore Data Integration and Analysis**:
    a. Integrate data sources using Azure Sentinel connectors and analyze Windows Security Event logs to gain actionable insights.
5. **Leverage Advanced Threat Detection**:
    a. Use KQL (Kusto Query Language) for custom queries and write custom analytics rules to detect security incidents effectively.
6. **Utilize MITRE ATT&CK Framework**:
    a. Map detected threats to adversary tactics, techniques, and procedures to align detection efforts with a recognized cybersecurity standard.
7. **Create a Comprehensive Project Portfolio**:

a. Document the project in written and video formats to serve as a showcase of my technical capabilities for prospective employers.

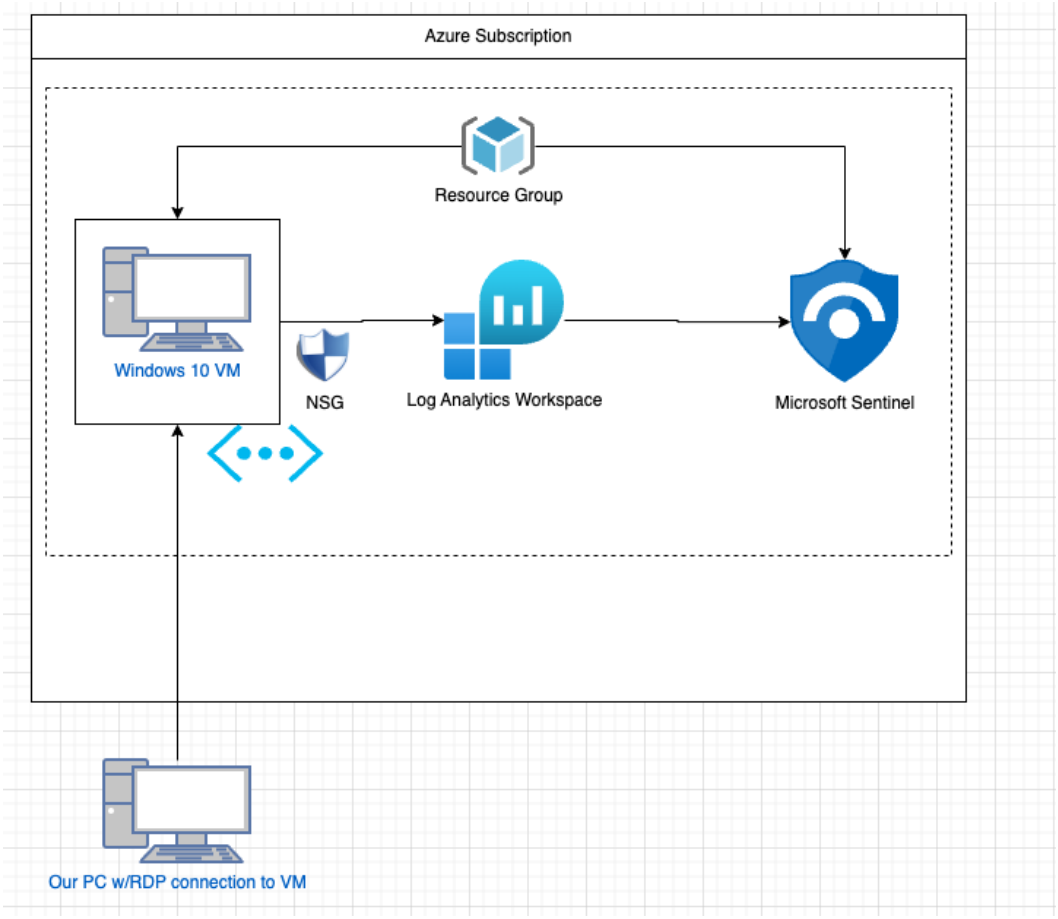
By achieving these goals, this project will demonstrate my ability to work with cutting-edge cybersecurity tools and platforms, reinforcing my readiness to tackle real-world challenges in a professional setting.

Key Learning Objectives

This hands-on lab is designed to provide a comprehensive understanding of Azure Sentinel and its role in enhancing an organization's security posture. By completing this project, you will:

1. **Understand the Core Features and Capabilities of Azure Sentinel**
    a. Gain insights into the primary functionalities and advantages of using Azure Sentinel as a SIEM and SOAR solution.
2. **Learn About Azure Sentinel's Integration with Other Azure Services**
    a. Explore how Azure Sentinel seamlessly integrates with Azure-native tools and third-party solutions to create a unified security framework.
3. **Navigate the Azure Sentinel Console**
    a. Develop familiarity with the Azure Sentinel interface.
    b. Access and utilize tools required to manage and analyze security events.
4. **Master Data Collection and Integration**
    a. Discover the process of connecting various data sources to Azure Sentinel.
    b. Enable comprehensive data visibility across your network environment.
5. **Engage in Real-Time Threat Monitoring**
    a. Perform real-time threat monitoring and incident detection using advanced analytics.
    b. Leverage correlation capabilities to identify patterns and anomalies.
6. **Leverage Threat Intelligence and Automate Responses**
    a. Understand how to incorporate threat intelligence feeds into Azure Sentinel.
    b. Create and manage automated responses to security incidents, reducing reaction time and enhancing mitigation efforts.
7. **Gain Hands-On Experience**
    a. Conduct practical exercises and simulations that replicate real-world cybersecurity scenarios.
    b. Strengthen your problem-solving and analytical skills through hands-on application.

Project Topology



This network topology illustrates the setup of the lab environment within an Azure subscription, designed to deploy and utilise Microsoft Sentinel for security monitoring and incident response.

1. **Azure Subscription:**
    - The entire environment resides within a single Azure subscription, providing the cloud-based infrastructure required for resource provisioning and management.

2. **Resource Group:**
    - All resources, including the Windows 10 Virtual Machine (VM), Network Security Group (NSG), Log Analytics Workspace, and Microsoft Sentinel, are organized within a single **Resource Group**. This grouping allows for streamlined management and monitoring of related resources.

3. **Windows 10 Virtual Machine (VM):**
    - A Windows 10 VM acts as the primary endpoint for this lab. It is configured to send system and security logs to the **Log Analytics Workspace** for centralized analysis.

- The VM is accessed remotely via **Remote Desktop Protocol (RDP)** from the user's local PC, allowing hands-on interaction for tasks such as configuring security policies and simulating security events.

4. **Network Security Group (NSG)**:
    - The NSG provides a layer of security for the VM by controlling inbound and outbound network traffic. This ensures that only authorized connections, such as RDP, are permitted.

5. **Log Analytics Workspace**:
    - The **Log Analytics Workspace** acts as the central repository for logs and telemetry data collected from the VM. It facilitates detailed analysis, querying, and visualization of security-related data.

6. **Microsoft Sentinel**:
    - Microsoft Sentinel is deployed on top of the **Log Analytics Workspace** to provide advanced SIEM and SOAR capabilities. It utilizes the collected data to detect threats, analyse incidents, and automate responses to potential security events.

7. **Local PC Connection**:
    - The user's local PC establishes a remote desktop connection to the Windows 10 VM, allowing the user to configure resources, simulate security events, and analyse data using tools such as KQL queries and custom analytic rules.

Data Flow:

1. Logs and telemetry data generated on the Windows 10 VM are sent to the Log Analytics Workspace.
2. Microsoft Sentinel analyses this data in real-time to detect and respond to threats
3. The user manages and monitors the environment via RDP from their local PC.

Setup and Implementation

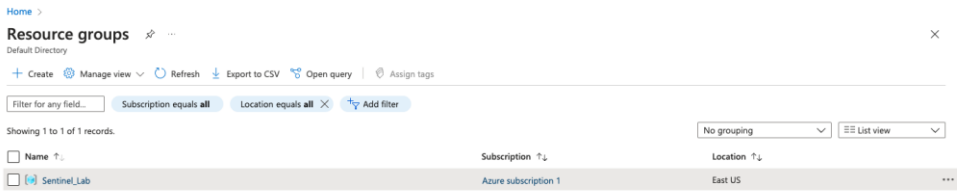
Step 1: Create Azure Account

In order to begin this project, we need an Azure account and this can be done relatively easy. Using the link below we can set up our account and this will automatically associate an azure subscription for us.

https://azure.microsoft.com/en-us/free/

Step 2: Create a Resource Group

When working with Azure, we can use Resource groups to group all our resources. Resource Groups are logical containers for our resources and within this group will include our Windows 10 VM, Log Analytics Workspace, and Azure Sentinel Resource.

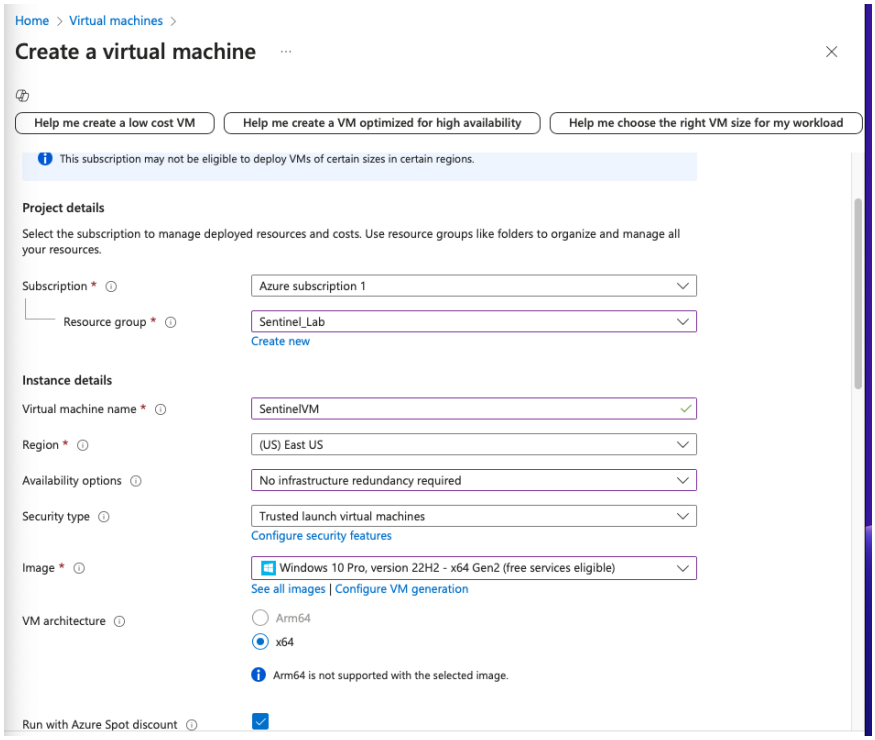


Here is the screenshot of our resource group. For this lab, its labelled 'Sentinel_Lab' and the settings are as shown. We can use the basic settings and do not need any extra.

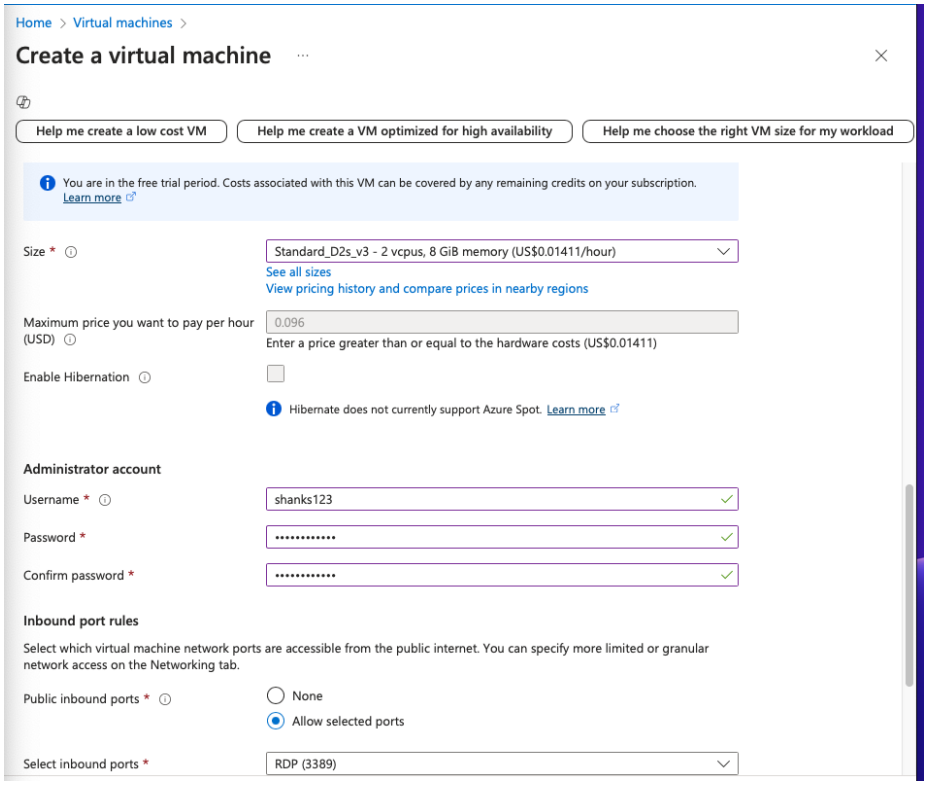
Step 3: Deploy Virtual Machine

The data we will be collecting is from a Windows Virtual machine so we will need to deploy one of these.

