Contents

1 SC-200: Microsoft Security Operations Analyst

This repository includes lab instructions for the following courses:

- SC-200T00: Microsoft Security Operations Analyst

Download Latest Student Handbook and AllFiles Content

Are you a MCT? - Have a look at our GitHub User Guide for MCTs

Need to manually build the lab instructions? - Instructions are available in the MicrosoftLearning/Docker-Build repository

1.1 What are we doing?

- To support this course, we will need to make frequent updates to the course content to keep it current with the Azure and Microsoft 365 services used in the course. We are publishing the lab instructions and lab files on GitHub to allow for open contributions between the course authors and MCTs to keep the content current with changes in the Azure platform.
- We hope that this brings a sense of collaboration to the labs like we've never had before - when Azure changes and you find it first during a live delivery, go ahead and make an enhancement right in the lab source. Help your fellow MCTs.

1.2 How should I use these files relative to the released MOC files?

- The instructor handbook and PowerPoints are still going to be your primary source for teaching the course content.
- These files on GitHub are designed to be used in conjunction with the student handbook, but are in GitHub as a central repository so MCTs and course authors can have a shared source for the latest lab files.
- It will be recommended that for every delivery, trainers check GitHub for any changes that may have been made to support the latest Azure or Microsoft 365 services, and get the latest files for their delivery.

1.3 What about changes to the student handbook?

- We will review the student handbook on a quarterly basis and update through the normal MOC release channels as needed.

1.4 How do I contribute?

- Any MCT can submit a pull request to the code or content in the GitHub repro, Microsoft and the course author will triage and include content and lab code changes as needed.
- You can submit bugs, changes, improvement and ideas. Find a new Azure feature before we have? Submit a new demo!

1.5 Notes

1.5.1 Classroom Materials

1.6 It is strongly recommended that MCTs and Partners access these materials and in turn, provide them separately to students. Pointing students directly to GitHub to access Lab steps as part of an ongoing class will require them to access yet another UI as part of the course, contributing to a confusing experience for the student. An explanation to the student regarding why they are receiving separate Lab instructions can highlight the nature of an always-changing cloud-based interface and platform. Microsoft Learning support for accessing files on GitHub and support for navigation of the GitHub site is limited to MCTs teaching this course only.

1.7 title: Online Hosted Instructions permalink: index.html layout: home

2 Content Directory

Hyperlinks to each of the lab exercises and demos are listed below.

2.1 Labs

{% assign labs = site.pages | where_exp:"page", "page.url contains '/Instructions/Labs'" %} | Module | Lab | | --- | --- | {% for activity in labs %}| {{ activity.lab.module }} | [{{ activity.lab.title }}{% if activity.lab.type %} - {{ activity.lab.type }}{% endif %}](/home/ll/Azure_clone/Azure_new/SC-200T00A-Microsoft-Security-Operations-Analyst/{{ site.github.url }}{{ activity.url }}) | {% endfor %}

2.2 Demos

2.3 {% assign demos = site.pages | where_exp:"page", "page.url contains '/Instructions/Demos'" %} | Module | Demo | | --- | --- | {% for activity in demos %}| {{ activity.demo.module }} | [{{ activity.demo.title }}](/home/ll/Azure_clone/Azure_new/SC-200T00A-Microsoft-Security-Operations-Analyst/{{ site.github.url }}{{ activity.url }}) | {% endfor %}

2.4 demo: title: 'Demo: Deploying an ARM Template' module: 'Module 1: Exploring Azure Resource Manager'

3 Demo: Deploying an ARM Template

3.1 Instructions

1. Quisque dictum convallis metus, vitae vestibulum turpis dapibus non.
    1. Suspendisse commodo tempor convallis.
    2. Nunc eget quam facilisis, imperdiet felis ut, blandit nibh.
    3. Phasellus pulvinar ornare sem, ut imperdiet justo volutpat et.
2. Class aptent taciti sociosqu ad litora torquent per conubia nostra, per inceptos himenaeos.
3. Vestibulum hendrerit orci urna, non aliquet eros eleifend vitae.
4. Curabitur nibh dui, vestibulum cursus neque commodo, aliquet accumsan risus.

    ```
    Sed at malesuada orci, eu volutpat ex
    ```

5. In ac odio vulputate, faucibus lorem at, sagittis felis.
6. Fusce tincidunt sapien nec dolor congue facilisis lacinia quis urna.

    Note: Ut feugiat est id ultrices gravida.

7. Phasellus urna lacus, luctus at suscipit vitae, maximus ac nisl.
    - Morbi in tortor finibus, tempus dolor a, cursus lorem.
    - Maecenas id risus pharetra, viverra elit quis, lacinia odio.
    - Etiam rutrum pretium enim.
8. Curabitur in pretium urna, nec ullamcorper diam.

4 Module 1 - Lab 1 - Exercise 1 - Deploy Microsoft Defender for Endpoint

4.1 Lab scenario

You are a Security Operations Analyst working at a company that is implementing Microsoft Defender for Endpoint. Your manager plans to onboard a few devices to provide insight into required changes to the SecOps team response procedures.

You start by initializing the Defender for Endpoint environment. Next, you onboard the initial devices for your deployment by running the onboarding script on the devices. You configure security for the environment. Lastly, you create Device groups and assign the appropriate devices.

4.1.1 Task 1 - Obtain Your Microsoft 365 Credentials

Once you launch the lab, a free trial tenant will be made available to you to access in the Microsoft Virtual Lab environment. This tenant will be automatically assigned a unique username and password. You must retrieve this username and password so that you can sign in to Azure and Microsoft 365 within the Microsoft Virtual Lab environment.

Because this course can be offered by learning partners using any one of several authorized lab hosting providers, the actual steps involved to retrieve the tenant ID associated with your tenant may vary by lab hosting provider. Therefore, your instructor will provide you with the necessary instructions on how to retrieve this information for your course. The information that you should note for later use includes:

```
- **Tenant suffix ID.** This ID is for the onmicrosoft.com accounts that you will use to sign in to Mic
- **Tenant password.** This is the password for the admin account provided by your lab hosting provider
```

4.1.2 Task 2: Initialize Microsoft Defender for Endpoint.

In this task, you will perform the initialization of the Microsoft Defender for Endpoint portal.

1. Log in to WIN1 virtual machine as Admin with the password: **Pa55w.rd**.
2. Open the Microsoft Edge browser, search for "edge browser update", download, and install the new Microsoft Edge browser. This is necessary to ensure you're running the latest version of Microsoft Edge in your hosted virtual machine. Start the new Edge browser.
3. In the Edge browser, go to the Microsoft Defender Security Center at (https://securitycenter.microsoft.com).
4. In the **Sign in** dialog box, copy and paste in the tenant Email account for the admin username provided by your lab hosting provider and then select **Next**.
5. In the **Enter password** dialog box, copy and paste in the admin's tenant password provided by your lab hosting provider and then select **Sign in**.

Note: if you receive a message "You can't access this section.", wait 5 minutes and try again. Sometimes the access rules need to propagate the tenant.

6. On the **Microsoft Security Center** setup Step 2, select **Next**.
7. On Step 3 **Set up preferences**, select the data storage location appropriate for where this training tenant is being managed. You may want to validate with your course instructor.
8. Confirm the Preview features are **On**.
9. Select **Next**.
10. Select Continue on the **Create your cloud instance**
11. After the **Creating your Microsoft Defender for Endpoint account** progress bar completes. Step 4 options will be displayed. Select **Start using Microsoft Defender for Endpoint**.
12. In the Setup incomplete dialog box select **Proceed anyway**.

Note: The setup is **Complete**. You will onboard Devices in the next task.

4.1.3 Task 3: Onboard a Device.

In this task, you will onboard a device to Microsoft Defender for Endpoint.

1. Go to the Microsoft Defender Security Center at (https://securitycenter.microsoft.com) and login with the **Tenant Email** credentials if you are not currently in the portal.
2. Select **Settings** from the left menu bar.
3. Select **Onboarding** in the Device management section.
4. In the Onboard a device area select **Download Package** button.
5. Extract the downloaded zip file to a local folder like the Documents folder.
6. Right-click on the extracted file WindowsDefenderATPLocalOnboardingScript.cmd and choose **Run as Administrator**. If you encounter the Windows SmartScreen choose to Run anyway.

Note By default, the file should be in the c:\users\admin\downloads directory.

7. Answer **Y** to questions presented by the script. When complete you should see a message in the command screen that says something like "Successfully onboarded machine..."
8. From the Onboarding page in the portal, copy the detection test script and run it in an open command window. You may have to open a new **Administrator: Command Prompt** window by typing *CMD* in the windows search bar and choose to **run as Administrator**.
9. In the Microsoft Defender Security Center portal menu, select **Device inventory**. You should now see your device in the list.

Note It can take up to 5 minutes for the device to be displayed in the portal.

4.1.4 Task 4: Configure Role

In this task, you will configure roles for use with device groups.

1. In the Microsoft Defender Security Center portal select **Settings** from the left menu bar.
2. Select **Roles** in the permissions area.
3. Select the **Turn on roles** button.
4. Select **Add item**.
5. In the Add Role dialog enter the following: Role Name: Tier Live Response capabilities: select checkbox Advanced: select.
6. Select **Next**.
7. In the Assigned user groups tab. Select **sg-IT** and then select **Add selected groups**.
8. Select **Save**.

4.1.5 Task 5: Configure Device Groups

In this task, you will configure device groups that allow for access control and automation configuration.

