

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

Description	1
Introduction.....	3
Project Goals	3
Key Learning Objectives.....	4
Project Topology.....	5
Setup and Implementation.....	7
Step 1: Create Azure Account.....	7
Step 2: Create a Resource Group.....	7
Step 3: Deploy Virtual Machine	7
Network and Virtual Machine Security.....	9
Windows Defender	10
Setup	10
Log Analytics and Microsoft Sentinel.....	12
Setup	12
Injecting data into Sentinel.....	14
Remote Accessing and Generating Security Events.....	16
Kusto Query Language	18
Writing Analytic Rules and Generating Scheduled Tasks	19
Scheduled Task and Persistence Techniques.....	19
Creating our Scheduled Task	21
Writing the analytic rule	22
MITRE ATT&CK.....	25
Sub-Techniques	25
Detection and Mitigation.....	26
Conclusion	26

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.

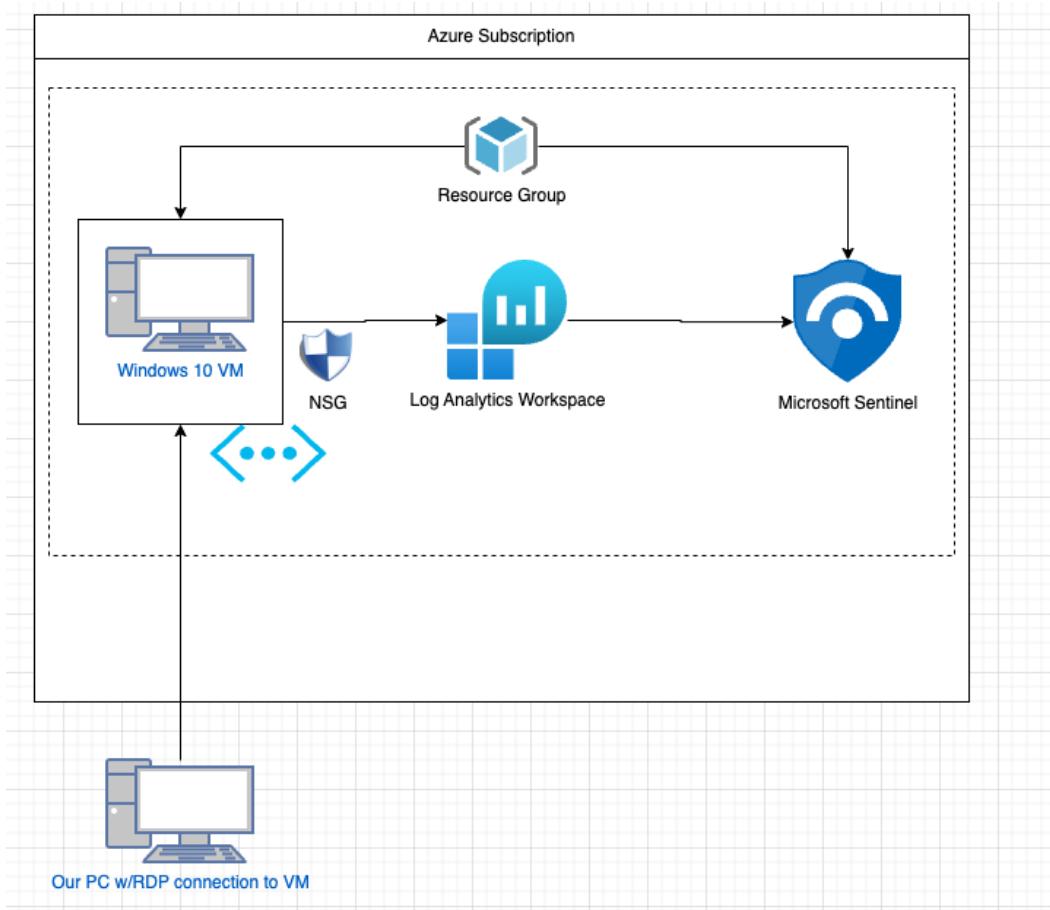
By achieving these goals, this project will demonstrate my ability to work with cutting-edge cybersecurity tools and platforms, reinforcing my readiness to tackle real-world challenges in a professional setting.

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

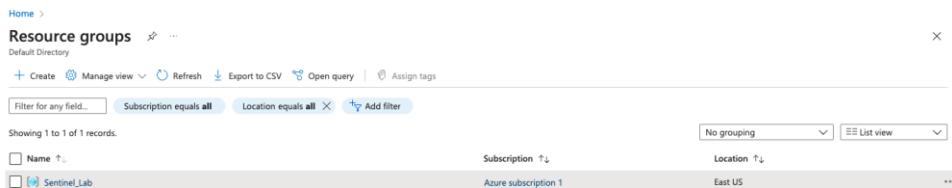
Step 1: Create Azure Account

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.



The screenshot shows the Azure Resource Groups blade. At the top, there are buttons for 'Create', 'Manage view', 'Refresh', 'Export to CSV', 'Open query', and 'Assign tags'. Below these are filter options: 'Subscription equals all', 'Location equals all', and 'Add filter'. A message indicates 'Showing 1 to 1 of 1 records.' The main table lists one resource group: 'Sentinel_Lab' under 'Name', 'Azure subscription 1' under 'Subscription', and 'East US' under 'Location'. There are also 'No grouping' and 'List view' dropdowns at the bottom of the table.

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.