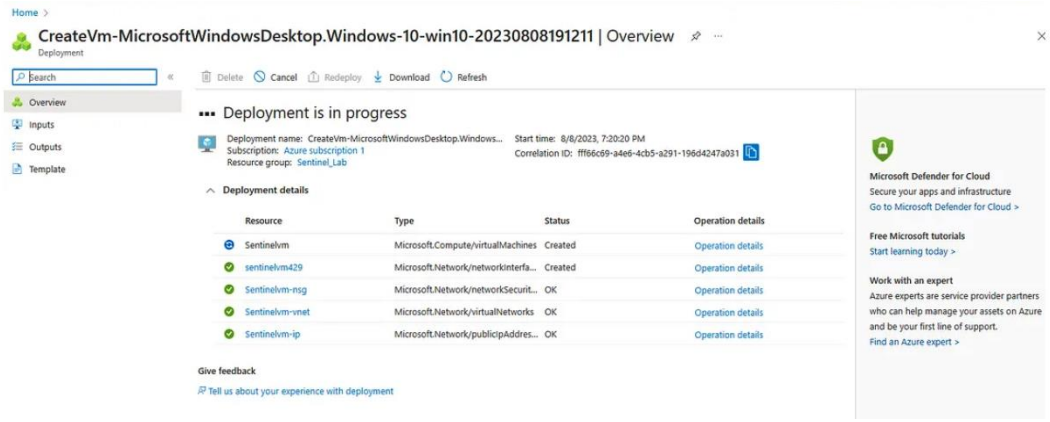
1. In the Resource Group tab, we can select Virtual machine tab to create our VM. First we need to set up the details of our project and the specific instance we need.



2. Next, we must set up the size of our VM as well as the administrator details



3. Once we press create, our VM is then ready for us to use and we get this result.

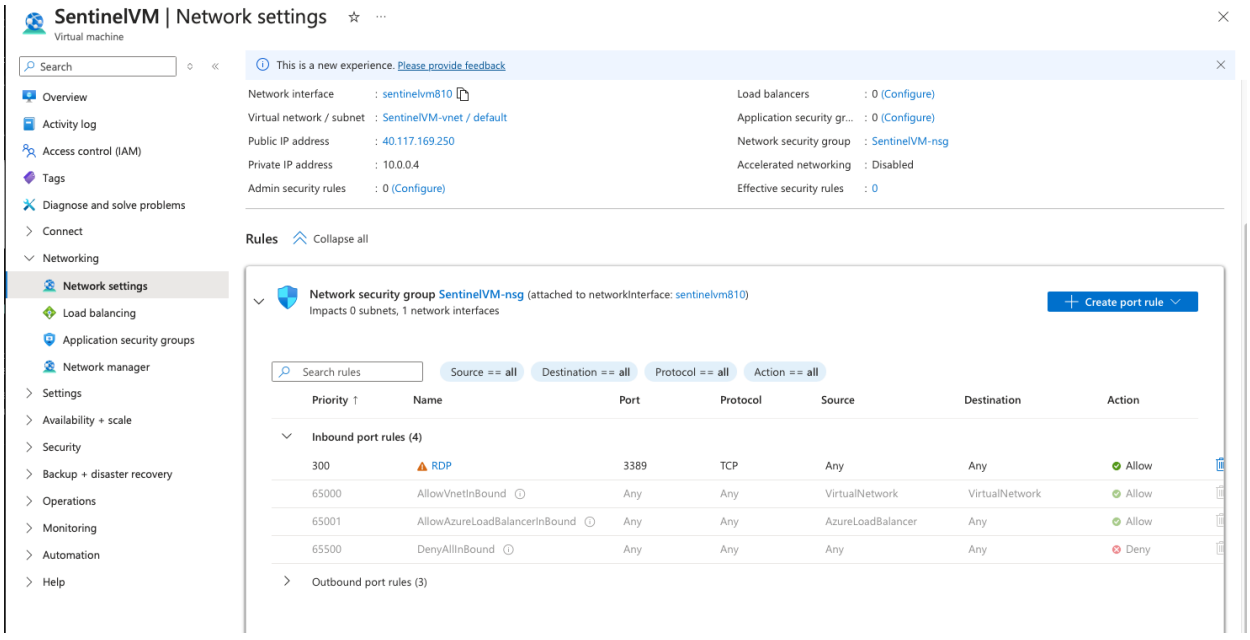


Network and Virtual Machine Security

When deploying a Virtual Machine in Azure, the VM is assigned to a virtual Network (Vnet). The VM is assigned an IP address on that network as well as a network interface.

Another feature Azure has is the ability to implement a basic firewall. This can be done using the Network Security Groups (NSG). An NSG can be used to filter network traffic to and from Azure resources and since it is a firewall it filters based on rules that dictate source and destination ports and network protocols that are allowed or deny.

- In the VM settings, we can head to Networking to see our NSG and the default rules.



Windows Defender

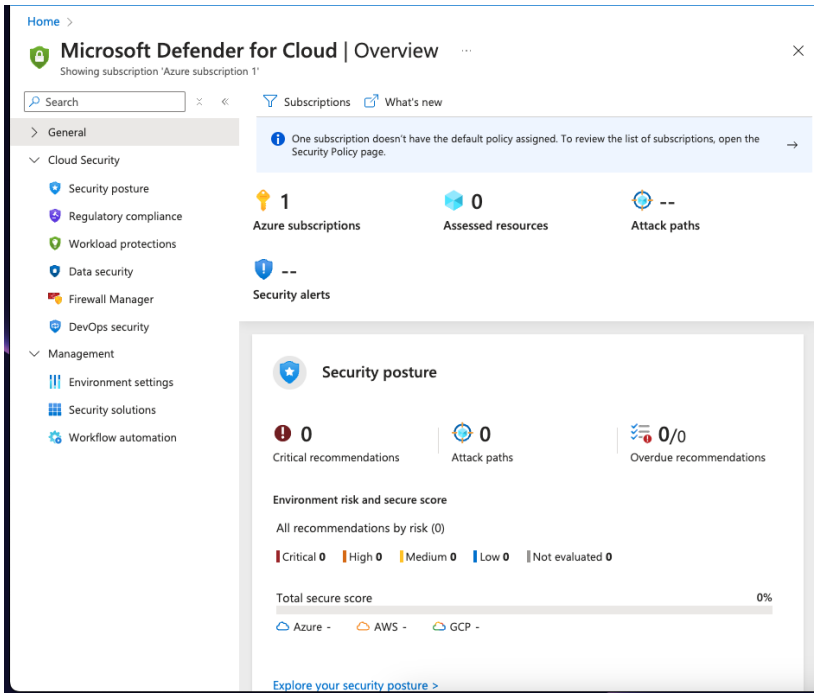
When we were setting up our VM, we enabled a feature where inbound RDP traffic is allowed from any source to any destination. If you look in the above screenshot you will be able to see this. RDP is necessary for us to access our VM however with the current setting it can allow anyone who obtains our public IP to potentially connect to our VM as it is public facing.

This presents a security risk as our VM is vulnerable to a brute force or password spray attack. In order to reduce our attack surface, we need to enable a security feature called 'Just in Time Access'.

- Just in Time Access: this works by only providing access to our VM when necessary, via time-based restrictions as well as implements the principle of lease privilege by giving the option to restrict access to certain IP's as well as RBAC roles.
- If an individual wants access to the VM they will need to request and based on their IP and assigned role, they will be either granted or denied access.
- As this is our azure account, we are Global Administrator so upon request we will be granted access to the VM.

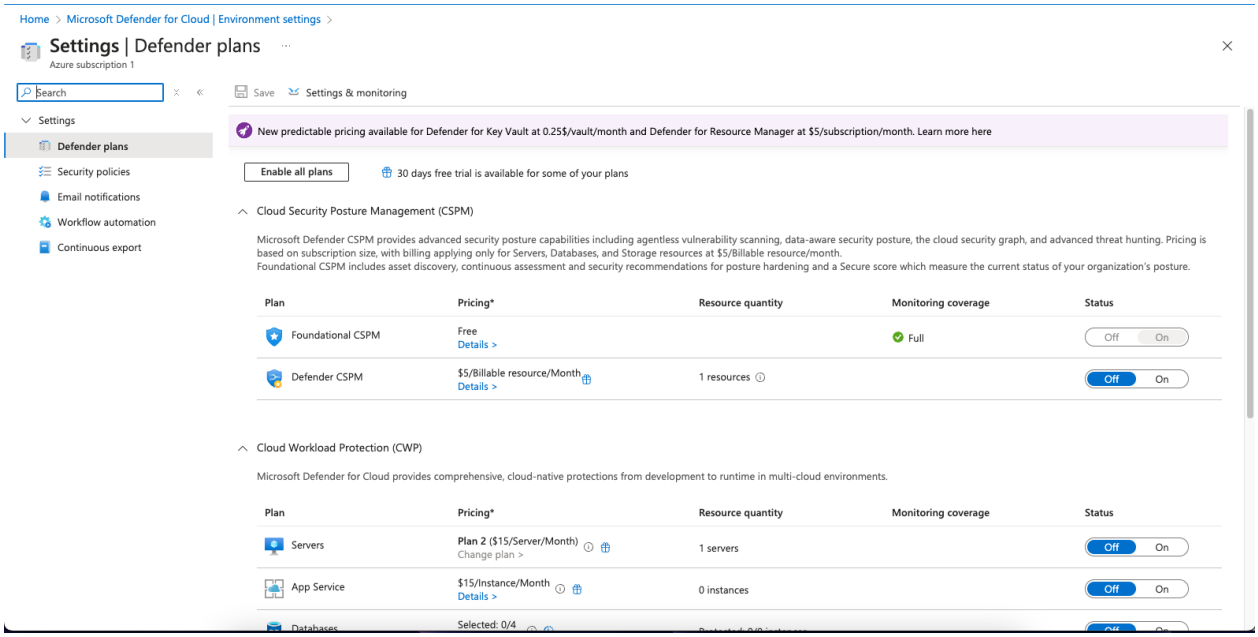
Setup

1. In order to setup Defender, we need to head over to the Defender tab in Azure.



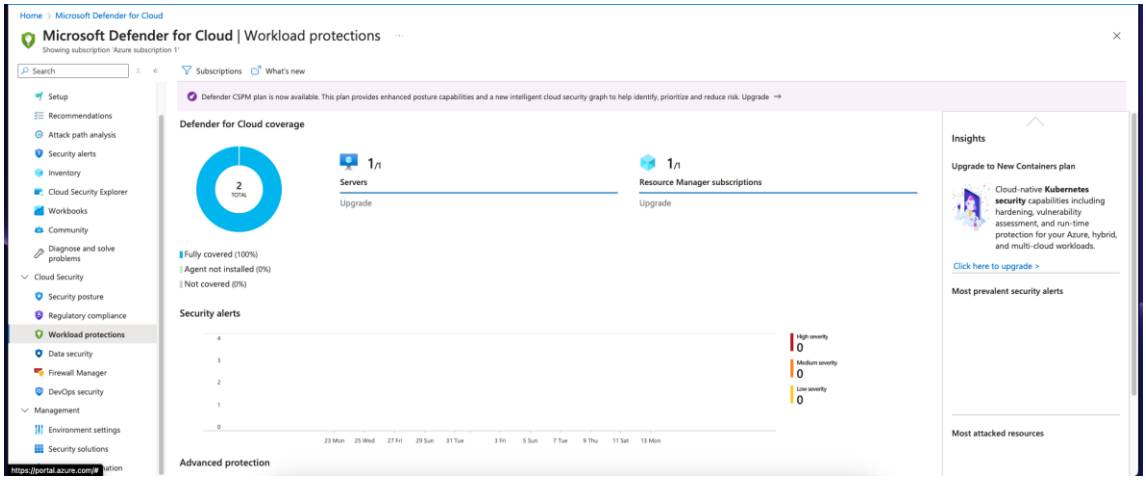
From here we need to head over to 'Environment Settings'.