1. Select **Settings** from the left menu bar.
2. In the permissions area select **Device groups**.
3. Select **Add device group**.
4. Enter the following information on the General tab:

- Device group name: Regular
- Automation level: Full - remediate threats automatically
- Members: Name equals TESTLAB

5. Select **Next**.
6. For the User access tab, select **sg-IT** and then select **Add selected groups**
7. Select **Done**.
8. Device group configuration has changed. Apply changes to check matches and recalculate groupings.

4.2 Proceed to Exercise 2

5 Module 1 - Lab 1 - Exercise 2 - Mitigate Attacks with Microsoft Defender for Endpoint

5.1 Lab scenario

You are a Security Operations Analyst working at a company that is implementing Microsoft Defender for Endpoint. Your manager plans to onboard a few devices to provide insight into required changes to the SecOps team response procedures.

To explore the Defender for Endpoint attack mitigation capabilities, you run six simulated attacks.

5.1.1 Task 1: Simulated Attacks

In this task, you will run six simulated attacks to explore the capabilities of Microsoft Defender for Endpoint.

1. If you are not already at the Microsoft Defender Security Center in your browser, go to the Microsoft Defender Security Center at (https://securitycenter.microsoft.com) logged in as Admin for your tenant.
2. From the menu, select **Evaluation and tutorials** and then **Simulations and tutorials** from the left side.
3. Complete Scenario 1, Scenario 2, Scenario 3, Scenario 4, Scenario 5, and Scenario 7. Follow the instructions in the provided walkthrough in the portal.

Warning Do not perform Scenario 6.

5.2 You have completed the lab.

6 Module 2 - Lab 1 - Exercise 1 - Explore Microsoft 365 Defender

6.1 Lab scenario

You are a Security Operations Analyst working at a company that is implementing Microsoft 365 Defender. You start by exploring the features of the Microsoft 365 security portal.

6.1.1 Task 1: Explore the Microsoft 365 security portal

In this task, you will explore the options in the Microsoft 365 security portal.

1. Login to WIN1 virtual machine as Admin with the password: **Pa55w.rd**.
2. Open the browser, search for, download, and install the new Microsoft Edge browser if you didn't already do this in the previous lab. Start the new Edge browser.
3. In the Edge browser, go to the Microsoft Defender Security Center at (https://security.microsoft.com).
4. In the **Sign in** dialog box, copy and paste in the tenant Email account for the admin username provided by your lab hosting provider and then select **Next**.
5. In the **Enter password** dialog box, copy and paste in the admin's tenant password provided by your lab hosting provider and then select **Sign in**.
6. Explore the menu options in the portal.

6.2 You have completed the lab.

7 Module 3 - Lab 1 - Exercise 1 - Enable Azure Defender

7.1 Lab scenario

You're a Security Operations Analyst working at a company that is implementing cloud workload protection with Azure Defender. In this lab you will enable Azure Defender.

7.1.1 Task 1: Access the Azure portal and set up a Subscription.

In this task, you will set up an Azure Subscription required to complete this lab and future labs.

1. Log in to WIN1 virtual machine as Admin with the password: **Pa55w.rd**.
2. Open the browser, search for, download, and install the new Microsoft Edge browser if you didn't do this in the previous labs. Start the new Edge browser.
3. In the Edge browser, navigate to the Azure portal at https://portal.azure.com.
4. In the **Sign in** dialog box, copy and paste in the tenant Email account for the admin username provided by your lab hosting provider and then select **Next**.
5. In the **Enter password** dialog box, copy and paste in the admin's tenant password provided by your lab hosting provider and then select **Sign in**.
6. In the Search bar of the Azure portal, type *Subscription*, then select **Subscriptions**.

Important: You must create the Azure Subscription as the Admin user for the tenant.

7. Select **Add** for a new Subscription.
8. Follow the page instructions to create a new subscription that is appropriate for you. Most people will choose "Free Trial".

Note: It could take up to 30 minutes before the subscription can be used.

7.1.2 Task 2: Create a Log Analytics Workspace.

In this task, you will create a Log Analytics workspace for use with Azure Defender.

1. In the Search bar of the Azure portal, type *Log Analytics*, then select **Log Analytics workspaces**.
2. Select **+Create** from the command bar.
3. Select **Create new** for the Resource group.
4. Enter *rg-AzureDefender*.
5. For the Name, enter something unique like: *uniquename_AzureDefender*
6. Select **Review + Create**.
7. Once the workspace validation has passed, select **Create**.

Note: Wait for the new workspace to be provisioned, this may take a few minutes.

7.1.3 Task 3: Enable Azure Defender.

In this task, you will enable and configure Azure Defender.

1. In the Search bar of the Azure portal, type *Security*, then select **Security Center**.
2. On the **Getting started** page of Security Center go to the **Upgrade** section and make sure your subscription is selected, and then select **Upgrade** button.
3. The next page shows the option to install the agent on virtual machines already in the subscription. Do nothing here.
4. Select **Pricing & settings** from the Management area of the portal menu.
5. Select your Subscription.
6. Review the resources and fees. Turn Servers **Off** then select **Save**. Confirm if prompted.

Note: This is for lab purposes only. It is good to understand which resources will be automatically covered and the fees involved. The next steps are to disable Azure Defender for Servers. The purpose of this is to manage the cost in your Azure subscription. Normally, you would leave this enabled.

7. Select **Auto provisioning** from the Settings area.
8. Review the Auto provisioning - Extensions. Confirm that **Log Analytics agent for Azure VMs** is **Off**.
9. Go back to the Security Center portal and select the **Pricing and settings** again.
10. Select the workspace ID you created earlier **uniquename_AzureDefender**
11. Turn Server **Off**, then select **Save** if the Servers plan is not already off.

7.1.4 Task 4: Install Azure Arc on an On-Premises Server.

To make onboarding of your on-premises server easier. Install Azure Arc, which will then enable Azure to manage the on-premises server.

In this task, you will install Azure Arc on an on-premises server.

1. Log in to WINServer virtual machine as Administrator with the password: **Passw0rd!**.
2. In the Edge browser, navigate to the Azure portal at https://portal.azure.com.
3. In the **Sign in** dialog box, copy and paste in the **Tenant Email** account provided by your lab hosting provider and then select **Next**.

4. In the **Enter password** dialog box, copy and paste in the **Tenant Password** provided by your lab hosting provider and then select **Sign in**.
5. In the Search bar of the Azure portal, type *Azure Arc*, then select **Azure Arc**.
6. Select **Servers** from the Azure Arc page menu.
7. Select **+ Add**.
8. Select **Generate script** in the Add a single server section.
9. Select **Next : Resource details >**.
10. Select the Resource group you created earlier. Hint: *rg-AzureDefender*

Note If you haven't already created a resource group. Open another tab and create the resource group.

11. Select **Next: Tags >**.
12. Select **Next: Download and run script >**.
13. Select **Register**.

Note Wait three minutes for processing.

14. Select the **Download** button.
15. Open Windows PowerShell and select **Run as Administrator**.
16. Enter Administrator for the User name if prompted.
17. Enter Passw0rd! for the password if prompted.
18. Enter: cd Downloads The screen should show: PS C:\Users\Administrators\Downloads
19. Type *Set-ExecutionPolicy -ExecutionPolicy Unrestricted* and press enter.
20. Enter **A** for Yes to All and press enter.
21. Type *.\OnboardingScript.ps1* and press enter.
22. Select **R** to Run once and press enter (this may take a minute).
23. Follow the on-screen instructions to complete the device registration. This will include authentication of the device.
24. On the Azure Arc portal page, select **Servers**.
25. Select **Refresh** until your server name appears.

Note This could take a few minutes.

7.1.5 Task 5: Protect an On-Premise Server.

In this task, you will manually install the required agent on the Windows Server.

1. Go to the Azure Security Center and select the **Getting Started** page.
2. Select the **Get Started** tab.
3. Select **Configure** under the Add non-Azure servers section.
4. Select **Upgrade** next to the workspace you created earlier. Hint: uniqueAzureDefender
5. Select **+ Add Servers** next to the workspace you created earlier.
6. Select **Download Windows Agent (64 bit)**.
7. Run the downloaded file.
8. Select **next** until the wizard page for Agent Setup Options appears, Select **Connect the Agent to Log Analytics (OMS)**, then select **Next**.
9. Copy and paste the Workspace ID and Primary Key from the Azure portal into the wizard page fields as appropriate and select **Next**.
10. Continue with the Install. Then select **Finish** when complete.

11. Go to the Security Center portal and select **Inventory**.
12. The Server should appear in the list. You may have to select **Refresh** to see the update and it may take a couple minutes.

Note The Server should appear as unprotected. This is correct as we turned off the Azure Defender plans for Servers.

8 Proceed to Exercise 2

9 Module 3 - Lab 1 - Exercise 2 - Mitigate threats using Azure Defender

9.1 Lab scenario

You're a Security Operations Analyst working at a company that implemented Azure Defender. You need to respond to security alerts generated by Azure Defender.

9.1.1 Task 1: Mitigate security alerts

In this task, you will load sample security alerts and review the alert details.

1. Log in to WIN1 virtual machine as Admin with the password: **Pa55w.rd**.
2. In the Edge browser, open the Azure portal at https://portal.azure.com.
3. In the **Sign in** dialog box, copy and paste in the **Tenant Email** account provided by your lab hosting provider and then select **Next**.
4. In the **Enter password** dialog box, copy and paste in the **Tenant Password** provided by your lab hosting provider and then select **Sign in**.
5. In the Search bar of the Azure portal, type *Security*, then select **Security Center**.
6. Select **Security alerts** in the portal menu.
7. Select **Sample Alerts** from the command bar.

Note Wait for the sample alerts to load.