Home > Virtual machines >

Create a virtual machine

Help me create a low cost VM | Help me create a VM optimized for high availability | Help me choose the right VM size for my workload

! This subscription may not be eligible to deploy VMs of certain sizes in certain regions.

Project details

Select the subscription to manage deployed resources and costs. Use resource groups like folders to organize and manage all your resources.

Subscription *

Resource group * [Create new](#)

Instance details

Virtual machine name * ✓

Region *

Availability options

Security type [Configure security features](#)

Image * Windows 10 Pro, version 22H2 - x64 Gen2 (free services eligible) [See all images](#) | [Configure VM generation](#)

VM architecture Arm64 x64
! Arm64 is not supported with the selected image.

Run with Azure Spot discount

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

Home > Virtual machines >

Create a virtual machine

Help me create a low cost VM | Help me create a VM optimized for high availability | Help me choose the right VM size for my workload

! You are in the free trial period. Costs associated with this VM can be covered by any remaining credits on your subscription. [Learn more](#)

Size * [See all sizes](#) [View pricing history and compare prices in nearby regions](#)

Maximum price you want to pay per hour (USD) Enter a price greater than or equal to the hardware costs (US\$0.01411)

Enable Hibernation
! Hibernate does not currently support Azure Spot. [Learn more](#)

Administrator account

Username * ✓

Password * ✓

Confirm password * ✓

Inbound port rules

Select which virtual machine network ports are accessible from the public internet. You can specify more limited or granular network access on the Networking tab.

Public inbound ports * None Allow selected ports

Select inbound ports *

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

The screenshot shows the Azure portal's deployment overview for a VM named 'CreateVm-MicrosoftWindowsDesktop.Windows-10-win10-20230808191211'. The deployment status is 'Deployment is in progress'. Key details include:

- Deployment name:** CreateVm-MicrosoftWindowsDesktop.Windows-10-win10-20230808191211
- Subscription:** Azure subscription 1
- Start time:** 8/8/2023, 7:20:20 PM
- Correlation ID:** fff6cd9-a4e6-4cb5-a291-196d4247a031
- Resource group:** Sentinel_Lab

Deployment details:

Resource	Type	Status	Operation details
Sentinelvm	Microsoft.Compute/virtualMachines	Created	Operation details
sentinelvm429	Microsoft.Network/networkInterfaces	Created	Operation details
Sentinelvm-nsg	Microsoft.Network/networkSecurityGroups	OK	Operation details
Sentinelvm-vnet	Microsoft.Network/virtualNetworks	OK	Operation details
Sentinelvm-ip	Microsoft.Network/publicIPAddresses	OK	Operation details

Give feedback: Tell us about your experience with deployment.

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.

The screenshot shows the Azure portal's network settings for a VM named 'SentinelVM'. The 'Networking' tab is selected, specifically the 'Network settings' sub-tab. Key details include:

- Network interface:** sentinelvm810
- Virtual network / subnet:** SentinelVM-vnet / default
- Public IP address:** 40.117.169.250
- Private IP address:** 10.0.0.4
- Admin security rules:** 0 (Configure)
- Load balancers:** 0 (Configure)
- Application security groups:** 0 (Configure)
- Network security group:** SentinelVM-nsg
- Accelerated networking:** Disabled
- Effective security rules:** 0

Rules:

Priority ↑	Name	Port	Protocol	Source	Destination	Action
300	RDP	3389	TCP	Any	Any	Allow
65000	AllowVnetInBound	Any	Any	VirtualNetwork	VirtualNetwork	Allow
65001	AllowAzureLoadBalancerInBound	Any	Any	AzureLoadBalancer	Any	Allow
65500	DenyAllInBound	Any	Any	Any	Any	Deny

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 least 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.

The screenshot shows the Microsoft Defender for Cloud Overview page. The left sidebar has sections for General, Cloud Security (Security posture, Regulatory compliance, Workload protections, Data security, Firewall Manager, DevOps security), Management (Environment settings, Security solutions, Workflow automation), and a navigation bar with Home, Subscriptions, What's new, and a search bar. A message at the top right says 'One subscription doesn't have the default policy assigned. To review the list of subscriptions, open the Security Policy page.' Below this are four main metrics: 1 Azure subscription, 0 Assessed resources, -- Attack paths, and -- Security alerts. A large central box titled 'Security posture' displays 0 Critical recommendations, 0 Attack paths, and 0/0 Overdue recommendations. It also shows an environment risk and secure score with a legend: Critical (red), High (orange), Medium (yellow), Low (blue), and Not evaluated (grey). A total secure score bar is at 0%. At the bottom, there are links for 'Explore your security posture >' and 'View details'.

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:

The screenshot shows the Microsoft Defender for Cloud Settings | Defender plans page. It displays two plans:

Plan	Pricing*	Resource quantity	Monitoring coverage	Status
Foundational CSPM	Free Details >		Full	<input checked="" type="button"/> Off <input type="button"/> On
Defender CSPM	\$5/Billable resource/Month Details >	1 resources		<input type="button"/> Off <input checked="" type="button"/> On

Below this, there's a section for Cloud Workload Protection (CWP) with two sub-plans:

Plan	Pricing*	Resource quantity	Monitoring coverage	Status
Servers	Plan 2 (\$15/Server/Month) Change plan >	1 servers		<input type="button"/> Off <input checked="" type="button"/> On
App Service	\$15/Instance/Month Details >	0 instances		<input type="button"/> Off <input checked="" type="button"/> On

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:

The screenshot shows the Microsoft Defender for Cloud | Workload protections page. It displays the following information:

- Defender for Cloud coverage:** Shows 2 total items, with 1 server and 1 Resource Manager subscription. Upgrade buttons are available for both.
- Security alerts:** A chart showing 4 alerts across the period from 23 Mar to 13 May. A legend indicates High severity (0), Medium severity (0), and Low severity (0).
- Advanced protection:** A section showing insights and upgrade options for containers and Kubernetes.