2. From here, we need to select our Azure subscription and we will be given this screen below:



From here, we want to select the 'Enable All Microsoft Defender for Cloud Plans' as 'Enhanced security' is off by default.

3. Once this is done, we head over to 'Workload Protections' which will give us this screen below:



In the Advanced protection section, we will select 'Just in Time VM access' and also select 'Enable JIT on VM'. We need to make sure that we select our VM that we previously created.

We can also do it via the settings in our VM by going to 'Connect' and configuring 'Just in Time Access'.

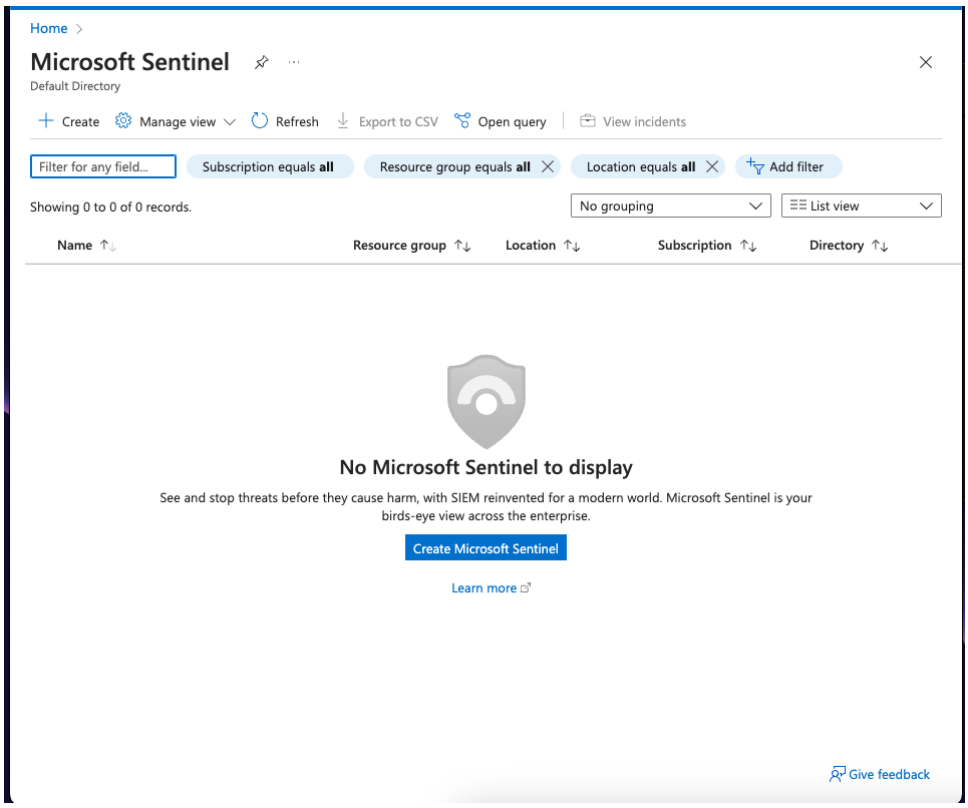
Log Analytics and Microsoft Sentinel

When working with log data in Azure, we need somewhere to store that data and Log Analytics Workspace is used to collect and store log data from Azure Resources.

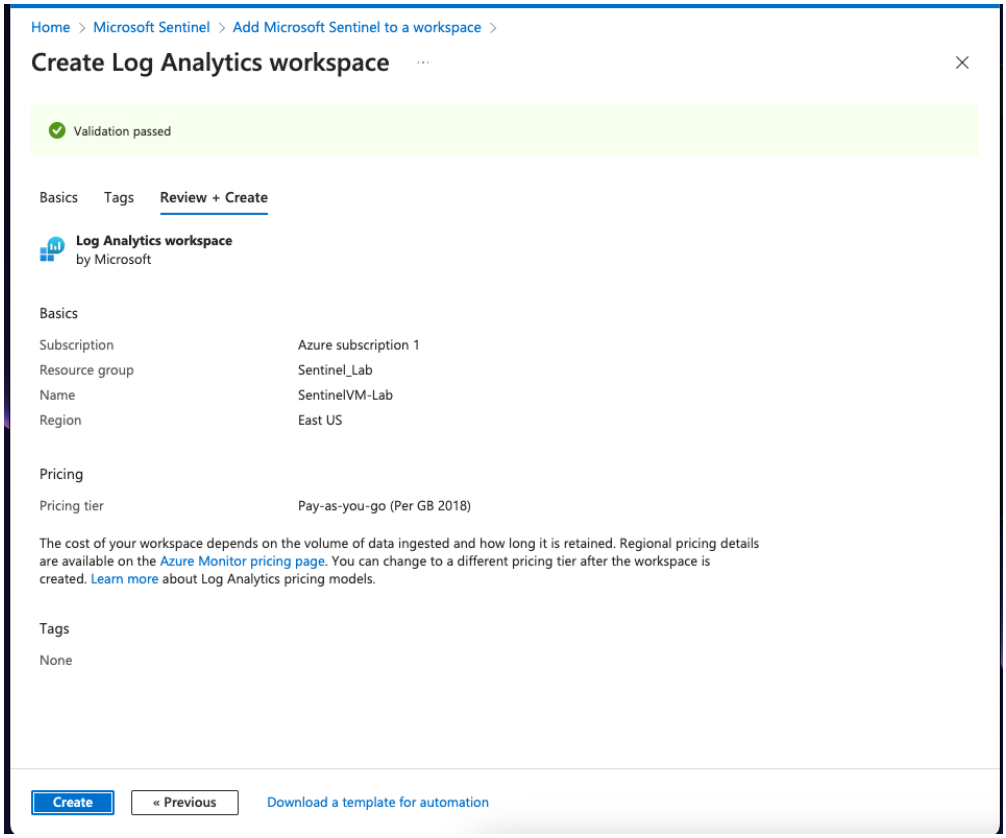
Setup

When working with log data in azure we need somewhere to store that data. Log Analytics Workspace is used to collect and store log data from Azure Resources.

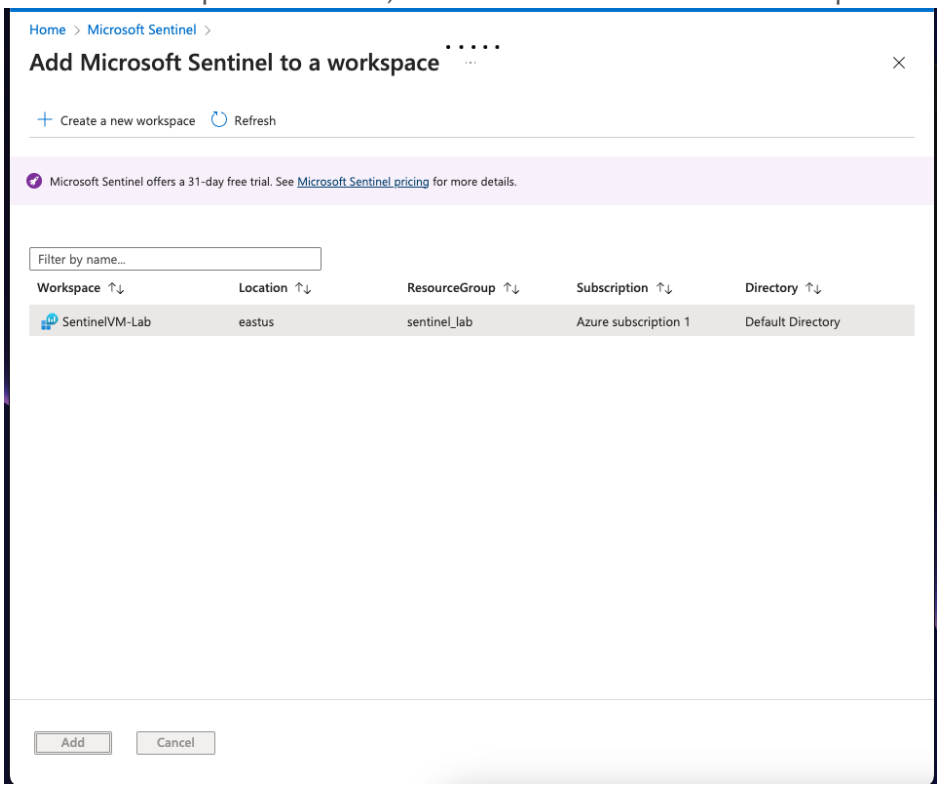
1. In order to set up a log analytic workspace we need to into the 'Microsoft Sentinel' tab in Azure portal where we can create the workspace.