8. In the Create sample alerts (Preview) pane make sure your subscription is selected. Make sure all sample alerts are selected and select **Create sample alerts**.

NoteThis may take a few minutes to complete.

9. For each of the alerts listed. Perform the following actions:

   A. Select the alert, information about the alert should appear. Select **View full details**.

   B. Review the Alert details.

   C. Select the **Take action** tab.

   D. Review the Take action information. Notice the buttons available to take action depending on the type of alert.

9.2 You have completed the lab.

10 Module 4 - Lab 1 - Exercise 1 - Create queries for Azure Sentinel using Kusto Query Language (KQL)

10.1 Lab scenario

You are a Security Operations Analyst working at a company that is implementing Azure Sentinel. You are responsible for performing log data analysis to search for malicious activity, display visualizations, and perform threat hunting. To query log data, you use the Kusto Query Language (KQL).

10.1.1 Task 1: Access the KQL testing area.

In this task, you will access a Log Analytics environment where you can practice writing KQL statements.

1. Login to WIN1 virtual machine as Admin with the password: **Pa55w.rd**.
2. Go to https://aka.ms/lademo in your browser. Login with the MOD Administrator credentials.
3. Explore the available tables listed in the tab on the left side of the screen.
4. In the query editor, enter the following query and select the Run button. You should see the query results in the bottom window.

```
SecurityEvent
```

5. Next to the first record, select the > to expand the information for the row.

10.1.2 Task 2: Run Basic KQL Statements

In this task, you will build basic KQL statements.

1. The following statement demonstrates the use of the let statement to declare variables. In the Query Window. Enter the following statement and select **run**:

```
let timeOffset = 7d;
let discardEventId = 4688;
SecurityEvent
| where TimeGenerated > ago(timeOffset*2) and TimeGenerated < ago(timeOffset)
| where EventID != discardEventId
```

2. The following statement demonstrates the use of the let statement to declare a dynamic list. In the Query Window enter the following statement and select **run**:

```
let suspiciousAccounts = datatable(account: string) [
    @"\administrator",
    @"NT AUTHORITY\SYSTEM"
];
SecurityEvent | where Account in (suspiciousAccounts)
```

3. The following statement demonstrates the use of the let statement to declare a dynamic table. In the Query Window. Enter the following statement and select **run**:

```
let LowActivityAccounts =
    SecurityEvent
    | summarize cnt = count() by Account
    | where cnt < 10;
LowActivityAccounts | where Account contains "Mal"
```

Note: When you run this script you should get no results.

4. The following statement demonstrates searching across all tables and columns for records within the query time range display in the query window. In the Query Window before running this script change the Time range to "Last hour". Enter the following statement and select **run**:

```
search "err"
```

Warning: Make sure you change back the Time range to "Last 24 hours" for the next scripts.

5. The following statement demonstrates searching across tables listed with the "in" clause for records within the query time range display in the query window. In the Query Window. Enter the following statement and select **run**:

```
search in (SecurityEvent,SecurityAlert,A*) "err"
```

6. The following statements demonstrates filter using the where operator. In the Query Window. Enter the following statement and select **run**:

Note: You should "run" after entering the query from each code block below.

```
SecurityEvent
| where TimeGenerated > ago(1d)
```

```
SecurityEvent
| where TimeGenerated > ago(1h) and EventID == "4624"
```

```
SecurityEvent
| where TimeGenerated > ago(1h)
| where EventID == 4624
| where AccountType =~ "user"
```

```
SecurityEvent | where EventID in (4624, 4625)
```

7. The following statement demonstrates creating fields using the extend operator In the Query Window. Enter the following statement and select **run**:

```
SecurityAlert
| where TimeGenerated > ago(7d)
| extend severityOrder = case (
    AlertSeverity == "High", 3,
    AlertSeverity == "Medium", 2,
    AlertSeverity == "Low", 1,
    AlertSeverity == "Informational", 0,
    -1)
```

8. The following statement demonstrates a real-world example that combines the let, dynamics list creation, and creating fields using extend. In the Query Window. Enter the following statement and select run:

```
let timeframe = 1d;
let DomainList = dynamic(["tor2web.org", "tor2web.com"]);
Syslog
| where TimeGenerated >= ago(timeframe)
| where ProcessName contains "squid"
| extend
  HTTP_Status_Code = extract("(TCP_(([A-Z]+)…-9]{3}))",8,SyslogMessage),
  Domain = extract("(([A-Z]+ [a-z]{4…Z]+ )([^ :\\/]*))",3,SyslogMessage)
| where HTTP_Status_Code == "200"
| where Domain contains "."
| where Domain has_any (DomainList)
```

Note: When you run this script you should get no results.

9. The following statement demonstrates sorting results using the order by operator. In the Query Window. Enter the following statement and select **run**:

```
SecurityAlert
| where TimeGenerated > ago(7d)
| extend severityOrder = case (
    AlertSeverity == "High", 3,
    AlertSeverity == "Medium", 2,
    AlertSeverity == "Low", 1,
    AlertSeverity == "Informational", 0,
    -1)
| order by severityOrder desc
```

10. The following statements demonstrates specifying fields for the result set using the project operators.

Note: You should ”run” after entering the query from each code block below.

In the Query Window. Enter the following statement and select **run**:

```
SecurityEvent
| project Computer, Account
```

```
SecurityAlert
| where TimeGenerated > ago(7d)
| extend severityOrder = case (
    AlertSeverity == "High", 3,
    AlertSeverity == "Medium", 2,
    AlertSeverity == "Low", 1,
    AlertSeverity == "Informational", 0,
```

```
    -1)
| order by severityOrder
| project-away severityOrder
```

10.1.3 Task 3: Analyze Results in KQL with the Summarize Operator

In this task, you will build KQL statements to prepare data.

1. The following statement demonstrates the count function. In the Query Window. Enter the following statement and select **run**:

```
SecurityEvent
| where EventID == "4688"
| summarize count() by Process, Computer
```

2. The following statement demonstrates the count function. In the Query Window. Enter the following statement and select **run**:

```
SecurityEvent
| where TimeGenerated > ago(1h)
| where EventID == 4624
| summarize cnt=count() by AccountType, Computer
```

3. The following statement demonstrates the dcount function. In the Query Window. Enter the following statement and select **run**:

```
SecurityEvent
| summarize dcount(IpAddress)
```

4. The following statement is an Azure Sentinel Analytical rule to detect a password spray attempt.

The first three where operators will filter the result set to failed logins to disabled accounts. Next, the statement "summarize" a distinct count of application name and group by User and IP Address. Finally, there is a check against a variable created (threshold) to see if the number exceeds the allowed amount. In the Query Window. Enter the following statement and select **run**:

```
let timeframe = 1d;
let threshold = 3;
SigninLogs
| where TimeGenerated >= ago(timeframe)
| where ResultType == "50057"
| where ResultDescription =~ "User account is disabled. The account has been disabled by an administrat
| summarize applicationCount = dcount(AppDisplayName) by UserPrincipalName, IPAddress
| where applicationCount >= threshold
```

Note: When you run this script you should get no results.

5. The following statement demonstrates the arg_max function.

The following statement will return the most current row from the SecurityEvent table for the computer SQL12.NA.contosohotels.com. The * in the arg_max function requests all columns for the row. In the Query Window. Enter the following statement and select **run**:

```
SecurityEvent
| where Computer == "SQL12.na.contosohotels.com"
| summarize arg_max(TimeGenerated,*) by Computer
```

6. The following statement demonstrates the arg_min function.

In this statement, the oldest SecurityEvent for the computer SQL12.NA.contosohotels.com will be returned as the result set. In the Query Window. Enter the following statement and select **run**:

```
SecurityEvent
| where Computer == "SQL12.na.contosohotels.com"
| summarize arg_min(TimeGenerated,*) by Computer
```

7. The following statements demonstrate the importance of understanding results based on the order of the pipe "|". In the Query Window. Enter the following statements and run each separately:

Statement 1

```
SecurityEvent
| summarize arg_max(TimeGenerated, *) by Account
| where EventID == "4624"
```

Statement 2

```
SecurityEvent
| where EventID == "4624"
| summarize arg_max(TimeGenerated, *) by Account
```

Statement 1 will have Accounts for which the last activity was a login.

The SecurityEvent table will first be summarized and return the most current row for each Account. Then only rows with EventID equals 4624 (login) will be returned.

Statement 2 will have the most recent login for Accounts that have logged in.

The SecurityEvent table will be filtered to only include EventID = 4624. Then these results will be summarized for the most current login row by Account.

8. The following statement demonstrates the make_list function.

The function returns a dynamic (JSON) array of all the values of Expression in the group. This KQL query will first filter the EventID with the where operator. Next, for each Computer, the results are a JSON array of Accounts. The resulting JSON array will include duplicate accounts.

In the Query Window. Enter the following statement and select run:

```
SecurityEvent
| where EventID == "4624"
| summarize make_list(Account) by Computer
```

9. The following statement demonstrates the make_list function.

make_list returns a dynamic (JSON) array containing distinct values that Expression takes in the group. This KQL query will first filter the EventID with the where operator. Next, for each Computer, the results are a JSON array of unique Accounts. In the Query Window. Enter the following statement and select **run**:

```
SecurityEvent
| where EventID == "4624"
| summarize make_set(Account) by Computer
```

10.1.4 Task 4: Create visualizations in KQL with the Render Operator

In this task, you will use generate visualizations with KQL statements.