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.

The screenshot shows the 'Microsoft Sentinel' blade in the Azure portal. At the top, there are navigation links for 'Home', 'Microsoft Sentinel', 'Default Directory', and a '...' button. Below this is a toolbar with 'Create', 'Manage view', 'Refresh', 'Export to CSV', 'Open query', and 'View incidents'. A search bar says 'Filter for any field...'. There are three filter buttons: 'Subscription equals all', 'Resource group equals all', and 'Location equals all'. Below the filters is a message 'Showing 0 to 0 of 0 records.' and sorting options for 'Name', 'Resource group', 'Location', 'Subscription', and 'Directory'. The main content area features a large shield icon with a circular arrow. Below it, the text 'No Microsoft Sentinel to display' is centered. A subtext reads 'See and stop threats before they cause harm, with SIEM reinvented for a modern world. Microsoft Sentinel is your birds-eye view across the enterprise.' A blue 'Create Microsoft Sentinel' button is at the bottom, along with a 'Learn more' link and a 'Give feedback' button.

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

Validation passed

Basics Tags Review + Create

Log Analytics workspace by Microsoft

Basics

Subscription	Azure subscription 1
Resource group	Sentinel_Lab
Name	SentinelVM-Lab
Region	East US

Pricing

Pricing tier	Pay-as-you-go (Per GB 2018)
--------------	-----------------------------

The cost of your workspace depends on the volume of data ingested and how long it is retained. Regional pricing details are available on the [Azure Monitor pricing page](#). You can change to a different pricing tier after the workspace is created. [Learn more](#) about Log Analytics pricing models.

Tags

None

Create « Previous Download a template for automation

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

Home > Microsoft Sentinel >

Add Microsoft Sentinel to a workspace

+ Create a new workspace ⏪ Refresh

Microsoft Sentinel offers a 31-day free trial. See [Microsoft Sentinel pricing](#) for more details.

Workspace ↑↓	Location ↑↓	ResourceGroup ↑↓	Subscription ↑↓	Directory ↑↓
SentinelVM-Lab	eastus	sentinel_lab	Azure subscription 1	Default Directory

Add Cancel

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.

The screenshot shows the Microsoft Sentinel Incidents page. The left sidebar has a 'Threat management' section with 'Incidents' selected. The main area displays a message: 'No incidents were found'. Below this, sections explain what incidents are, how they work, and activities you can perform. It includes icons for 'View related alerts' and 'Triage and investigate'.

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.

The screenshot shows the Microsoft Sentinel Data connectors page. The left sidebar has a 'Configuration' section with 'Data connectors' selected. The main area shows a message about removed connectors and a list of 'Data connectors' with sections for 'Getting started' and 'Featured data connectors'.

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.

The screenshot shows the Microsoft Sentinel Content hub interface. On the left, there's a navigation sidebar with various sections like Threat management, Content hub, Configuration, and Automation. The main area displays a grid of content items. At the top, it shows 375 Solutions, 307 Standalone contents, 0 Installed, and 0 Updates. A search bar at the top allows filtering by content title, status, provider, support, category, and content source. Below the search bar is a table with columns for Content title, Status, Content source, Provider, Support, Category, and Content type. Some entries are marked as 'FEATURED' or 'PREVIEW'. One entry, 'Windows Security Events via AMA', is highlighted.

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.

This screenshot shows the Microsoft Sentinel Content hub after performing a search for 'Windows security Events via AMA'. The search results are displayed in a grid format. The first result, 'Windows Security Events via AMA', is shown in more detail on the right side of the screen. It includes information such as the provider (Microsoft), support (Microsoft), version (3.0.9), and a note about installing the solution. There are also links for Release Notes and View details.

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

The screenshot shows two windows side-by-side. On the left is the 'Windows Security Events via AMA' page, which displays a status bar ('Disconnected Status'), a provider ('Microsoft Provider'), and a log received indicator ('Last Log Received'). It includes a description of how to stream security events from Windows machines, a chart showing data received over time (from January 7 to January 11), and links to workbooks, queries, and analytics rules templates. On the right is the 'Create Data Collection Rule' dialog box, specifically the 'Review + create' tab. It shows validation passed, basic settings like a rule name ('sentinel-watcher'), subscription ('Azure subscription 1'), and resource group ('Sentinel_Lab'), and selected resources ('sentinelvm' of type 'microsoft.compute/virtualmachines'). A list of selected events ('AllEvents') is also shown.