2. We need to make sure that we use the same resource group we used for the VM, when creating the workspace.

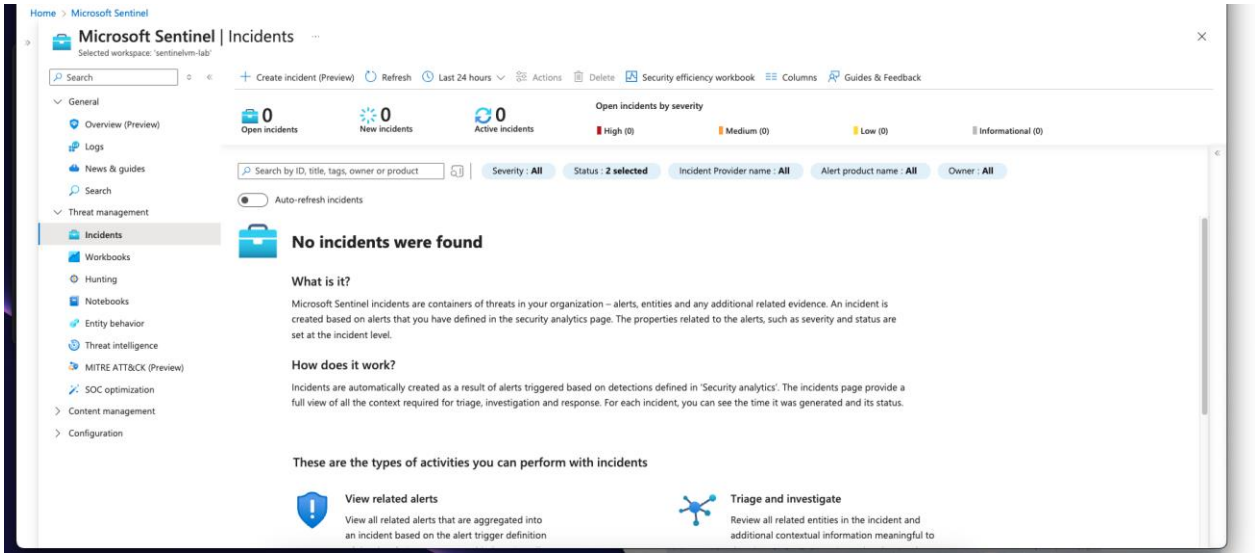


3. Once the workspace is created, we can add it Sentinel to the workspace



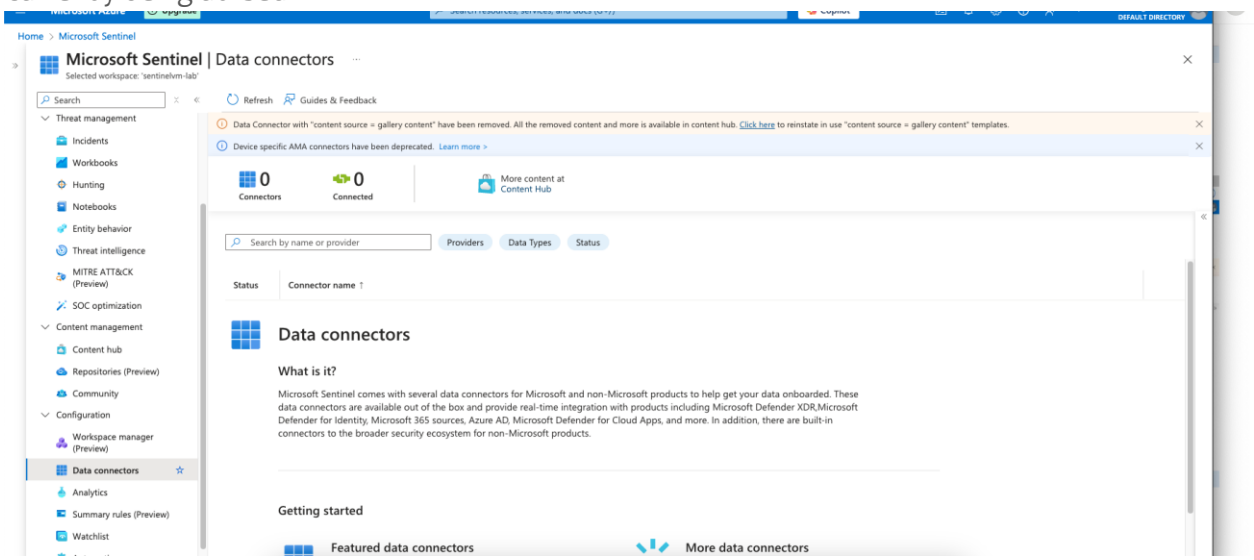
Injecting data into Sentinel

Now that we have deployed Sentinel, we need to simulate some incidents in order to test out the feature. If we go into the incidents tab in sentinel, we can see that we have no incidents currently as there is no data being fed into Sentinel.

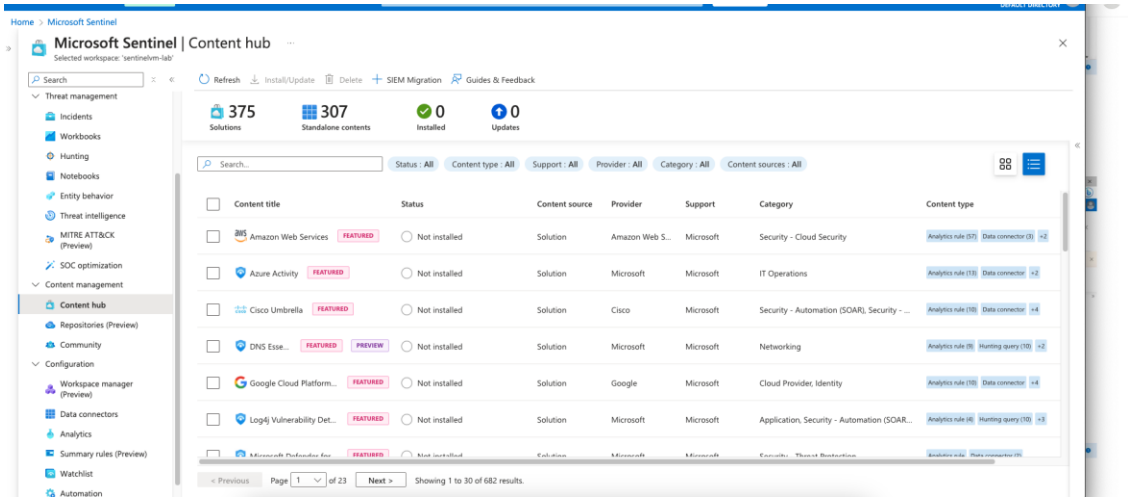


We need to utilise data connectors and create data collection rules to bring in data from our Windows 10 VM to Sentinel.

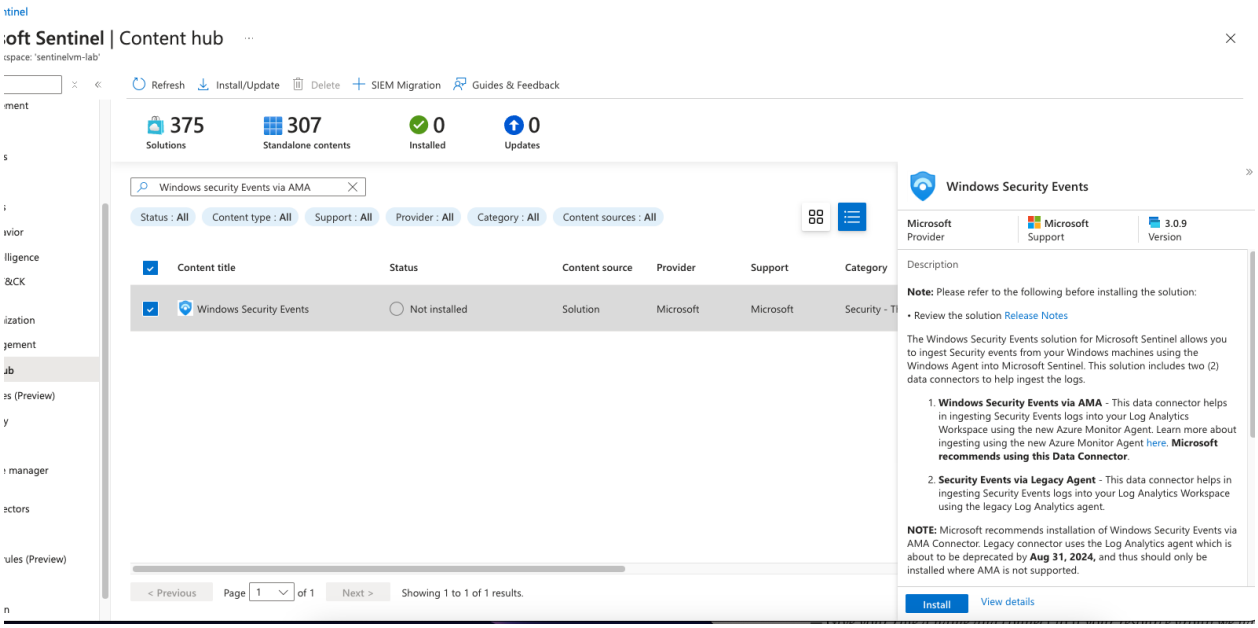
1. First, we need to go into Configuration and here we can see there is no data connectors currently being utilised.



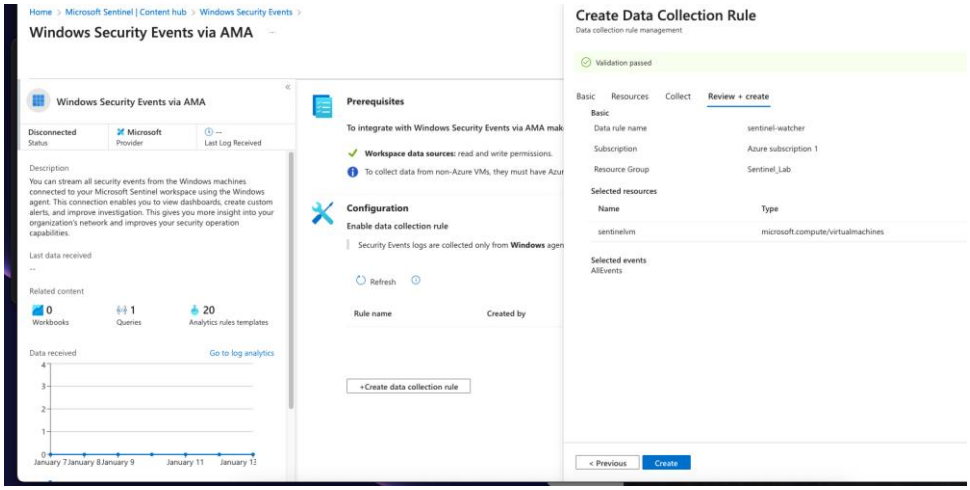
2. This is where we go into Content Hub as this where we can select the type of data collectors, we want to bring into our SIEM.



3. The data collector we are looking for is Windows security Events via AMA and we can search for that option using the search bar.