1. The following statement demonstrates the render function visualizing results with a barchart. In the Query Window. Enter the following statement and select **run**:

```
SecurityEvent
| summarize count() by Account
| render barchart
```

2. The following statement demonstrates the render function visualizing results with a time series.

The bin() function rounds values down to an integer multiple of the given bin size. Used frequently in combination with summarize by .... If you have a scattered set of values, the values are grouped into a smaller set of specific values. Combining the generated time series and pipe to a render operator with a type of timechart provides a time series visualization. In the Query Window. Enter the following statement and select **run**:

```
SecurityEvent
| summarize count() by bin(TimeGenerated, 1d)
| render timechart
```

10.1.5 Task 5: Build multi-table statements in KQL

In this task, you will build multi-table KQL statements.

1. The following statement demonstrates the union operator that takes two or more tables and returns the rows of all of them. Understanding how results are passed and impacted with the pipe character is essential. Based on the time window set in the Query window:

Query 1 will return all rows of SecurityEvent and all rows of SecurityAlert

Query 2 will return one row and column, which is the count of all rows of SecurityEvent and all rows of SecurityAlert

Query 3 will return all rows of SecurityEvent and one row for SecurityAlert. The row for SecurityAlert will have the count of the SecurityAlert rows.

Run each Query separately to see the results.

In the Query Window. Enter the following statements and select **run** for each:

Query 1

```
SecurityEvent
| union SecurityAlert
```

Query 2

```
SecurityEvent
| union SecurityAlert
| summarize count()
| project count_
```

Query 3

```
SecurityEvent
| union (SecurityAlert  | summarize count())
| project count_
```

2. The following statement demonstrates the union operator support for wildcards to union multiple tables. In the Query Window. Enter the following statement and select **run**:

```
union Security*
| summarize count() by Type
```

3. The following statement demonstrates the join operator, which merges the rows of two tables to form a new table by matching the specified columns' values from each table. In the Query Window. Enter the following statement and select **run**:

```
SecurityEvent
| where EventID == "4624"
| summarize LogOnCount=count() by EventID, Account
| project LogOnCount, Account
| join kind = inner (
     SecurityEvent
     | where EventID == "4634"
     | summarize LogOffCount=count() by EventID, Account
     | project LogOffCount, Account
) on Account
```

The first table specified in the join is considered the Left table. The table after the join keyword is the right table. When working with columns from the tables, the $left.Column name and $right.Column name is to distinguish which tables column are referenced.

10.1.6 Task 6: Work with string data in KQL

In this task, you will work with structured and unstructured string fields with KQL statements.

1. The following statement demonstrates the extract function. Extract gets a match for a regular expression from a text string. You have the option to convert the extracted substring to the indicated type. In the Query Window. Enter the following statement and select **run**:

```
print extract("x=([0-9.]+)", 1, "hello x=45.6|wo") == "45.6"
```

2. The following statements use the extract function to pull out the Account Name from the Account field of the SecurityEvent table. In the Query Window. Enter the following statement and select **run**:

```
let top5 = SecurityEvent
| where EventID == 4625 and AccountType == 'User'
```

```
| extend Account_Name = extract(@"^(.*\\)?([^@]*)(@.*)?$", 2, tolower(Account))
| summarize Attempts = count() by Account_Name
| where Account_Name != ""
| top 5 by Attempts
| summarize make_list(Account_Name);

SecurityEvent
| where EventID == 4625 and AccountType == 'User'
| extend Name = extract(@"^(.*\\)?([^@]*)(@.*)?$", 2, tolower(Account))
| extend Account_Name = iff(Name in (top5), Name, "Other")
| where Account_Name != ""
| summarize Attempts = count() by Account_Name
```

3. The following statement demonstrates the parse function. Parse evaluates a string expression and parses its value into one or more calculated columns. The computed columns will have nulls for unsuccessfully parsed strings.

Review the following statement, but do not run it:

```
let SQlData = Event
| where Source has "MSSQL"
;
let Sqlactivity = SQlData
| where RenderedDescription !has "LGIS" and RenderedDescription !has "LGIF"
| parse RenderedDescription with * "action_id:" Action:string
                                    " " *
| parse RenderedDescription with * "client_ip:" ClientIP:string
" permission" *
| parse RenderedDescription with * "session_server_principal_name:" CurrentUser:string
" " *
| parse RenderedDescription with * "database_name:" DatabaseName:string
"schema_name:" Temp:string
"object_name:" ObjectName:string
"statement:" Statement:string
"." *
;
let FailedLogon = SQlData
| where EventLevelName has "error"
| where RenderedDescription startswith "Login"
| parse kind=regex RenderedDescription with "Login" LogonResult:string
                                            "for user '" CurrentUser:string
                                            "'. Reason:" Reason:string
                                            "provided" *
| parse kind=regex RenderedDescription with * "CLIENT" * ":" ClientIP:string
                                            "]" *
;
let dbfailedLogon = SQlData
| where RenderedDescription has " Failed to open the explicitly specified database"
| parse kind=regex RenderedDescription with "Login" LogonResult:string
                                            "for user '" CurrentUser:string
                                            "'. Reason:" Reason:string
                                            " '" DatabaseName:string
                                            "'" *
| parse kind=regex RenderedDescription with * "CLIENT" * ":" ClientIP:string
                                            "]" *
;
let successLogon = SQlData
| where RenderedDescription has "LGIS"
| parse RenderedDescription with * "action_id:" Action:string
                                  " " LogonResult:string
                                  ":" Temp2:string
                                  "session_server_principal_name:" CurrentUser:string
```

```
                                    " " *
| parse RenderedDescription with * "client_ip:" ClientIP:string
                                    " " *
;
(union isfuzzy=true
Sqlactivity, FailedLogon, dbfailedLogon, successLogon )
| project TimeGenerated, Computer, EventID, Action, ClientIP, LogonResult, CurrentUser, Reason, Databas
```

4. The following statement demonstrates working with Dynamics Fields:

Within a Log Analytics table, there are field types defined as Dynamic. Dynamic fields contain a key-value pair such as: {"eventCategory":"Autoscale","eventName":"GetOperationStatusResult","operationId":"xxxxxxxx-6a53-4aed-bab4-575642a10226","eventProperties":"{"OldInstancesCount":6,"NewInstancesCount":5}","eventDataId":"xxxxxxxx -efe3-43c2-8c86-cd84f70039d3","eventSubmissionTimestamp":"2020-11-30T04:06:17.0503722Z","resource":"ch-appfevmss-pri","resourceGroup":"CH-RETAILRG-PRI","resourceProviderValue":"MICROSOFT.COMPUTE","subscriptio xxxxxxxx -7fde-4caf-8629-41dc15e3b352","activityStatusValue":"Succeeded"}

To access the strings within a Dynamic field, use the dot notation. The Properties_d field from the AzureActivity table is of type dynamic. In this example, you could access the eventCategory with the Properties_d.eventCategory field name.

In the Query Window. Enter the following statement and **run**:

```
AzureActivity
| project Properties_d.eventCategory
```

Note: When you run this script you should get no results.

Review the following statement only, do not run it:

```
SigninLogs
| where TimeGenerated >= ago(1d)
| extend OS = DeviceDetail.operatingSystem, Browser = DeviceDetail.browser
| extend ConditionalAccessPol0Name = tostring(ConditionalAccessPolicies[0].displayName), ConditionalAcc
| extend ConditionalAccessPol1Name = tostring(ConditionalAccessPolicies[1].displayName), ConditionalAcc
| extend ConditionalAccessPol2Name = tostring(ConditionalAccessPolicies[2].displayName), ConditionalAcc
| extend StatusCode = tostring(Status.errorCode), StatusDetails = tostring(Status.additionalDetails)
| extend State = tostring(LocationDetails.state), City = tostring(LocationDetails.city)
| extend Date = startofday(TimeGenerated), Hour = datetime_part("Hour", TimeGenerated)
| summarize count() by Date, Identity, UserDisplayName, UserPrincipalName, IPAddress, ResultType, Resul
| sort by Date
```

5. The following statement demonstrates functions to manipulate JSON stored in string fields. Many logs submit data in JSON format, which requires you to know how to transform JSON data to queryable fields.

In the Query Window. Enter the following statements individually and select **Run**:

```
SecurityAlert
| extend ExtendedProperties = todynamic(ExtendedProperties)
| extend ActionTaken = ExtendedProperties.ActionTaken
| extend AttackerIP = ExtendedProperties["Attacker IP"]
```

```
SecurityAlert
| mv-expand entity = todynamic(Entities)
```

```
SecurityAlert
| where TimeGenerated >= ago(7d)
| mv-apply entity = todynamic(Entities) on
( where entity.Type == "account" | extend account = strcat (entity.NTDomain, "\\", entity.Name))
```

6. Parsers are functions that define a virtual table with already parsed unstructured strings fields such as Syslog data. The following is a KQL query created by the community for Mailbox forwarding monitoring.