Remote Accessing and Generating Security Events

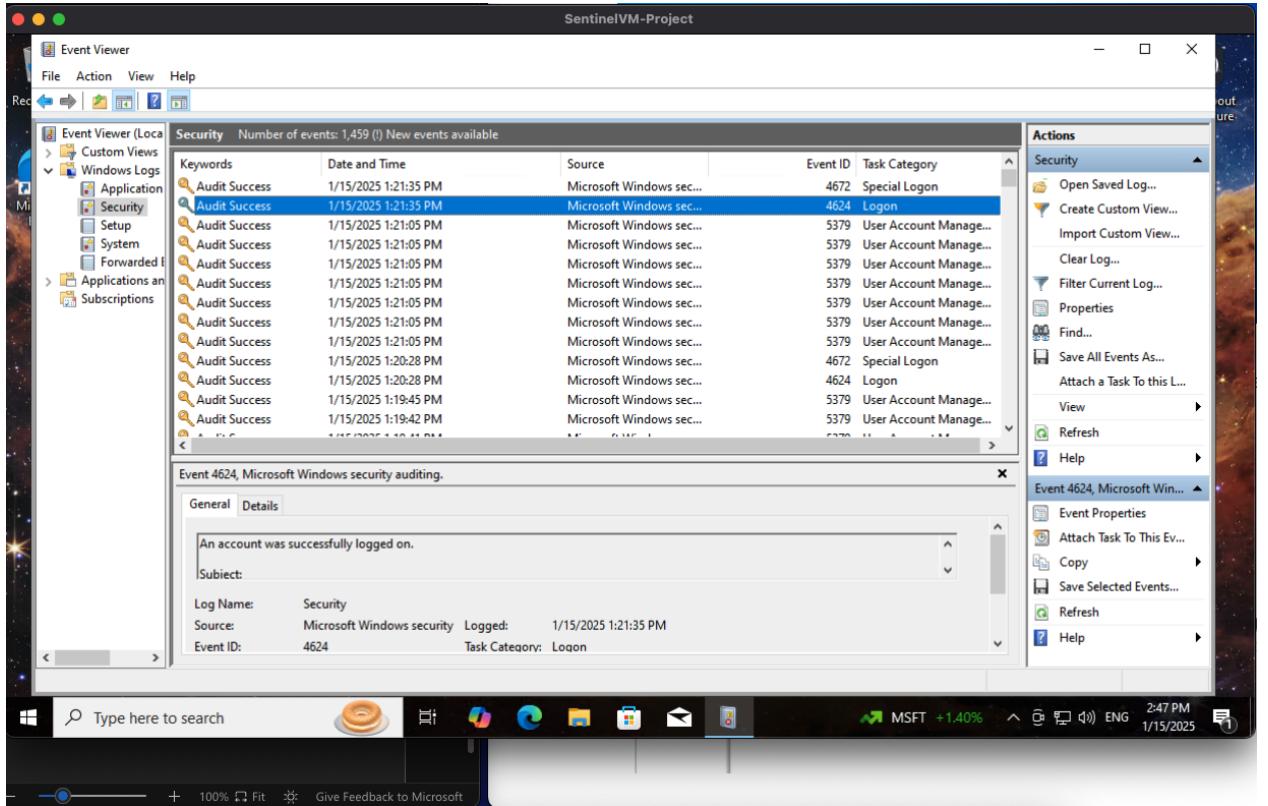
Now that our VM is connected to sentinel and our Log Analytics Workspace, we need to ingest some data into our Logs. To do this, we simply need to perform some actions on the Windows 10 VM that will generate security alerts.

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.

- 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.

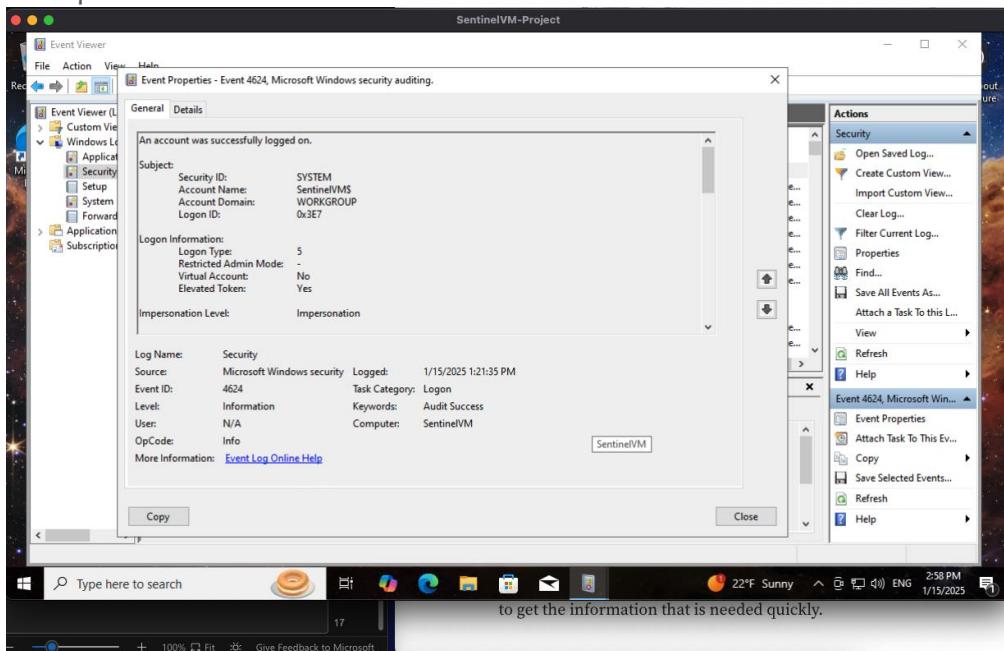
The screenshot shows the 'SentinelVM | Connect' blade in the Azure portal. The left sidebar lists navigation options: Overview, Activity log, Access control (IAM), Tags, Diagnose and solve problems, Connect (selected), Bastion, Windows Admin Center, Networking, Network settings, Load balancing, Application security groups, Network manager, Settings, Availability + scale, Security, Backup + disaster recovery, Operations, and Monitoring. The main content area shows the connection status ('Connecting using Public IP address | 40.117.169.250'), connection details (Admin username 'shanks123', Port 3389, Just-in-time policy 'Configure'), and a 'Most common' section for connecting to a local machine via Native RDP. It includes a note about using native RDP without additional software and provides a public IP address (40.117.169.250) and download links for an RDP file.