4. Next, we need to install the data connector and create a data collection rule.

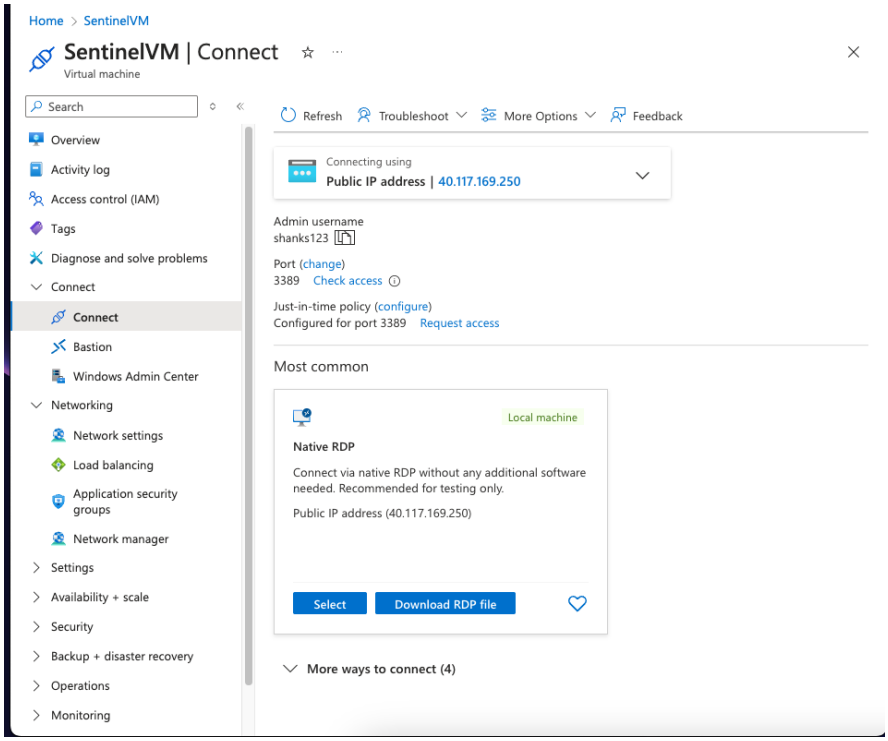


Remote Accessing and Generating Security Events

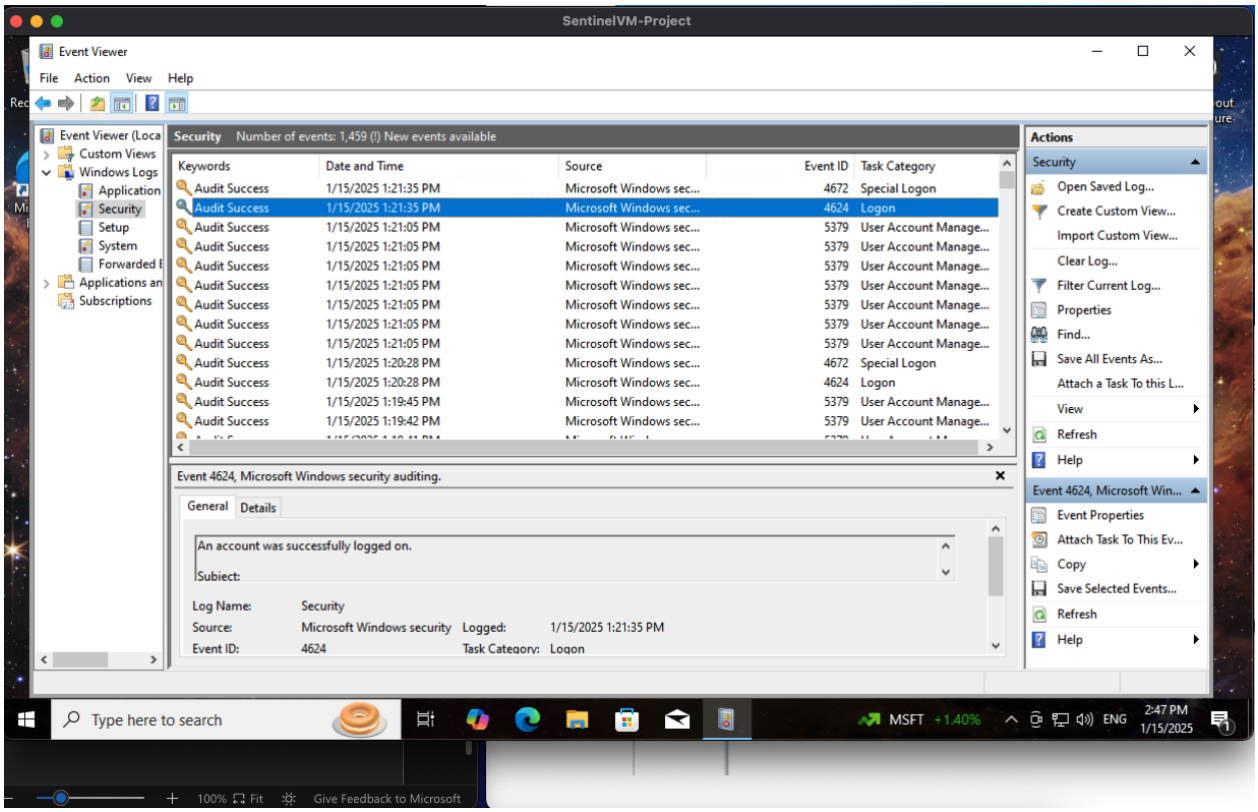
Now that our VM is connected to sentinel and our Log Analytics Workspace, we need to ingest some data into our Logs. To do this, we simply need to perform some actions on the Windows 10 VM that will generate security alerts.

Windows keeps a record of several types of security events. These events cover several potential scenarios such as privilege use, logon events, processes, policy changes and more. Let start by observing some Windows Security events on our VM.

1. First, we need to log into our VM, to do this we will go to the connect tab for our VM to acquire the IP address we will use and then we can RDP into the VM.

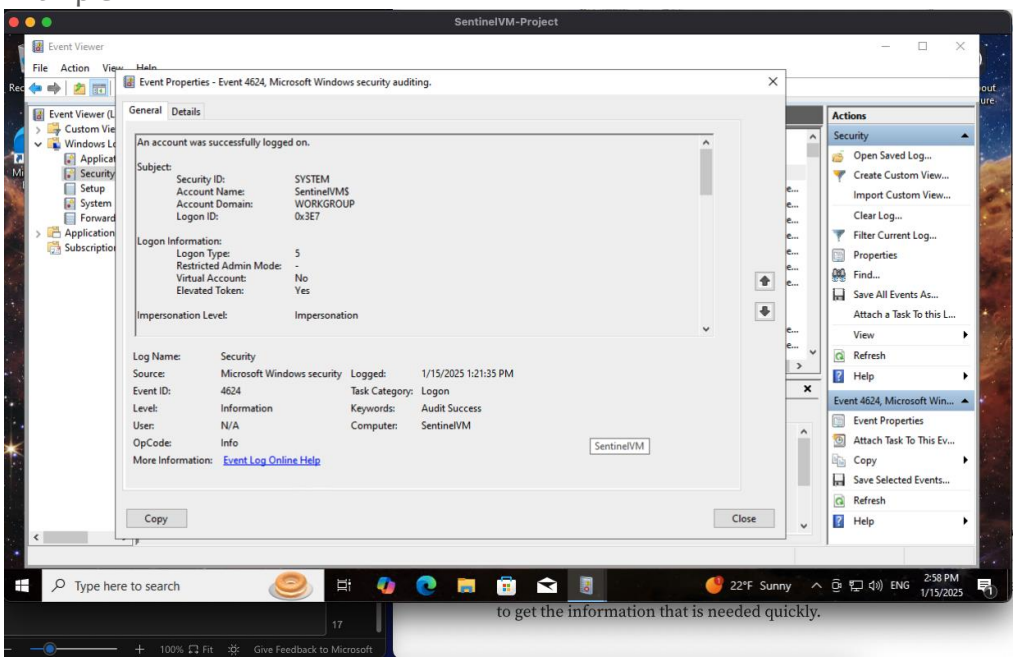


If we head over to ‘Event Viewer’ inside the VM we can observe the many different logs that Windows collects. Our focus is the security events.



The security events we are looking for have the Event ID ‘4624’ which are for successful logon.

Example:



to get the information that is needed quickly.

Kusto Query Language

The purpose of a SIEM such as Azure Sentinel is to bring data like this into one centralised location. In an enterprise, we would want data coming in from all our endpoints and virtual machines to make it easier for a security analyst to get the information that is needed quickly.

Let's check out Sentinel to see these security events. To do this we will go back to our Azure Portal and go into Sentinel tab and go to 'Logs'.

Every SIEM has a search language that allows an analyst to extract data from the logs. In Sentinel, the language is called KQL or Kusto Query Language. It is similar to SPL which is used in SIEM's such as Splunk. Let's look at the commands we are going to use to pull some logs related to the security event ID 4624.

Query:

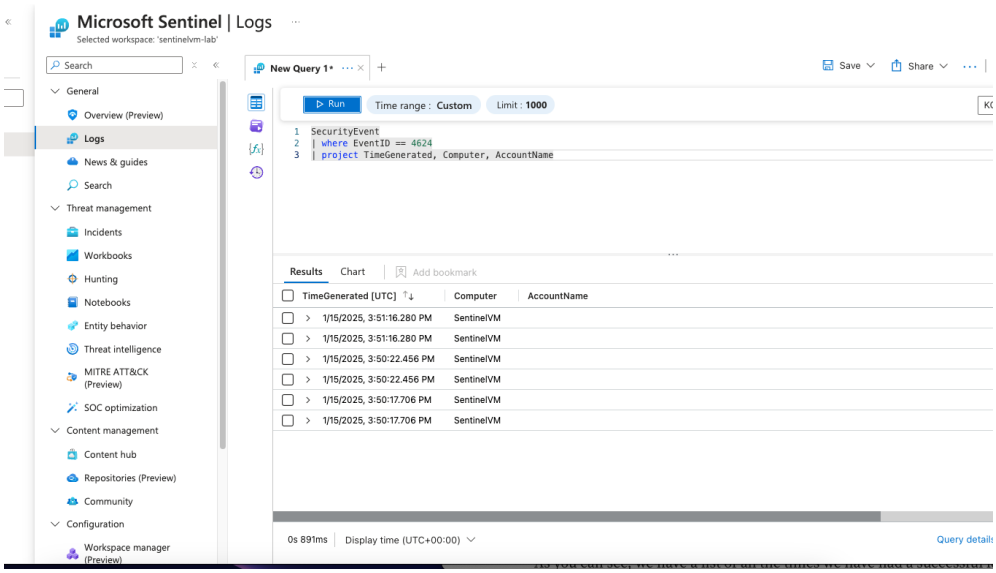
```
SecurityEvent
| where EventID == 4624
| project TimeGenerated, Computer, AccountName
```

'SecurityEvent' refers to the event table we are pulling the data from. All the events we observed in the event viewer are stored there.

The 'where' command filters on a specific category. In our case, we only want events that correspond with successful logons.

'Project' command will specify what data to display when the query is run so, in our case, we want to see the time the logon event occurred, what computer it came from and what account on this computer generated the event.

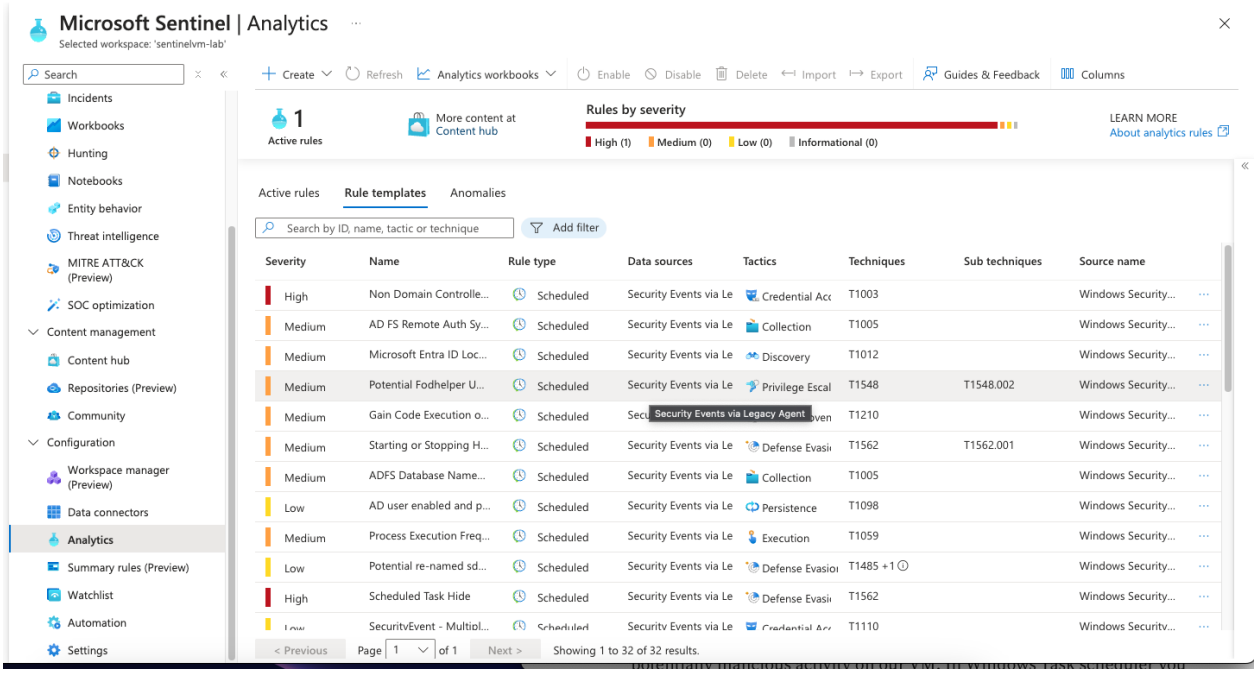
This is the result of the query:



As you can see above, we have a list of all the times we've had a successful logon on our VM. However, we can see that the 'Account Name' field is empty as Sentinel is not putting data into that field. We will configure that through analytic rules.

Writing Analytic Rules and Generating Scheduled Tasks

Sentinel offers many different features and one is that we can have the option to be alerted to certain events. We can do this by setting up analytic rules, these rules will check our VM for the activity that matches the rule logic and generate an alert any time that activity is observed. The alerts will provide details that can help analysts to start their investigation into determining whether the event is a false positive or true positive.



As you can see above, there are a plethora of different rules we can use that come with Sentinel. By clicking on a rule, we can expand it and see what they are, what the rule logic is and to even enable the rule.

Scheduled Task and Persistence Techniques

Let's now create our own custom rule to detect potentially malicious activity on our VM. In Windows Task Scheduler, we have the option to create a scheduled task. A scheduled task is essentially a way to automate certain activities on our machine.