Review the following statement, but do not run it:

```
OfficeActivity
    | where TimeGenerated >= ago(30d)
    | where Operation == 'New-InboxRule'
    | extend details = parse_json(Parameters)
```

```
| where details contains 'ForwardTo' or details contains 'RedirectTo'
| extend ForwardTo = iif(details[0].Name contains 'ForwardTo', details[0].Value,
    iif(details[1].Name contains 'ForwardTo', details[1].Value,
        iif(details[2].Name contains 'ForwardTo', details[2].Value,
            iif(details[3].Name contains 'ForwardTo', details[3].Value,
                iif(details[4].Name contains 'ForwardTo', details[4].Value,
                    'Check Parameters')))))
| extend RedirectTo = iif(details[0].Name contains 'RedirectTo', details[0].Value,
    iif(details[1].Name contains 'RedirectTo', details[1].Value,
        iif(details[2].Name contains 'RedirectTo', details[2].Value,
            iif(details[3].Name contains 'RedirectTo', details[3].Value,
                iif(details[4].Name contains 'RedirectTo', details[4].Value,
                    'Check Parameters')))))
| extend RuleName = iif(details[3].Name contains 'Name', details[3].Value,
     iif(details[4].Name contains 'Name', details[4].Value,
        iif(details[5].Name contains 'Name', details[5].Value,
            'Check Parameters')))
| extend RuleParameters = iif(details[2].Name != 'ForwardTo' and  details[2].Name != 'RedirectTo',
    strcat(tostring(details[2].Name), '-', tostring(details[2].Value)),
    iif(details[3].Name != 'ForwardTo' and  details[3].Name != 'RedirectTo' and details[3].Name !=
        strcat(tostring(details[3].Name), '-', tostring(details[3].Value)),
            iff(details[4].Name != 'ForwardTo' and details[4].Name != 'RedirectTo' and details[4].N
            strcat(tostring(details[4].Name), '-', tostring(details[4].Value)),
            'All Mail')))
| project TimeGenerated, Operation, RuleName, RuleParameters, iif(details contains 'ForwardTo', For
| project-rename Email_Forwarded_To = Column1, Creating_User = UserId
```

To create a function:

After running the query, click the Save button, enter the Name: MailboxForward, and select Save As Function from the drop-down.

The function will be available in KQL by using the function alias:

Note: You will not be able to do this in the lademo environment used for data in this lab, but it's an important concept to be used in your environment.

```
MailboxForward
```

10.2 You have completed the lab.

11 Module 5 - Lab 1 - Exercise 1 - Configure your Azure Sentinel environment

11.1 Lab scenario

You're a Security Operations Analyst working at a company that is implementing Azure Sentinel. You're responsible for setting up the Azure Sentinel environment to meet the company requirement to minimize cost, meet compliance regulations, and provide the most manageable environment for your security team to perform their daily job responsibilities.

11.1.1 Task 1: Initialize the Azure Sentinel Workspace.

In this task, you will create an Azure Sentinel workspace.

1. Log in to WIN1 virtual machine as Admin with the password: **Pa55w.rd**.
2. Open the browser, search for, download, and install the new Microsoft Edge browser. Start the new Edge browser.
3. In the Edge browser, navigate to the Azure portal at https://portal.azure.com.
4. In the **Sign in** dialog box, copy and paste in the **Tenant Email** account provided by your lab hosting provider and then select **Next**.

5. In the **Enter password** dialog box, copy and paste in the **Tenant Password** provided by your lab hosting provider and then select **Sign in**.
6. In the Search bar of the Azure portal, type *Sentinel*, then select **Azure Sentinel**.
7. Select **+ Create**.
8. Next, select **+ Create a new workspace**.

Note First, you create a new Log Analytics Workspace.

9. Select your proper Subscription.
10. Select the **Create New** link for the Resource Group and enter a new resource group name of your choosing.
11. Under Instance details in the name field enter a name for your choosing for the Log Analytics Workspace.

Note: This name will also be the Azure Sentinel workspace name.

12. Select the region that is appropriate for you. The appropriate region may default or your instructor may have specific advice on which region to select.
13. Select **Review + Create**.
14. Select **Create**. Wait for the new Log Analytics workspace to appear in the list on the Add Azure Sentinel to a workspace page. This may take a minute.
15. Select the newly created workspace when it appears, then select **Add**.
16. Navigate around the newly created Azure Sentinel workspace to become familiar with the user interface options.

11.1.2 Task 2: Create a Watchlist.

In this task, you will create a watchlist.

1. In the search box at the bottom of the screen, enter *Notepad*. Select **Notepad** from the results.
2. Type *Hostname* then enter for a new line.
3. In Row 2 through 6, add the following hostnames: Host1 Host2 Host3 Host4 Host5
4. From the menu select, **File - Save As**, Name the file *HighValue.csv*. Then change the file type to **All files(.)**. Then select **Save**.
5. Close Notepad.
6. In Azure Sentinel, select the **Watchlist** option in the Configuration area.
7. Select **Add New** from the command bar.
8. In the Watchlist wizard, enter the following: Name: HighValueHosts Description: High Value Hosts Watchlist alias: HighValueHosts
9. Select, **Next: Source >**.
10. Browse for the *HighValue.csv* file you just created.
11. Select **Next: Review and Create >**.
12. Select **Create**.
13. The screen returns to the watchlists list.
14. Select your new watchlist. On the right tab, select **View in Log Analytics**.
15. The following KQL statement is automatically executed with the results displayed.

```
_GetWatchlist('HighValueHosts')
```

Note It could take a minute for the import to complete.

You can now use the _GetWatchlist('HighValueHosts') in your own KQL statements to access the list. The column to reference would be *Hostname*.

11.1.3 Task 3: Create a Threat Indicator.

In this task, you will create an indicator.

1. In Azure Sentinel, Select the **Threat intelligence** option in the Threat management area.
2. Select **Add New** from the command bar.
3. Review the different indicator types available in the Types dropdown. Then select **domain-name**. Enter your initials in the Domain box. An example would be fmg.com.
4. For the threat type, select **malicious-activity**.
5. For the name, enter the same value used for the Domain. An example would be fmg.com.
6. Set the valid from field to today's date.
7. Select **apply**.

Note It could take a minute for the indicator to appear.

8. Select **Logs** option in the General area. You may have to disable the "Always show queries" option to get to the query window.
9. Run the following KQL statement.

```
ThreatIntelligenceIndicator
```

Scroll the results to the right to see the DomainName column. You can also run the following KQL statement to just see the DomainName column.

```
ThreatIntelligenceIndicator
| project DomainName
```

11.2 You have completed the lab.

12 Module 6 - Lab 1 - Exercise 1 - Connect data to Azure Sentinel using data connectors

12.1 Lab scenario

You're a Security Operations Analyst working at a company that implemented Azure Sentinel. You must learn how to connect log data from the many different data sources in your organization. The organization has data from Microsoft 365, Microsoft 365 Defender, Azure resources, non-azure virtual machines, and network appliances.

You plan on using the Azure Sentinel data connectors to integrate the log data from the various sources. You need to write a connector plan for management that maps each of the organization's data sources to the proper Azure Sentinel data connector.

Important Warning! The Virtual Machine WIN1 and WIN2 are used in Module 7. Save your virtual machines. If you exit the lab without saving, you will be required to install the connectors again on WIN1 and WIN2.

12.1.1 Task 1: Access the Azure Sentinel Workspace.

In this task, you will access your Azure Sentinel workspace.

1. Log in to WIN1 virtual machine as Admin with the password: **Pa55w.rd**.
2. Open the browser, search for, download, and install the new Microsoft Edge browser. Start the new Edge browser.
3. In the Edge browser, navigate to the Azure portal at https://portal.azure.com.
4. In the **Sign in** dialog box, copy and paste in the **Tenant Email** account provided by your lab hosting provider and then select **Next**.
5. In the **Enter password** dialog box, copy and paste in the **Tenant Password** provided by your lab hosting provider and then select **Sign in**.
6. In the Search bar of the Azure portal, type *Sentinel*, then select **Azure Sentinel**.

7. Select your Azure Sentinel Workspace that you created in the previous lab.

12.1.2 Task 2: Connect the Azure Active Directory connector.

In this task, you will connect the Azure Active Directory connector.

1. In the Configuration area select **Data connectors**. In the Data Connectors page, select the **Azure Active Directory** connector from the list.
2. Select **Open connector page** on the connector information blade.
3. Select the **Sign-in Logs** and **Audit Logs** options from the Configuration area, then select **Apply Changes**.

12.1.3 Task 3: Connect the Azure Active Directory Identity Protection connector.

In this task, you will connect the Azure Active Directory Identity Protection connector.

1. From the Data Connectors Tab, select the **Azure Active Directory Identity Protection** connector from the list.
2. Select **Open connector page** on the connector information blade.
3. Select the **Connect** button.

12.1.4 Task 4: Connect the Azure Defender connector.

In this task, you will connect the Azure Defender connector.

1. From the Data Connectors tab, select the **Azure Defender** connector from the list.
2. Select **Open connector page** on the connector information blade.
3. Review the Connecting Options. Don't connect. This is for informational purposes only.

12.1.5 Task 5: Connect the Microsoft Cloud App Security connector.

In this task, you will connect the Microsoft Cloud App Security connector.

1. From the Data Connectors Tab, select the **Microsoft Cloud App Security** connector from the list.
2. Select **Open connector page** on the connector information blade.
3. Select **Alerts** and then select **Apply Changes**.

12.1.6 Task 6: Connect the Microsoft Defender for Office 365 connector.

In this task, you will connect the Microsoft Defender for Office 365 connector.

1. From the Data Connectors tab, select the **Microsoft Defender for Office 365** connector from the list.
2. Select **Open connector page** on the connector information blade.
3. Select **Connect**.

12.1.7 Task 7: Connect the Microsoft Defender for Identity connector.

In this task, you will connect the Microsoft Defender for Identity connector.

1. From the Data Connectors Tab, select the **Microsoft Defender for Identity** connector from the list.
2. Select **Open connector page** on the connector information blade.
3. Review the Connecting Options. Don't connect. This is for informational purposes only.

12.1.8 Task 8: Connect the Microsoft Defender for Endpoint connector.

In this task, you will connect the Microsoft Defender for Endpoint connector.

1. From the Data Connectors Tab, select the **Microsoft Defender for Endpoint** connector from the list.
2. Select Open connector page on the connector information blade.
3. Select **Connect**.

12.1.9 Task 9: Connect the Microsoft 365 Defender connector.

In this task, you will connect the Microsoft 365 Defender connector.