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:

The screenshot shows the Microsoft Sentinel interface with the 'Logs' workspace selected. On the left, a navigation sidebar lists categories like General, Threat management, Content management, and Configuration. The 'Logs' section is currently active. The main area displays a 'New Query 1*' window with the following KQL code:

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

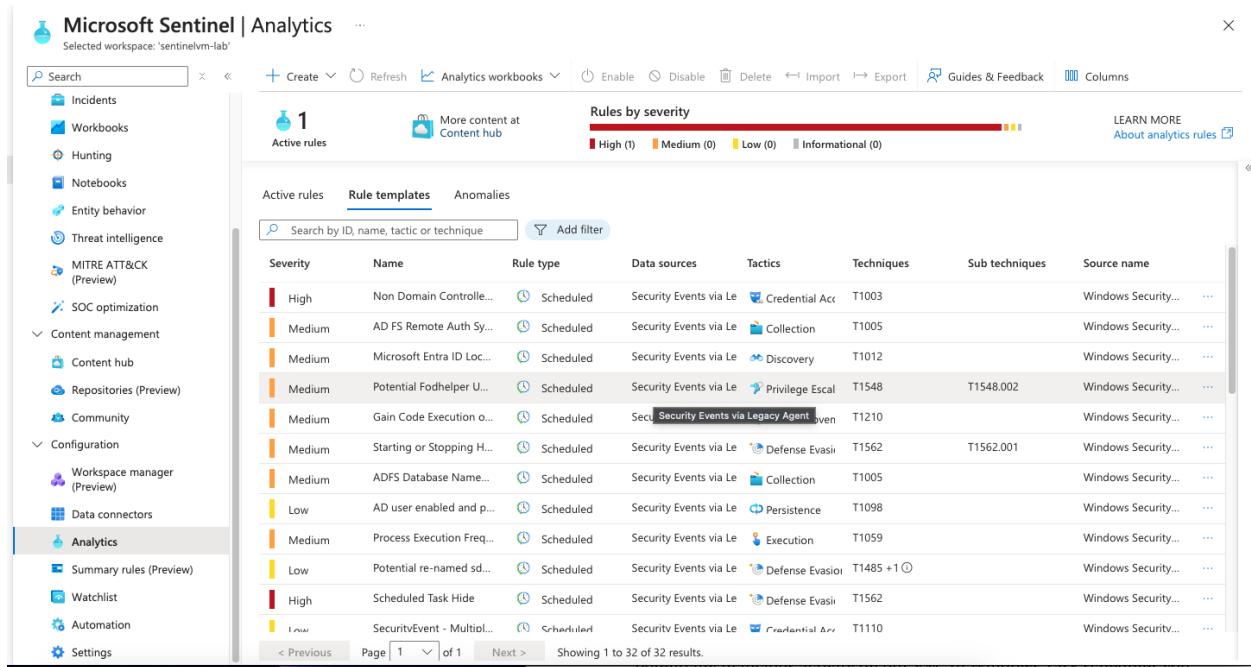
Below the query editor, the results are presented in a table with columns: TimeGenerated [UTC], Computer, and AccountName. The results show several logon events for the computer 'SentinelVM' occurring between 1/15/2025, 3:51:16.280 PM and 1/15/2025, 3:50:22.456 PM.

TimeGenerated [UTC]	Computer	AccountName
> 1/15/2025, 3:51:16.280 PM	SentinelVM	
> 1/15/2025, 3:51:16.280 PM	SentinelVM	
> 1/15/2025, 3:50:22.456 PM	SentinelVM	
> 1/15/2025, 3:50:22.456 PM	SentinelVM	
> 1/15/2025, 3:50:17.706 PM	SentinelVM	
> 1/15/2025, 3:50:17.706 PM	SentinelVM	

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.



The screenshot shows the Microsoft Sentinel Analytics interface. The left sidebar includes options like Incidents, Workbooks, Hunting, Notebooks, Entity behavior, Threat intelligence, MITRE ATT&CK (Preview), SOC optimization, Content management, Content hub, Repositories (Preview), Community, Configuration, Workspace manager (Preview), Data connectors, Analytics (selected), Summary rules (Preview), Watchlist, Automation, and Settings. The main pane displays a list of 'Active rules' with 1 rule found. The rule details are as follows:

Severity	Name	Rule type	Data sources	Tactics	Techniques	Sub techniques	Source name
High	Non Domain Controle...	Scheduled	Security Events via Le...	Credential Acc...	T1003		Windows Security...
Medium	AD FS Remote Auth Sy...	Scheduled	Security Events via Le...	Collection	T1005		Windows Security...
Medium	Microsoft Entra ID Loc...	Scheduled	Security Events via Le...	Discovery	T1012		Windows Security...
Medium	Potential Fodhelper U...	Scheduled	Security Events via Le...	Privilege Escal...	T1548	T1548.002	Windows Security...
Medium	Gain Code Execution o...	Scheduled	Security Events via Le...	Legacy Agent	T1210		Windows Security...
Medium	Starting or Stopping H...	Scheduled	Security Events via Le...	Defense Evasi...	T1562	T1562.001	Windows Security...
Medium	ADFS Database Name...	Scheduled	Security Events via Le...	Collection	T1005		Windows Security...
Low	AD user enabled and p...	Scheduled	Security Events via Le...	Persistence	T1098		Windows Security...
Medium	Process Execution Freq...	Scheduled	Security Events via Le...	Execution	T1059		Windows Security...
Low	Potential re-named sd...	Scheduled	Security Events via Le...	Defense Evasi...	T1485 +1	T1110	Windows Security...
High	Scheduled Task Hide	Scheduled	Security Events via Le...	Defense Evasi...	T1562		Windows Security...
Low	SecurityEvent - Multipl...	Scheduled	Security Events via Le...	Credential Acc...	T1110		Windows Security...