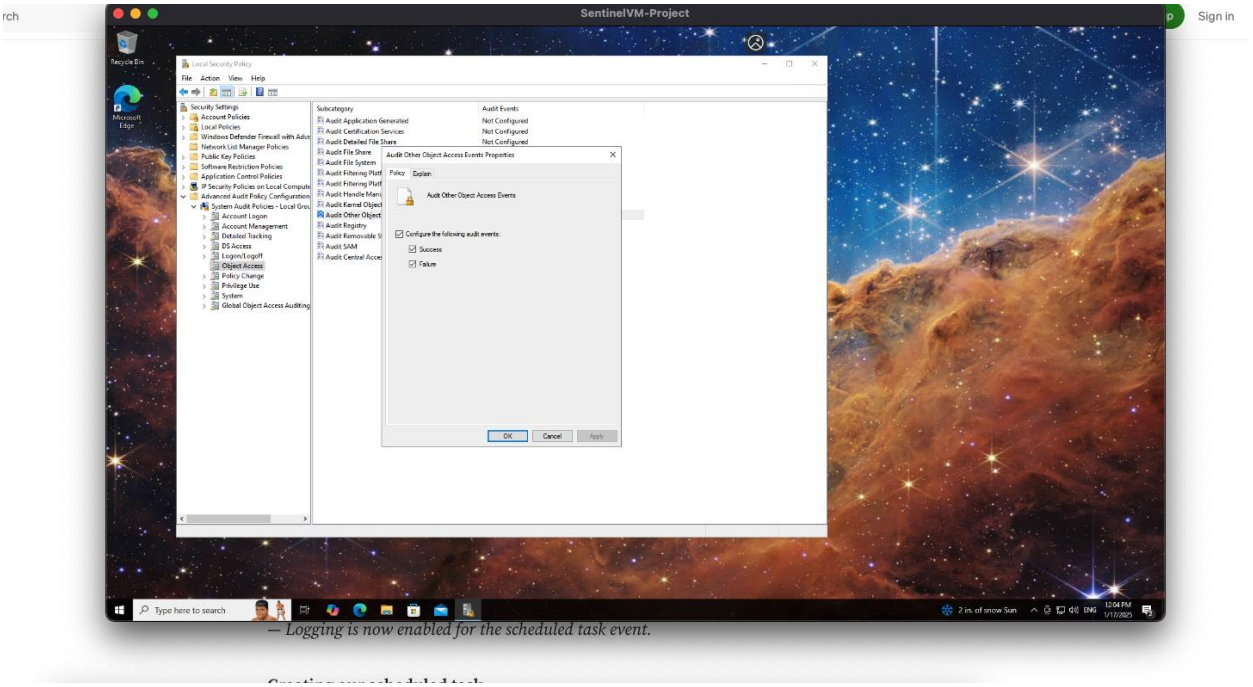
For instance, we can set up a scheduled task that opens google chrome at a certain time every day. While this feature can be harmless most of the time, it can be used as a persistence technique for malicious actors.

According to the MITRE ATT&CK Framework, "Adversaries may abuse task scheduling functionality to facilitate initial or recurring execution of malicious code. Utilities exist within all major operating systems to schedule program or scripts to be executed at a specified date and time".

For this lab, our scheduled task will not be associated with any malicious activity as we will set up a scheduled task that opens up Internet Explorer at a certain time. The goal of this is to simulate a scheduled task and an analytic rule to monitor that task and alert us through Sentinel.

The Windows Security event ID that corresponds to scheduled task creation is 4698. However, these events are not logged by default in the Windows Event Viewer. To enable logging for this event we need to make some changes to the Windows Security policy in our VM.

We need to go back into our VM and head over to 'Local Security Policy' and expand into 'Advanced Audit Policy Configuration'. Through there we head into 'System Audit Policies' and then 'Select Object Access'. The subcategory we are looking for is 'Audit Other Object Access' and we want to open it and enable 'Success' and 'Failure'.

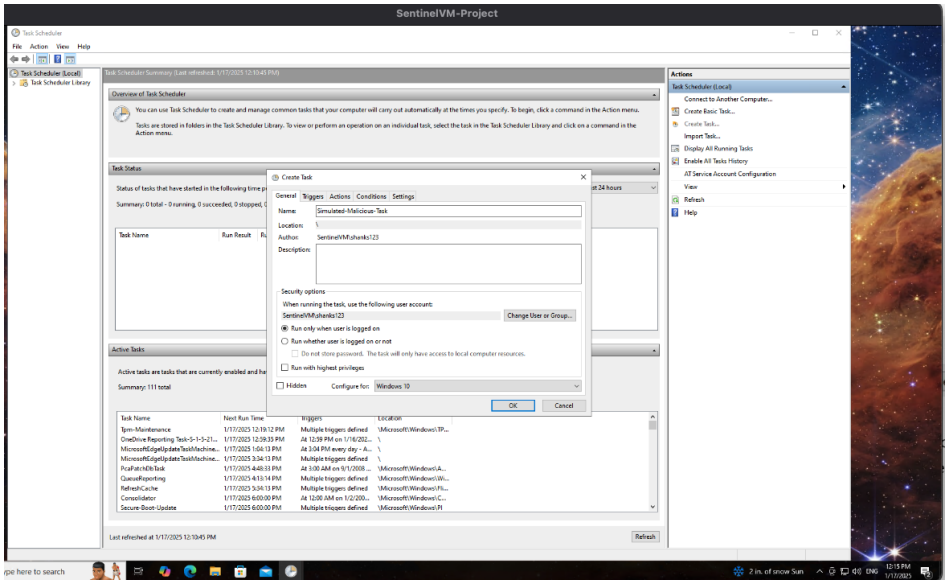


— Logging is now enabled for the scheduled task event.

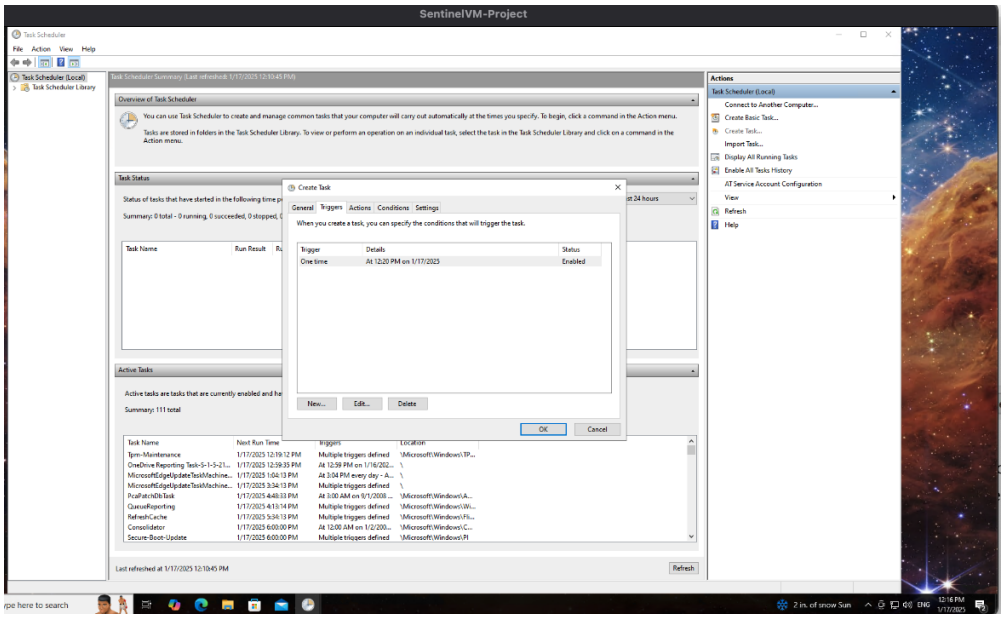
Creating our Scheduled Task

Now that we can log our scheduled task, let's go an create one. To do this, we need to head into 'Windows Task Scheduler' and create a task.

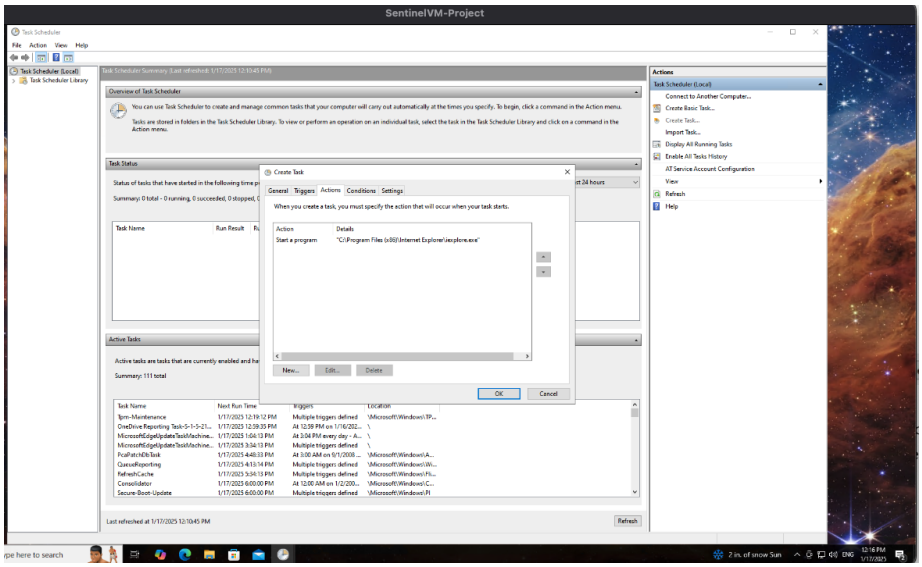
1. Next, we need to give it a name and change the 'Configure for' operating system to Windows 10.



2. Now we navigate to the 'triggers' tab and click 'new' and schedule the task for a time close to our current time.



3. Now we need to go to the 'Actions' tab and we have to select our program. For this we will be using Internet explorer so we click on that.

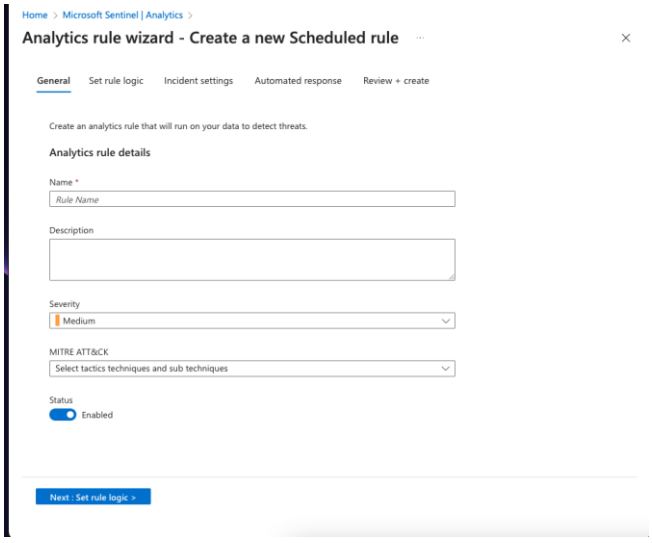


This will create our scheduled task and we can now go to event viewer and search for the event id that corresponds with creating a new scheduled task which is EventID 4698.

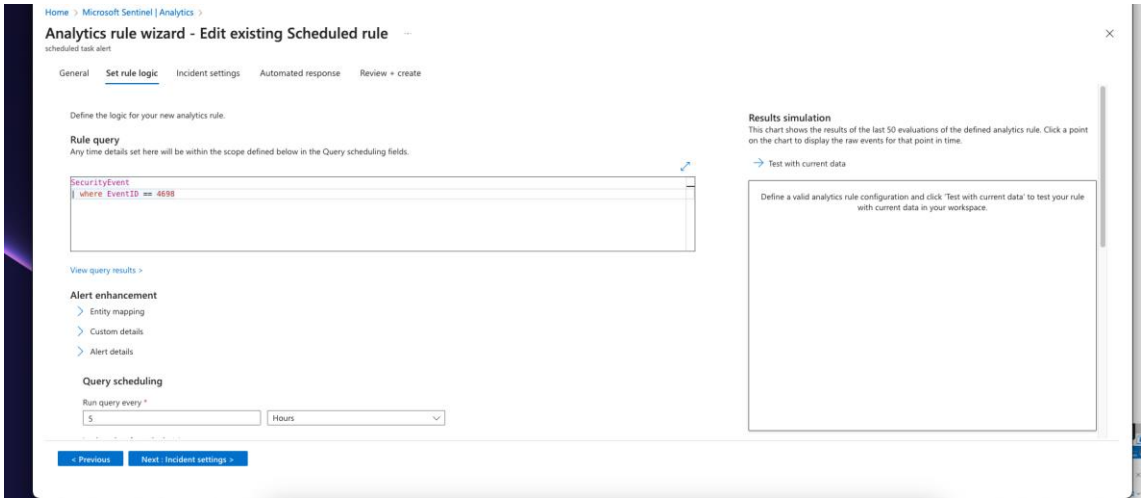
Writing the analytic rule

Now that our scheduled task is configured, we need to head over to Sentinel to write some KQL logic to alert us when a scheduled task is created.