1. From the Data Connectors Tab, select the **Microsoft 365 Defender** connector from the list.
2. Select **Open connector page** on the connector information blade.
3. Select all the checkboxes for Microsoft Defender for Endpoint.
4. Select **Apply Changes**.

12.2 Proceed to Exercise 2

13 Module 6 - Lab 1 - Exercise 2 - Connect Windows devices to Azure Sentinel using data connectors

13.0.1 Task 1: Create a Windows Virtual Machine in Azure.

In this task, you will create a Windows virtual machine.

1. Login to WIN1 virtual machine as Admin with the password: **Pa55w.rd**.
2. In the Edge browser, navigate to the Azure portal at https://portal.azure.com.
3. In the **Sign in** dialog box, copy and paste in the **Tenant Email** account provided by your lab hosting provider and then select **Next**.
4. In the **Enter password** dialog box, copy and paste in the **Tenant Password** provided by your lab hosting provider and then select **Sign in**.
5. Select **Create a Resource**.
6. In the **search the Marketplace** box, enter *Windows 10*.
7. Select the **Create** dropdown for Microsoft Windows 10. Then select **Windows 10 Enterprise, version 20H2**.
8. Select your Subscription.
9. Create a new Resource Group named **rg-AZWIN01** if you have not done so already.

Note: This needs to be a new resource group. You are going to delete the Virtual machine after the exercise.

10. Set the Virtual Machine name to AZWIN01.
11. Set the Region to the appropriate region for your area. The appropriate region may default.
12. Enter a Username of your choosing that is acceptable for Azure.
13. Enter a Password of your choosing.

Hint: It might be easiest to use your tenant password.

14. Select Licensing confirmation.
15. Select **Review + Create**.
16. Select **Create**. Wait for the Resource to be created, this may take a few minutes.

13.0.2 Task 2: Connect an Azure Windows VM.

In this task, you will connect an Azure Windows virtual machine to Azure Sentinel.

1. In the Search bar of the Azure portal, type *Sentinel*, then select **Azure Sentinel**.
2. Select your Azure Sentinel Workspace you created earlier.
3. From the Data Connectors Tab, select the **Security Events** connector from the list.
4. Select your Azure Sentinel Workspace if prompted.
5. Select **Open connector page** on the connector information blade.

Note: The instructions for Install agent on a Windows Virtual Machine and Install agent on a non-Azure Windows Machine may be reversed. The links take you to the proper location even with the reversed text.

6. Select the **Install agent on a Windows Virtual Machine** option.
7. Select **Download & install agent for Azure Windows Virtual machines**.
8. Select the **AZWIN01** virtual machine in the list that you just created in the previous step, then select **Connect**. Wait until the connecting message disappears.
9. Select **Virtual machines** in the navigation list. You should now see the machine is connected.

Note: The virtual machine is only used in this task.

10. In the Azure portal search, enter *resource groups*. Select **Resource Groups**.
11. Select **rg-AZWIN01** from the list.
12. Select **Delete resource group** from the command bar.
13. Enter **rg-AZWIN01** into the "Are you sure you want to delete" pane, then select **Delete**.

13.0.3 Task 3: Connect a non-Azure Windows Machine.

In this task, you will connect a non-Azure Windows virtual machine to Azure Sentinel.

1. Login to WIN2 virtual machine as Admin with the password: **Pa55w.rd**.
2. Open the browser, search for, download, and install the new Microsoft Edge browser. Start the new Edge browser.
3. Open a browser and log into the Azure Portal at https://portal.azure.com with your credentials.
4. In the Search bar of the Azure Portal, type *Sentinel*, then select **Azure Sentinel**.
5. Select your Azure Sentinel Workspace.
6. From the Data Connectors tab, select the **Security Events** connector from the list.
7. Select **Open connector page** on the connector information blade.
8. In the Select which events to stream area, select **All Events**, then select **Apply Changes**.
9. Select the **Install agent on a non-Azure Windows Virtual Machine**.

Note: The instructions for Install agent on a Windows Virtual Machine and Install agent on a non-Azure Windows Machine may be reversed. The links take you to the proper location even with the reversed text.

10. Select **Download & install agent for non-Azure Windows Virtual machines**.
11. Select the link for **Download Windows Agent (64 bit)**.
12. Run the .exe file that is downloaded and confirm and User Account Control prompt that may appear.
13. Select **Next** on the Welcome dialog.
14. Select **I Agree** on the Microsoft Software License Terms page. On the Destination prompt select **Next**.
15. On the Agent Setup Options prompt, select **Connect the agent to Azure Log Analytics (OMS)** option, then select **Next**.
16. In the browser, copy the **Workspace ID** from the Agents Management page and paste into the Workspace ID in the dialog.
17. In the browser, copy the **Primary key** from the Agents Management page and paste into the Primary key in the dialog.
18. Select **Next**.
19. Select **Next** on the Microsoft Update page.
20. Then select **Install**.

13.0.4 Task 4: Install and collect Sysmon logs.

In this task, you will install and collect Sysmon logs.

You should still be connected to the WIN2 virtual machine. The following instructions will install Sysmon with the default configuration. You should research community based configurations for Sysmon to be used on production machines.

1. In the browser, go to https://docs.microsoft.com/sysinternals/downloads/sysmon
2. Download Sysmon from the page by select **Download Sysmon**.
3. Open the downloaded file and extract the files to a new directory c:\sysmon
4. In the Windows Taskbar for WIN2 search box, enter *command*. The search results will show command prompt app. Right-click on the command prompt app and select **Run as Administrator**. Confirm any User Account Control prompts that appear.
5. Enter *cd \sysmon*
6. type *notepad sysmon.xml* to create a new file.
7. Open a tab in the browser and navigate to: https://github.com/SwiftOnSecurity/sysmon-config/blob/master/sysmonco export.xml
8. Copy the contents of that file from Github to the sysmon.xml notepad file you just create and save the file.
9. In the command prompt type the following and press enter: sysmon.exe -accepteula -i sysmon.xml
10. In the browser, navigate to the Azure portal at https://portal.azure.com
11. In the Search bar of the Azure portal, type *Sentinel*, then select **Azure Sentinel**.
12. In Azure Sentinel, select **Settings** from the Configuration area and then select **Workspace settings** tab.
13. Make sure your Azure Sentinel Workspace is selected.
14. Select **Agents configuration** in Settings.
15. Select the **Windows Event logs** tab.
16. Select **Add windows event log** button.
17. Enter **Microsoft-Windows-Sysmon/Operational** in the Log name field.
18. Select **Apply**.

13.0.5 Task 5: Onboard Microsoft Defender for Endpoint Device.

In this task, you will on-board a device to Microsoft Defender for Endpoint.

Note: If you completed the labs in the first module of this course you have already performed this task. If you're using the same virtual machine from that lab exercise you don't need to do this task.

1. Login to WIN1 virtual machine as Admin with the password: **Pa55w.rd**.
2. Go to the Microsoft Defender Security Center at (https://securitycenter.microsoft.com) and login with the **Tenant Email** credentials if you are not currently in the portal.
3. Select **Settings** from the left menu bar.
4. Select **Onboarding** in the Device management section.
5. Select **Download Package**.
6. Extract the downloaded .zip file.
7. Run the Windows Command Prompt as **Administrator** and agree to any User Account Control prompts that appear.
8. Run the WindowsDefenderATPLocalOnboardingScript.cmd file that you just extracted as administrator. **Note** By default the file should be in the c:\users\admin\downloads directory. Answer Y to questions presented by the script.

9. From the Onboarding page in the Microsoft Defender Security Center portal, copy the detection test script and run in the open **Administrator: Command Prompt** window.
10. In the Microsoft Defender Security Center portal menu, select **Devices inventory** icon from the left navigation. You should now see your device in the list. **Note** It can take up to 5 minutes for the device to be displayed in the portal.

13.1 Proceed to Exercise 3

14 Module 6 - Lab 1 - Exercise 3 - Connect Linux hosts to Azure Sentinel using data connectors

14.0.1 Task 1: Access the Azure Sentinel Workspace.

In this task, you will access your Azure Sentinel workspace.

1. Log in to WIN1 virtual machine as Admin with the password: **Pa55w.rd**.
2. Open the browser, search for, download, and install the new Microsoft Edge browser if you have not already done so. Start the new Edge browser.
3. In the Edge browser, navigate to the Azure portal at https://portal.azure.com.
4. In the **Sign in** dialog box, copy and paste in the **Tenant Email** account provided by your lab hosting provider and then select **Next**.
5. In the **Enter password** dialog box, copy and paste in the **Tenant Password** provided by your lab hosting provider and then select **Sign in**.
6. In the Search bar of the Azure portal, type *Sentinel*, then select **Azure Sentinel**.
7. Select your Azure Sentinel Workspace you created in a previous lab.

14.0.2 Task 2: Connect a Linux Host using the Common Event Format connector.

In this task, you will connect a Linux host to Azure Sentinel with the Common Event Format (CEF) connector.

1. Select **Data connectors** from the Configuration area in Azure Sentinel. From the Data Connectors tab, select the **Common Event Format** connector from the list.
2. Select **Open connector page** on the connector information blade.
3. Copy to the clipboard the command shown in 1.2 Install the CEF collector on the Linux machine.
4. The next steps are specific to limitations in copying from a Virtual Machine in the lab environment. In the browser, navigate to https://outlook.office.com.
5. Create a New Message to *MOD Administrator*.
6. Paste the clipboard command that was copied from the connector page and send the email message.
7. Open an InPrivate browser session on your local (not a lab virtual machine) and navigate to https://outlook.office.com.
8. In the **Sign in** dialog box, copy and paste in the **Tenant Email** account provided by your lab hosting provider and then select **Next**.
9. In the **Enter password** dialog box, copy and paste in the **Tenant Password** provided by your lab hosting provider and then select **Sign in**.
10. Copy the command that you emailed to yourself.
11. Log in to LIN1 virtual machine as root with the password: **Passw0rd!** or whatever Linux machine credentials your lab provider has assembled for you.
12. Paste the command in the terminal window.
13. In the command, where you see the word ”python” change it to ”python3” and press enter to execute the command.