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.

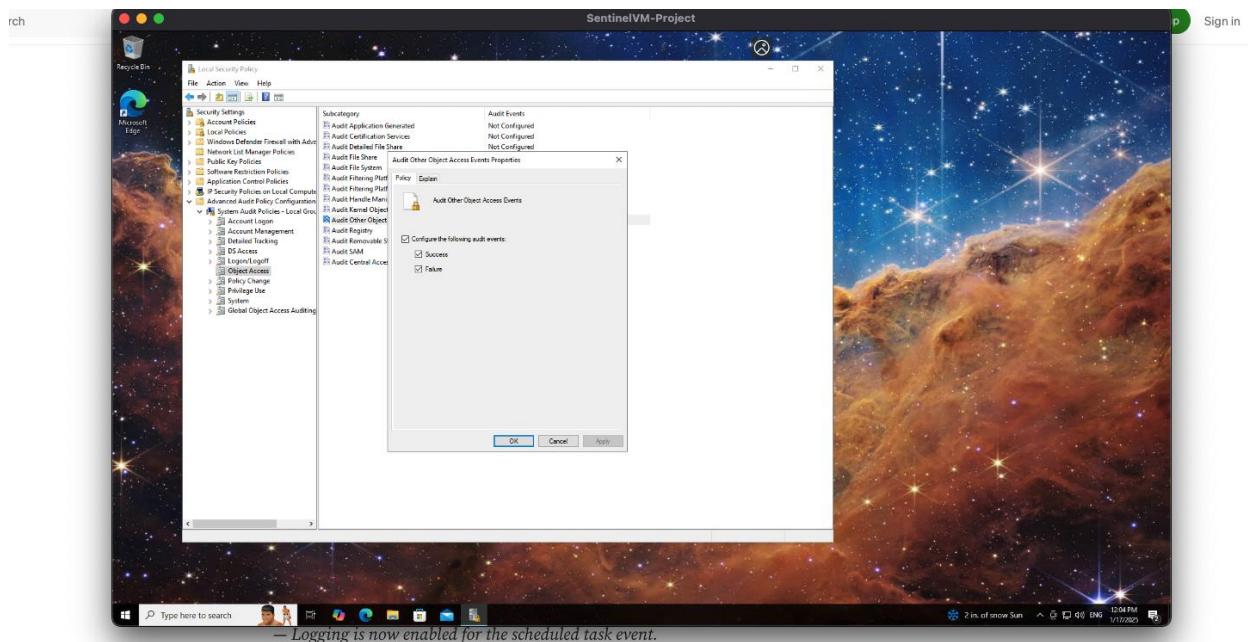
For instance, we can set up a scheduled task that opens google chrome at a certain time every day. While this feature can be harmless most of the time, it can be used as a persistence technique for malicious actors.

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.

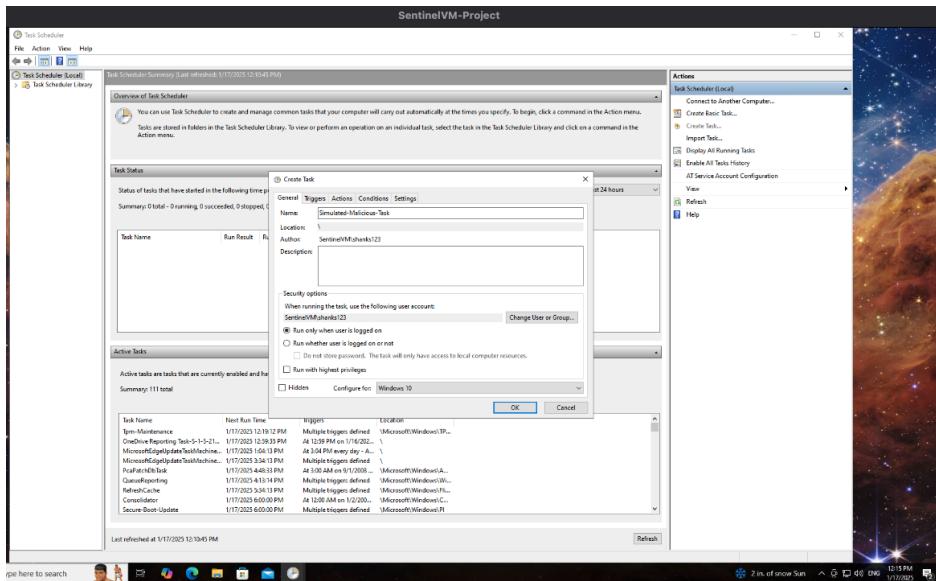
We need to go back into our VM and head over to ‘Local Security Policy’ and expand into ‘Advanced Audit Policy Configuration’. Through there we head into ‘System Audit Policies’ and then ‘Select Object Access’. The subcategory we are looking for is ‘Audit Other Object Access’ and we want to open it and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