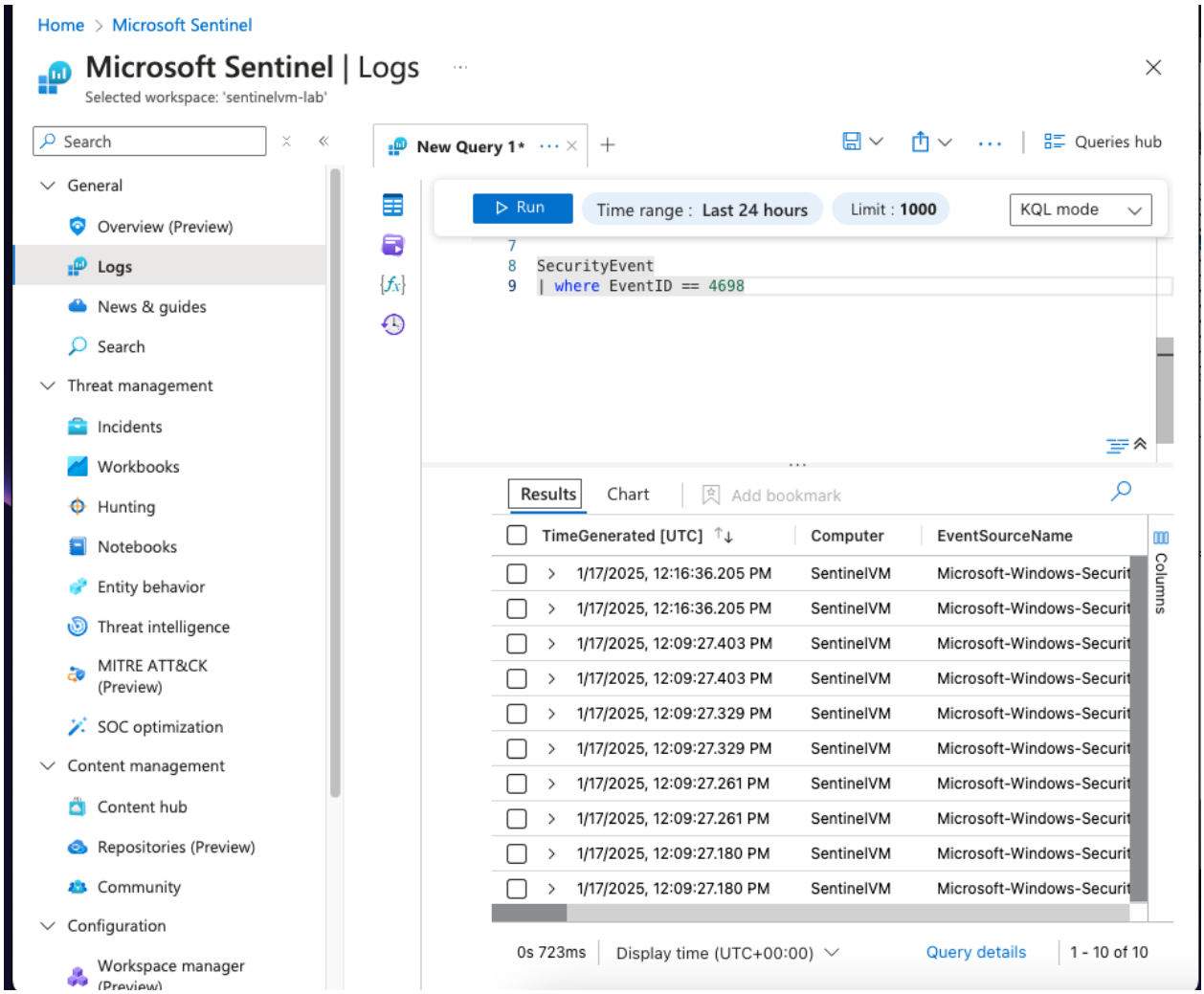
1. We need to head over to sentinel and go to Analytic rules and create a new rule and select the scheduled query option.



2. Once we have given our alert a name, we can set the rule logic and for that we will use the KQL query:

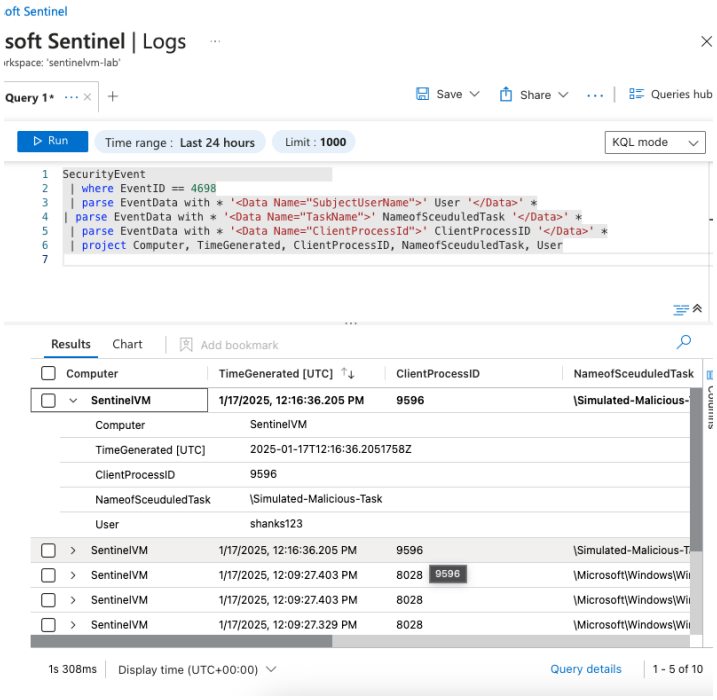


This query will pull instances of scheduled task creation as show here in our logs.



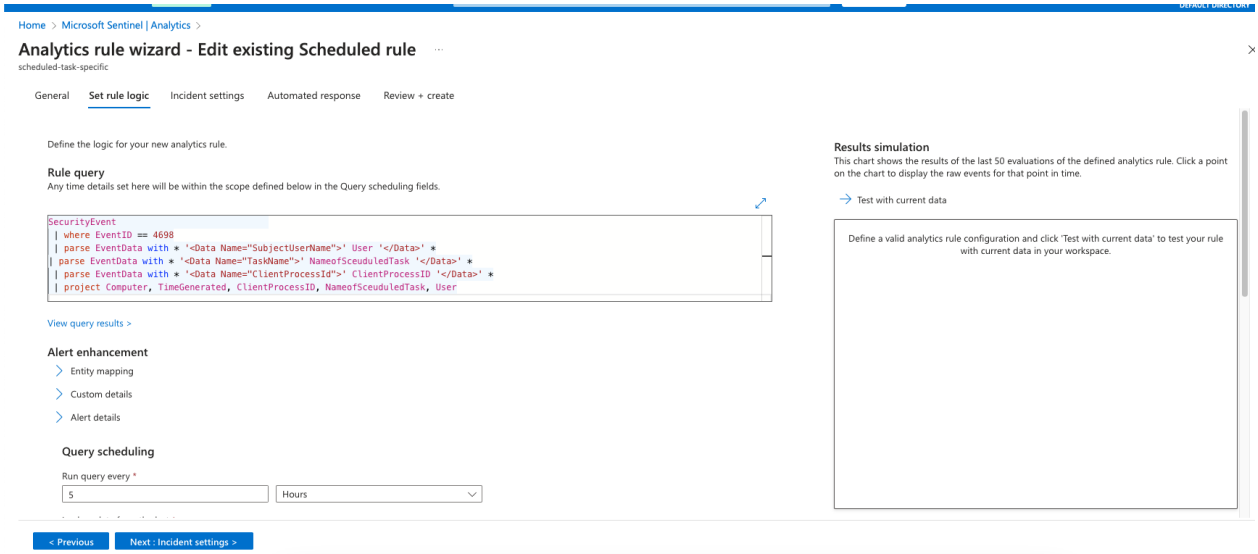
We can expand the logs to see more information regarding the alert. there is alot of useful data in here such as scheduled task, the task name field, the cliendprocessid, the username of the account that created the scheduled task amongst other info.

We can run a different command to display these data fields as columns to have more specific information without all the fluff.



As you can see with this command, we can pull the relevant data fields needed and in the above image we have pulled the fields, User, Taskname, ClientProcessID, time and computer. As you've seen, we are able to generate Event data and place it into its own category for readability.

We can use this new query and create a new alert so that it gives us the query with the exact fields we need.



We can enrich our alerts through the 'Description section in the 'General' tab, enriching allows us to add context to the alerts to make it easier for an analyst to investigate. Using the 'description' feature allows us to put the necessary data into the alert details as 'entities' so the analyst can begin investigating those specific components.

MITRE ATT&CK

Scheduled tasks allow a malicious actor to maintain a foothold in the victim environment. The technique, T1053, refers to the use of scheduled tasks toe execute malicious programs or scripts on a system at a specific time or interval. Attackers can use this method for persistence, privilege escalation or execution of commands on a compromised system.

Sub-Techniques

The **Scheduled Task/Job** technique includes several sub-techniques for different environments and mechanisms:

1. **T1053.001 - Scheduled Task**
    a. **Description**: Abuse of Windows Task Scheduler to execute tasks or programs.
    b. **Examples**:
        i. Using schtasks.exe to create or modify tasks.
        ii. Configuring malicious tasks to execute at system startup.
2. **T1053.002 - Cron**
    a. **Description**: Abuse of Linux/Unix cron jobs to schedule commands or scripts.
    b. **Examples**:
        i. Adding entries to the crontab file to execute malicious scripts at intervals.
3. **T1053.003 - At**
    a. **Description**: Use of the at command to schedule tasks on Windows.
    b. **Examples**:

        i. Using the at.exe command to schedule a malicious task.
4. **T1053.004 - Launchd**
    a. **Description**: Abuse of macOS Launchd to execute tasks.
    b. **Examples**:
        i. Modifying plist files to schedule persistence.
5. **T1053.005 - Systemd Timers**
    a. **Description**: Use of systemd timers in Linux to schedule malicious actions.
    b. **Examples**:
        i. Configuring a systemd service or timer to execute malicious code.
6. **T1053.006 - Container Orchestration Job**
    a. **Description**: Abuse of container orchestration tools (like Kubernetes CronJobs) to schedule tasks.
    b. **Examples**:
        i. Deploying malicious containers on a scheduled basis.

Detection and Mitigation

As observed, monitoring and logging of specific windows event ID's is best used to detect these types of activities. When it comes to mitigation there are a few techniques that can be utilised, below are a few:

MITRE ID M1019: User Account Management suggests that user account privileges should be limited to only authorise admins to create scheduled tasks on remote systems.

MITRE ID M1026: protect critical processes and task schedulers from being tampered with. This can be done by restricting administrative access to Task Scheduler and relate tools. Also using endpoint protection solutions to monitor task-related processes.

MITRE ID M1028: Harden operating system configuration to restrict unauthorised task scheduling. This can be done by enforcing group policy to restrict who can schedule tasks on windows.

Azure Threat Hunting

Overview

The above scheduled task simulation is an example of a threat and we can use sentinel logs to proactively hunt threats.