14.0.3 Task 3: Connect a Linux Host using the Syslog connector.

In this task, you will connect a Linux host to Azure Sentinel with the Syslog connector.

1. Connect to WIN1, which should already be in the Azure Sentinel portal.
2. From the Data Connectors tab, select the **Syslog** connector from the list.
3. Select **Open connector page** on the connector information blade.
4. Open the **Install agent on a non-Azure Linux Machine** section.
5. Select the link for **Download & install agent for non-Azure Linux machine**.
6. Select the tab for **Linux servers**.
7. Copy the command in the Download and onboard agent for Linux area.
8. The next steps are specific to limitations in copying from a Virtual Machine in the lab environment. In the browser, navigate to https://outlook.office.com.
9. Create a New Message to *MOD Administrator*.
10. Paste the clipboard command that was copied from the connector page.
11. Open a browser on your local (not a lab virtual machine) and navigate to https://outlook.office.com.
12. In the **Sign in** dialog box, copy and paste in the **Tenant Email** account provided by your lab hosting provider and then select **Next**.
13. In the **Enter password** dialog box, copy and paste in the **Tenant Password** provided by your lab hosting provider and then select **Sign in**.
14. Copy the command that you emailed to yourself.
15. Log in to LIN2 virtual machine as user *root* with the password: **Passw0rd!** or whatever Linux machine credentials your lab provider has assembled for you.
16. Paste the command in the terminal window and press **enter**.

14.0.4 Task 4: Configure the facilities you want to collect and their severities for the Syslog connector.

In this task, you will configure the Syslog collection facilities.

1. Connect to WIN1 virtual machine.
2. In Azure Sentinel portal, select **Settings** and then **Workspace settings** from the settings blade.
3. Select **Agents configuration** in the **Settings** area.
4. Select the **Syslog** tab.
5. Select the **Add facility** button.
6. Select **auth**.
7. Select the **Add facility** button.
8. Enter *authpriv* and press the +.
9. Select **Apply**.

14.1 Proceed to Exercise 4

15 Module 6 - Lab 1 - Exercise 4 - Connect Threat intelligence to Azure Sentinel using data connectors

15.0.1 Task 1: Connect Threat intelligence.

In this task, you will connect a Threat intelligence provider with the Threat intelligence - TAXII connector.

1. Login to WIN1 virtual machine as Admin with the password: **Pa55w.rd**.
2. In the Edge browser, navigate to the Azure portal at https://portal.azure.com.

3. In the **Sign in** dialog box, copy and paste in the **Tenant Email** account provided by your lab hosting provider and then select **Next**.
4. In the **Enter password** dialog box, copy and paste in the **Tenant Password** provided by your lab hosting provider and then select **Sign in**.
5. In the Search bar of the Azure portal, type *Sentinel*, then select **Azure Sentinel**.
6. Select your Azure Sentinel Workspace you created earlier.
7. From the Data Connectors tab, select the **Threat intelligence - TAXII (Preview)** connector.
8. Select **Open connector page** on the connector information blade.
9. In the Configuration area, for the Friendly name enter *PhishURLs*
10. For the API root URL enter https://limo.anomali.com/api/v1/taxii2/feeds/
11. Enter **107** for the Collection ID.
12. Enter **guest** for username.
13. Enter **guest** for the password.
14. Now select **Add** button.

Phishing URLs will be pulled and populate the ThreatIntelligenceIndicator table.

15.1 You have completed the lab.

16 Module 7 - Lab 1 - Exercise 1 - Activate a Microsoft Security rule

16.1 Lab scenario

You're a Security Operations Analyst working at a company that implemented Azure Sentinel. You must learn how to detect and mitigate threats using Azure Sentinel. You need to enable alerts from other Microsoft 365 and Azure services.

16.1.1 Task 1: Activate a Microsoft Security Rule

In this task, you will activate a Microsoft Security rule.

1. Log in to WIN1 virtual machine as Admin with the password: **Pa55w.rd**.
2. In the Edge browser, navigate to the Azure portal at https://portal.azure.com.
3. In the **Sign in** dialog box, copy and paste in the **Tenant Email** account provided by your lab hosting provider and then select **Next**.
4. In the **Enter password** dialog box, copy and paste in the **Tenant Password** provided by your lab hosting provider and then select **Sign in**.
5. In the Search bar of the Azure portal, type *Sentinel*, then select **Azure Sentinel**.
6. Select your Azure Sentinel Workspace.
7. Select **Analytics** from the Configuration area, then select the **Rule templates** tab.
8. In the search box, enter *defender*.
9. In the result set, select **Create incidents based on Microsoft Defender Advanced Threat Protection alerts**.

Note: The rule name could also be displayed as "Create incidents based on Microsoft Defender for Endpoint alerts".

10. On the right blade, select **Create rule**.
11. Change Filter by Severity to **Custom**.
12. Select **High** for the severity level.
13. Select the **Next : Automated response** button and then select **Next: Review** button.

14. Select the **Create** button.

17 Proceed to Exercise 2

18 Module 7 - Lab 1 - Exercise 2 - Create a Playbook

18.0.1 Task 1: Create a Security Operations Center Team in Microsoft Teams.

In this task, you will create a Microsoft Teams team for use in the lab.

1. Log in to WIN1 virtual machine as Admin with the password: **Pa55w.rd**.
2. Open the Microsoft Teams App from the Windows menu.
3. In the **Sign in** dialog box, copy and paste in the **Tenant Email** account provided by your lab hosting provider and then select **Next**.
4. In the **Enter password** dialog box, copy and paste in the **Tenant Password** provided by your lab hosting provider and then select **Sign in**.
5. Select **No, sign in to this app only**.
6. In Microsoft Teams, select **Teams**, then at the bottom, select **Join or create a team**.
7. Select **Create a Team** in the main window.
8. Select **From scratch**.
9. Select **Private**.
10. Give the team the name: **SOC** and select **Create**.
11. In the Add members to SOC, select **Skip**.
12. Click the **...** next to the team name SOC, and select **Add channel**.
13. Enter a channel name of *New Alerts* then select **Add**.

18.0.2 Task 2: Create a Playbook in Azure Sentinel.

In this task, you will create a Logic App that will be used as a Playbook in Azure Sentinel.

1. In the Edge browser, navigate to the Azure portal at https://portal.azure.com.
2. In the **Sign in** dialog box, copy and paste in the **Tenant Email** account provided by your lab hosting provider and then select **Next**.
3. In the **Enter password** dialog box, copy and paste in the **Tenant Password** provided by your lab hosting provider and then select **Sign in**.
4. In the Search bar of the Azure portal, type *Sentinel*, then select **Azure Sentinel**.
5. Select your Azure Sentinel Workspace you created earlier.
6. Select the **Community** page in the Configuration area on the left side of the page.
7. Select the **Go to Azure Sentinel community** button.
8. Select the **Playbooks** folder.
9. Select the **Post-Message-Teams** folder.
10. Select **Deploy to Azure** button.
11. Select your Azure Subscription.
12. For Resource Group, select **Create New** and enter *rg-Playbooks*.
13. For region, select the appropriate region for your situation. The default region will likely be optimal.
14. For User Name, enter the user name **Tenant Email**
15. Select **Review + create**.
16. Now select **create**.

Note Wait for the deployment to finish before proceeding to the next task.

18.0.3 Task 3: Update a Playbook in Azure Sentinel.

In this task, you will update the new playbook with the proper connection information.

1. In the Search bar of the Azure portal, type *Sentinel*, then select **Azure Sentinel**.
2. Select your Azure Sentinel Workspace.
3. Select the **Automation** from the Configuration area, and then select the **Playbooks** tab.
4. Click on the **Post-Message-Teams** playbook,
5. On the Logic App page for *Post-Message-Teams*, select **Edit**.
6. Click on the first Connections block at the top.
7. Select **Add new**, and sign in with your Azure subscription admin credentials.
8. Click on the second Connection block in the middle.
9. Select **Add new**, and sign in with your Azure subscription admin credentials.
10. Click on the third Connection block.
11. Select **Add new**, and sign in with your Azure subscription admin credentials.
12. In the Post a message block, for the Team, select the **X** at the end of the edit box. The edit box will be changed to a dropdown with a listing of the Teams. Select **SOC**.
13. For the Channel, select the **X** at the end of the edit box. The edit box will be changed to a dropdown with a listing of the Channels. Select **New Alerts**.
14. Select **Save**.

The Logic App will be used in a future lab.

18.1 Proceed to Exercise 3

19 Module 7 - Lab 1 - Exercise 3 - Create a Scheduled Query

19.0.1 Task 1: Create a Scheduled Query.

In this task, you will create a scheduled query.