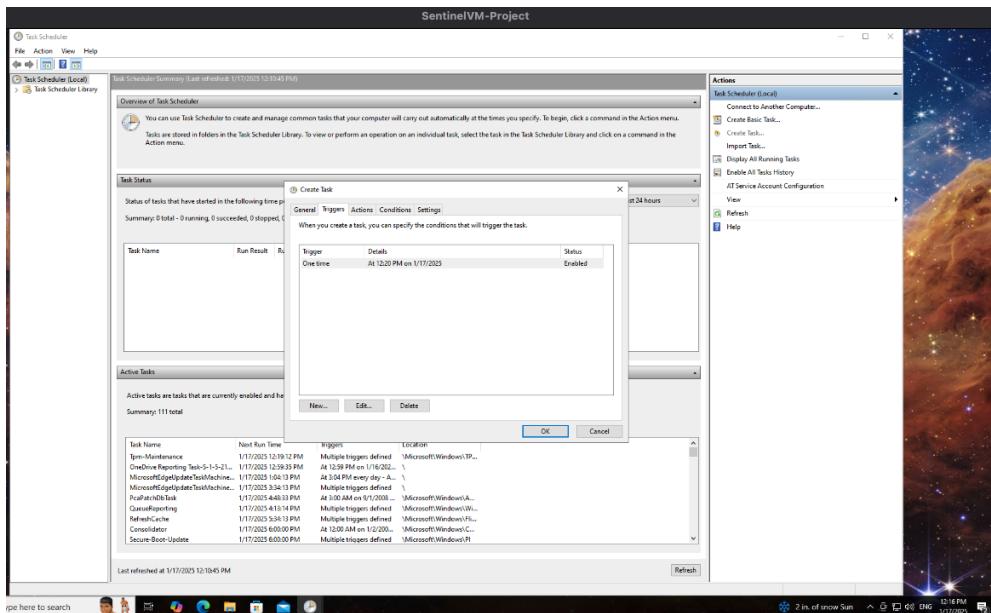
Creating our Scheduled Task

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

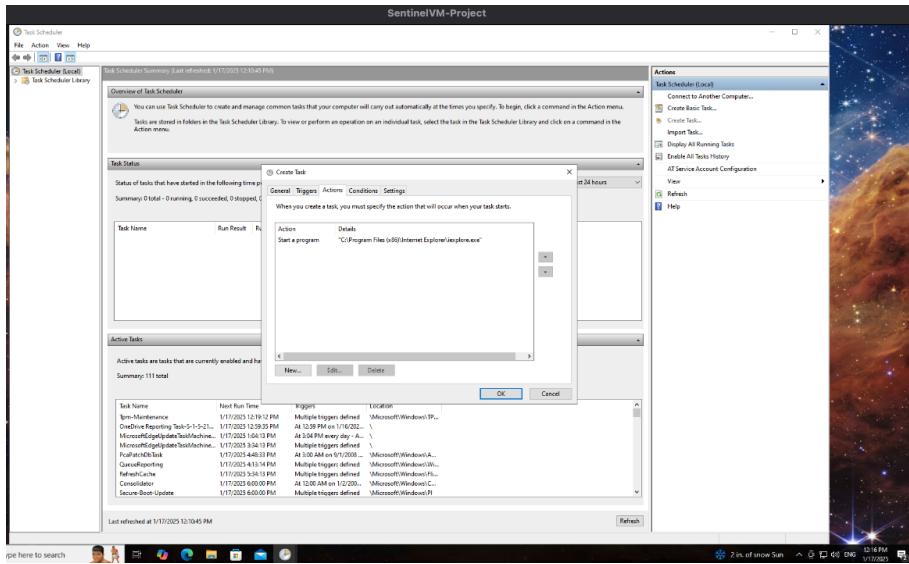
1. Next, we need to give it a name and change the 'Configure for' operating system to Windows 10.



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



- 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.

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

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

The screenshot shows the Microsoft Sentinel Analytics rule wizard interface. The current step is 'Edit existing Scheduled rule'. The 'Set rule logic' tab is active. In the 'Rule query' section, the KQL query is set to `SecurityEvent | where EventID == 4698`. To the right, there is a 'Results simulation' section showing a chart of raw event data over time.

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

The screenshot shows the Microsoft Sentinel Logs page. The 'Logs' item is selected in the left sidebar. A query editor on the right shows the KQL query `SecurityEvent | where EventID == 4698` with a time range of 'Last 24 hours' and a limit of 1000. The results table lists 10 events, all from the 'Microsoft-Windows-Security-Scheduled' source on the 'SentinelVM' computer at approximately 12:09:27.205 PM on January 17, 2025.

TimeGenerated [UTC]	Computer	EventSourceName
> 1/17/2025, 12:16:36.205 PM	SentinelVM	Microsoft-Windows-Security-Scheduled
> 1/17/2025, 12:16:36.205 PM	SentinelVM	Microsoft-Windows-Security-Scheduled
> 1/17/2025, 12:09:27.403 PM	SentinelVM	Microsoft-Windows-Security-Scheduled
> 1/17/2025, 12:09:27.403 PM	SentinelVM	Microsoft-Windows-Security-Scheduled
> 1/17/2025, 12:09:27.329 PM	SentinelVM	Microsoft-Windows-Security-Scheduled
> 1/17/2025, 12:09:27.329 PM	SentinelVM	Microsoft-Windows-Security-Scheduled
> 1/17/2025, 12:09:27.261 PM	SentinelVM	Microsoft-Windows-Security-Scheduled
> 1/17/2025, 12:09:27.261 PM	SentinelVM	Microsoft-Windows-Security-Scheduled
> 1/17/2025, 12:09:27.180 PM	SentinelVM	Microsoft-Windows-Security-Scheduled
> 1/17/2025, 12:09:27.180 PM	SentinelVM	Microsoft-Windows-Security-Scheduled

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.