Proactive threat hunting is an essential practice in modern cybersecurity operations. Unlike traditional reactive measures that rely on alerts and logs generated by automated systems, threat

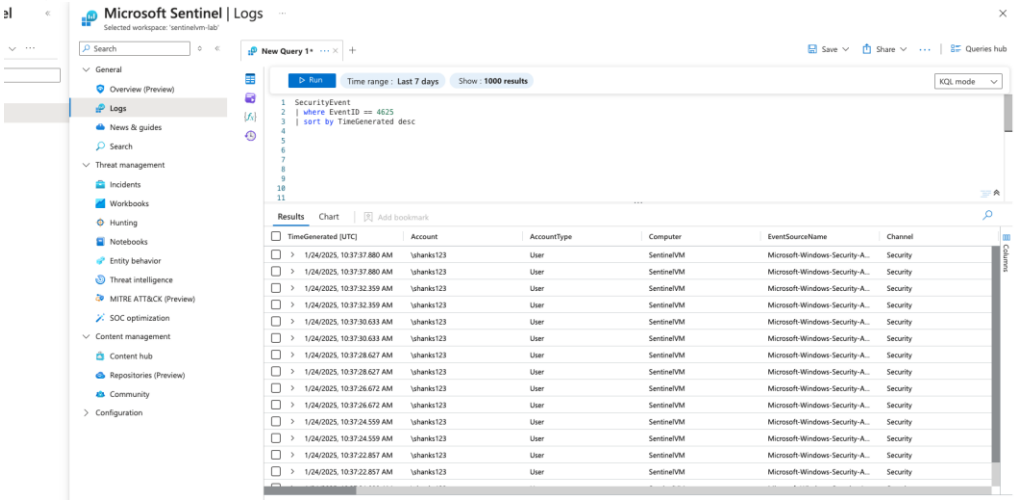
hunting involves actively searching for indicators of compromise (IOCs) and anomalous behaviors within an organization's environment. This approach helps uncover hidden threats that may evade standard detection methods, ultimately strengthening the overall security posture.

Azure Sentinel, offers advanced capabilities for real-time threat detection, analysis, and response. By integrating data from multiple sources and utilizing tools like Kusto Query Language (KQL), Azure Sentinel enables security analysts to gain deep insights into their infrastructure and respond to threats effectively.

Simulation 1: Brute Force Login

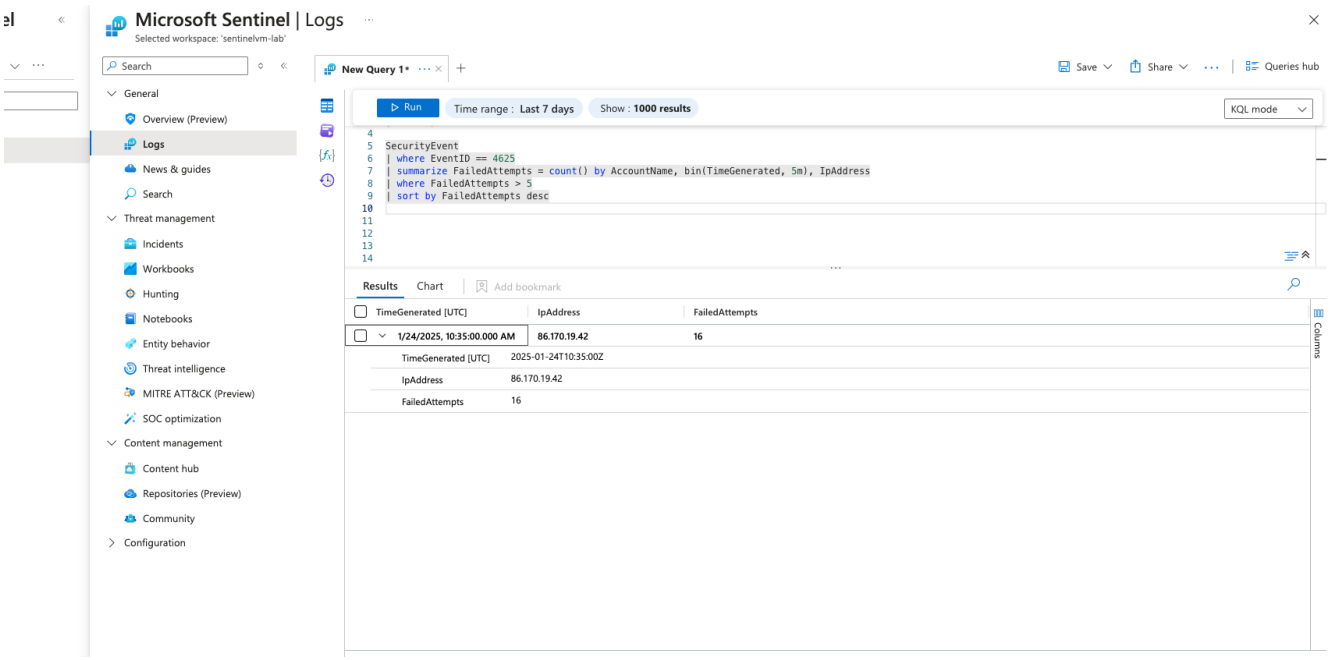
This simulation is brute force login, this is when an attacker attempts to login to a device with a username and password however they do not know it so they attempt various usernames and passwords. Our SIEM, Sentinel, can log this and this can alert us of credential access which can indicate a malicious user is trying to gain access to one of our systems. adversaries may use brute force techniques to gain access to accounts when passwords are unknown or when password hashes are obtained. Without knowledge of the password for an account or set of accounts, an adversary may systematically guess the password using a repetitive or iterative mechanism. Brute forcing passwords can take place via interaction with a service that will check the validity of those credentials or offline against previously acquired credential data, such as password hashes.

The event ID for this technique is 4625 and this is a good indication that our systems are being targeted. Brute force can happen at various phases commonly in the initial access phase.

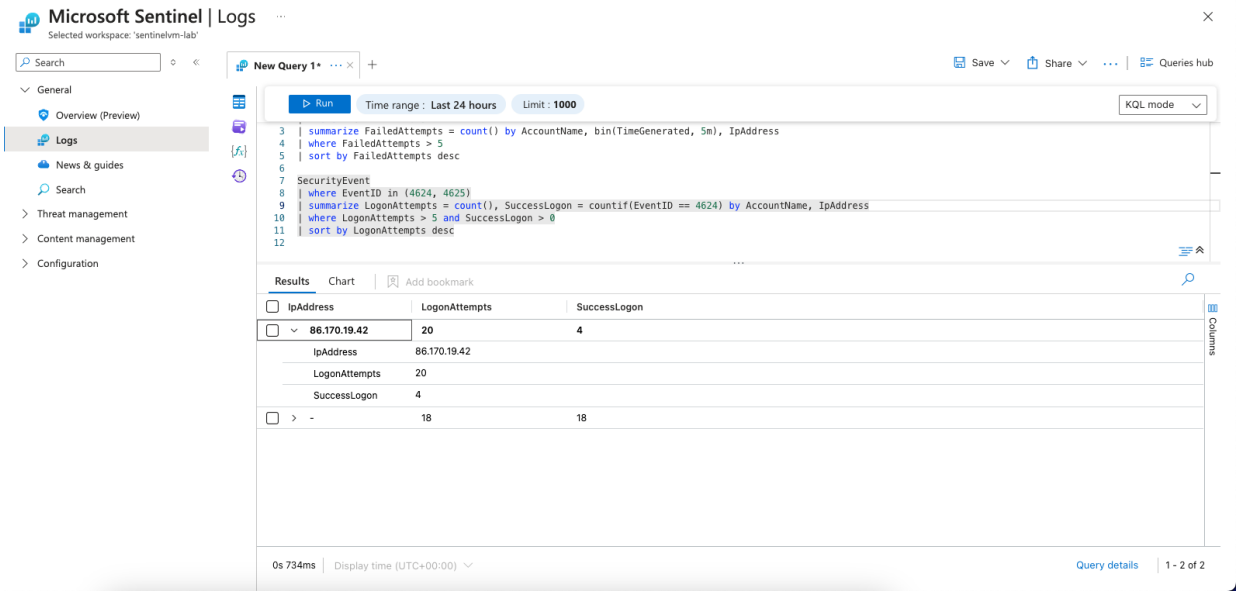


Above, we can see that when we query for EventID 4625, we can see loads of results which are consecutive if we look at the time of the logs which shows us that there are not just someone inputting their password wrong but more of a sign of a brute force. Attackers will try variations of passwords to guess the correct one which would result in consecutive attempts in a short timeframe.

As an analyst, in the event we get these scenarios, we would need to investigate and conclude whether we are being targeted or if this is a false flag. We can add to our query to further investigate and acquire the relevant information we need:



Above we can see that with our new query we get some valuable information. Here we can see the number of failed attempts as well as the IP address that caused this. We can use this information to further our investigation and come to a conclusion of whether this is a brute force attempt.

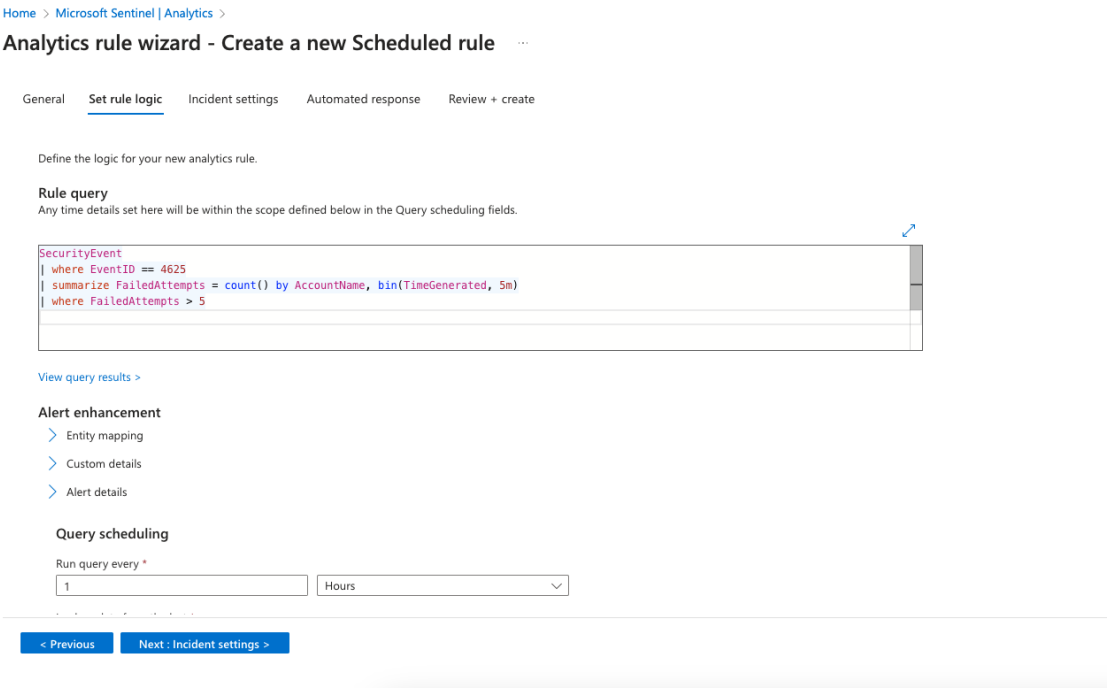


With this query we can compare the successful logons with the failed logons and we can see that there was a total of 20 logons and four were successful. From our previous query we saw that there were 16 failed logons from the same IP address.

These two queries can tell us a lot of information. First, we know the IP address that is targeting us as well as how many attempts they made. Since there are some successful logons, we can see that the attacker managed to access the system. In a real scenario this will be very problematic as this

means that the attacker has access to our system and we are compromised. From here an attacker can pivot through to achieve their goals.

As an analyst we need to prevent such actions so from this we can create some countermeasures to alert us as well as contain the attack if it arises.



Here we created an analytic rule which will alert us if there is a failed logon attempt. We have set it to 5 logon attempts as this will allow us to thin out any false flags. So, if an attacker wants to gain access to our systems we will know if they attempt multiple logins which will allow us to implement countermeasures. This allows us to not have to actively search but rather wait till the alert is triggered which eases the role of the analyst.

Conclusion

This project has successfully demonstrated the implementation and utility of Azure Sentinel in a cybersecurity context. By configuring Azure resources, adhering to security best practices, and analysing logs from Windows Security Events, the project showcased the capabilities of Azure Sentinel for real-time monitoring, threat detection, and incident response. Furthermore, leveraging the MITRE ATT&CK framework enabled a deeper understanding of adversary tactics and techniques, aligning detection and response efforts with industry standards. This hands-on lab experience not only solidified foundational knowledge of security information and event management (SIEM) systems but also highlighted the importance of integrating advanced tools like Azure Sentinel into modern cybersecurity operations.