The screenshot shows the Microsoft Sentinel Logs interface. A query has been run for the last 24 hours with a limit of 1000 results. The results pane displays raw XML event data for a single event (EventID 4698). The event details include fields like Computer (SentinelVM), TimeGenerated (2025-01-17T12:16:36.205Z), ClientProcessID (9596), and NameofSceuduledTask ('\\Simulated-Malicious-Task').

```

<Event>
<EventID>4698</EventID>
<TimeGenerated>2025-01-17T12:16:36.205Z</TimeGenerated>
<Computer>SentinelVM</Computer>
<ClientProcessID>9596</ClientProcessID>
<NameofSceuduledTask>\\Simulated-Malicious-Task</NameofSceuduledTask>
<TaskName>\\Simulated-Malicious-Task</TaskName>
<TaskContent><html version="1.0" encoding="UTF-16">&lt;Task version="1.4" xmlns="http://schemas.microsoft.com/windows/2004/02/mi/task"&gt;&lt;RegistrationInfo&gt;&lt;Date&gt;2025-01-17T12:16:36.1557349&lt;/Date&gt;&lt;Principal&gt;&lt;PrincipalName&gt;System&lt;/PrincipalName&gt;&lt;MultipleInstancesPolicy&gt;IgnoreNew&lt;/MultipleInstancesPolicy&gt;&lt;DisallowStartOnBatteries&gt;true&lt;/DisallowStartOnBatteries&gt;&lt;StopIfGoingOnBatteries&gt;true&lt;/StopIfGoingOnBatteries&gt;&lt;StopIfGoingOnBatteries&gt;true&lt;/StopIfGoingOnBatteries&gt;&lt;IgnoreIdleSessionAppSession&gt;false&lt;/IgnoreIdleSessionAppSession&gt;&lt;UseUnifiedSchedulingEngine&gt;false&lt;/UseUnifiedSchedulingEngine&gt;&lt;WakeToRun&gt;false&lt;/WakeToRun&gt;&lt;ClientProcessStartKey>53322479039656</ClientProcessStartKey>&lt;ClientProcessID>9596</ClientProcessID>&lt;RpcCallClientLocality>0</RpcCallClientLocality>&lt;FQDN>SentinelVM</FQDN>&lt;/TaskContent>
<EventData>
    ...
</EventData>

```

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

The screenshot shows the Microsoft Sentinel Logs interface with a different query. This query uses parsing commands to extract specific fields from the event data. The results are displayed as a table with columns: Computer, TimeGenerated [UTC], ClientProcessID, and NameofSceuduledTask. The table shows the same event data as the previous screenshot, but in a more structured and readable format.

```

1 SecurityEvent
2 | where EventID == 4698
3 | parseEventData with * '<Data Name="SubjectUserName">' User '</Data>' *
4 | parseEventData with * '<Data Name="TaskName">' NameofSceuduledTask '</Data>' *
5 | parseEventData with * '<Data Name="ClientProcessID">' ClientProcessID '</Data>' *
6 | project Computer, TimeGenerated, ClientProcessID, NameofSceuduledTask, User
7

```

Computer	TimeGenerated [UTC]	ClientProcessID	NameofSceuduledTask
SentinelVM	1/17/2025, 12:16:36.205 PM	9596	\\Simulated-Malicious-Task
			Computer: SentinelVM
			TimeGenerated [UTC]: 2025-01-17T12:16:36.205175Z
			ClientProcessID: 9596
			NameofSceuduledTask: \\Simulated-Malicious-Task
			User: shanks123
> SentinelVM	1/17/2025, 12:16:36.205 PM	9596	\\Simulated-Malicious-Task
> SentinelVM	1/17/2025, 12:09:27.403 PM	8028	\\Microsoft\\Windows\\Wi
> SentinelVM	1/17/2025, 12:09:27.403 PM	8028	\\Microsoft\\Windows\\Wi
> SentinelVM	1/17/2025, 12:09:27.329 PM	8028	\\Microsoft\\Windows\\Wi

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.

The screenshot shows the 'Analytics rule wizard - Edit existing Scheduled rule' interface. At the top, there are tabs for General, Set rule logic, Incident settings, Automated response, and Review + create. The 'Set rule logic' tab is selected. Below it, a section titled 'Rule query' contains a code editor with the following query:

```

SecurityEvent
| where EventID == 4698
| parseEventData with * '<Data Name="SubjectUserName">' User '</Data>' *
| parseEventData with * '<Data Name="TaskName">' NameofScheduledTask '</Data>' *
| parseEventData with * '<Data Name="ClientProcessID">' ClientProcessID '</Data>' *
| project Computer, TimeGenerated, ClientProcessID, NameofScheduledTask, User

```

Below the code editor are sections for 'Alert enhancement' (Entity mapping, Custom details, Alert details) and 'Query scheduling' (Run query every 5 hours). On the right, a 'Results simulation' chart is shown with the instruction: 'Define a valid analytics rule configuration and click 'Test with current data' to test your rule with current data in your workspace.' A button labeled 'Test with current data' is also present.

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 to 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.

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.