1. Log in to WIN1 virtual machine as Admin with the password: **Pa55w.rd**.
2. In the **Sign in** dialog box, copy and paste in the **Tenant Email** account provided by your lab hosting provider and then select **Next**.
3. In the **Enter password** dialog box, copy and paste in the **Tenant Password** provided by your lab hosting provider and then select **Sign in**.
4. In the Search bar of the Azure portal, type *Sentinel*, then select **Azure Sentinel**.
5. Select your Azure Sentinel Workspace.
6. Select **Analytics** from the Configuration area.
7. Select the **Create** button, and select **Scheduled query rule**.
8. On the General tab, enter the Name *Inactive Account sign in attempts*.
9. For Tactics, select **Initial Access**.
10. For Severity, select **Medium**
11. Select **Next : Set rule logic >** button:
12. For the rule query, paste in the following KQL statement:

```
SigninLogs
| where ResultType == "50057"
| where ResultDescription =~ "User account is disabled. The account has been disabled by an administrat
| summarize StartTimeUtc = min(TimeGenerated), EndTimeUtc = max(TimeGenerated), count(), applicationCou
applicationSet = makeset(AppDisplayName) by UserPrincipalName, IPAddress
| extend timestamp = StartTimeUtc, AccountCustomEntity = UserPrincipalName, IPCustomEntity = IPAddress
```

Warning: When using the Paste function to the virtual machine. Extra | (pipe) characters could be added. Make sure what is pasted looks like the following KQL statement.

Note: If you select the link to "View query results", you should not receive any results. You should also not receive an error.

13. Review the Map entities. The entities are shown as mapped in the query because the query output includes fields:

timestamp = StartTimeUtc, AccountCustomEntity = UserPrincipalName, IPCustomEntity = IPAddress

14. Back in the Analytics rule wizard - Create new rule blade in the Query scheduling area, enter **5** and select **Minutes** for the Run query every option.
15. In the Query scheduling area, enter **1** and select **Days** for the Lookup data from the last option.
16. For the Alert threshold area, leave the options unchanged.

Note: Best practices are to manage thresholds in the alert rule KQL query statement.

17. For the Event grouping area, leave the **Group all events into a single alert** as the selected option.
18. Select the **Next: Incident settings** button.
19. On the Incident settings tab, review the default options.
20. Select the **Next: Automated response** button.
21. On the Automated response tab, select the playbook Post-Message-Teams you had previously created.
22. Select the **Next: Review** button.
23. Select **Create**.

19.0.2 Task 2: Test our new rule.

In this task, you will create a test your new scheduled query rule.

1. In the Search bar of the Azure portal, type *Azure Active Directory*. Then select **Azure Active Directory**.
2. Select **Users** in the Manage area.
3. Select User **Christie Cline** in the list. The Christie Cline | Profile page is displayed.
4. Select **Edit**.
5. In the settings area, change **Block sign in** to **Yes**.
6. Now select **Save** from the Command bar.
7. In the Azure portal, select the user avatar at the top right and sign out.
8. Close your browser.
9. Open a browser and navigate to https://portal.office.com and try to login with user ChristieC@**Tenant Email domain** and password should be the same as your admin's tenant password. You should receive a warning that your account has been locked.
10. Close your browser. Wait 10 minutes for the alert to process.
11. In the Edge browser, go to the Azure portal at https://portal.azure.com.
12. In the **Sign in** dialog box, copy and paste in the **Tenant Email** account provided by your lab hosting provider for Admin user and then select **Next**.
13. In the **Enter password** dialog box, copy and paste in the **Tenant Password** provided by your lab hosting provider for Admin user and then select **Sign in**.

14. In the Search bar of the Azure portal, type *Sentinel*, then select **Azure Sentinel**.
15. Select your Azure Sentinel Workspace.
16. Select the **Incidents** menu option.
17. You should see the newly created Incident. Select the Incident and review the information in the right blade.
18. Open Microsoft Teams. Goto your *SOC* Team, ... and see the message post about the incident.

19.1 Proceed to Exercise 4

20 Module 7 - Lab 1 - Exercise 4 - Understand Detection Modeling

20.0.1 Task 1: Understand the Attacks

You will perform no actions in this exercise. This exercise is an explanation of the attacks you will perform.

The attack patterns are based on an open-source project: https://github.com/redcanaryco/atomic-red-team

NOTE Some settings are triggered in a smaller timeframe just for our lab purpose.

20.0.1.1 Attack 1 - Persistence with Registry Key Add.

This attack is run from a command prompt:

```
REG ADD "HKCU\SOFTWARE\Microsoft\Windows\CurrentVersion\Run" /V "SOC Test" /t REG_SZ /F /D "C:\temp\sta
```

20.0.1.2 Attack 2 - User Add and Elevate Privilege

Attackers will add new users and elevate the new user to the Administrators group. This enables the attacker to logon with a different account that is privileged.

```
net user theusernametoadd /add
net user theusernametoadd ThePassword1!
net localgroup administrators theusernametoadd /add
```

20.0.2 Attack 3 -DNS / C2

This attack will simulate a command and control (C2) communication.

```
param(
    [string]$Domain = "microsoft.com",
    [string]$Subdomain = "subdomain",
    [string]$Sub2domain = "sub2domain",
    [string]$Sub3domain = "sub3domain",
    [string]$QueryType = "TXT",
        [int]$C2Interval = 8,
        [int]$C2Jitter = 20,
        [int]$RunTime = 240
)


$RunStart = Get-Date
$RunEnd = $RunStart.addminutes($RunTime)

$x2 = 1
$x3 = 1
Do {
    $TimeNow = Get-Date
    Resolve-DnsName -type $QueryType $Subdomain".$(Get-Random -Minimum 1 -Maximum 999999)."$Domain -Qui

    if ($x2 -eq 3 )
```

```
    {
        Resolve-DnsName -type $QueryType $Sub2domain".$(Get-Random -Minimum 1 -Maximum 999999)."$Domain

        $x2 = 1

    }
    else
    {
        $x2 = $x2 + 1
    }

    if ($x3 -eq 7 )
    {

        Resolve-DnsName -type $QueryType $Sub3domain".$(Get-Random -Minimum 1 -Maximum 999999)."$Domain

        $x3 = 1

    }
    else
    {
        $x3 = $x3 + 1
    }


    $Jitter = ((Get-Random -Minimum -$C2Jitter -Maximum $C2Jitter) / 100 + 1) +$C2Interval
    Start-Sleep -Seconds $Jitter
}
Until ($TimeNow -ge $RunEnd)
```

20.0.3 Task 2: Understand Detection Modeling.

The attack-detect configuration cycle used in this lab represents all data sources even though you are only focused on two specific data sources.

To build a detection, you first start with building a KQL statement. Since you will attack a host, you will have representative data to start building the KQL statement.

The following lab runs the same attacks on a Windows host with Defender for Endpoint installed and Windows with Sysmon installed. As you build the detections, you will see the difference in data normalization for each.

After you have the KQL statement, you create the Analytical Rule.

Once the rule triggers and creates the alerts and incidents, you then investigate to decide if you are providing fields that help Security Operations Analysts in their investigation.

Next, make any other changes to the analytics rule.

21 Proceed to Exercise 5

22 Module 7 - Lab 1 - Exercise 5 - Conduct attacks

22.0.1 Task 1: Attack Windows configured with Defender for Endpoint.

In this task, you will perform attacks on a host with Microsoft Defender for Endpoint configured.

1. Login to WIN1 virtual machine as Admin with the password: **Pa55w.rd**.
2. In the search of the task bar, enter *Command*. Command Prompt will be displayed in the search results. Right-click on the Command Prompt and select **Run as Administrator**. Confirm any User Account Control prompts that appear.
3. In the command prompt, enter the command in each row pressing Enter key after each row:

```
cd \
mkdir temp
cd temp
```

4. Attack 1 - Copy and run this command:

```
REG ADD "HKCU\SOFTWARE\Microsoft\Windows\CurrentVersion\Run" /V "SOC Test" /t REG_SZ /F /D "C:\temp\sta
```

5. Attack 3 - Copy and run this command:

```
notepad c2.ps1
```

Select **Yes** to create a new file and copy the following PowerShell script into c2.ps1 and select **save**.

Note Paste into the Virtual Machine might have a limited length. Paste this in three sections to ensure all the script is pasted into the Virtual Machine. Make sure the script looks as it does in these instructions within the notepad c2.ps1 file.

```
param(
    [string]$Domain = "microsoft.com",
    [string]$Subdomain = "subdomain",
    [string]$Sub2domain = "sub2domain",
    [string]$Sub3domain = "sub3domain",
    [string]$QueryType = "TXT",
        [int]$C2Interval = 8,
        [int]$C2Jitter = 20,
        [int]$RunTime = 240
)


$RunStart = Get-Date
$RunEnd = $RunStart.addminutes($RunTime)

$x2 = 1
$x3 = 1
Do {
    $TimeNow = Get-Date
    Resolve-DnsName -type $QueryType $Subdomain".$(Get-Random -Minimum 1 -Maximum 999999)."$Domain -Qui

    if ($x2 -eq 3 )
    {
        Resolve-DnsName -type $QueryType $Sub2domain".$(Get-Random -Minimum 1 -Maximum 999999)."$Domain

        $x2 = 1

    }
    else
    {
        $x2 = $x2 + 1
    }

    if ($x3 -eq 7 )
    {

        Resolve-DnsName -type $QueryType $Sub3domain".$(Get-Random -Minimum 1 -Maximum 999999)."$Domain

        $x3 = 1

    }
    else
    {
        $x3 = $x3 + 1
```

```
    }


    $Jitter = ((Get-Random -Minimum -$C2Jitter -Maximum $C2Jitter) / 100 + 1) +$C2Interval
    Start-Sleep -Seconds $Jitter
}
Until ($TimeNow -ge $RunEnd)
```

At the command prompt, enter the following, enter the command in each row pressing Enter key after each row:

```
powershell
.\c2.ps1
```

Note: You will see resolve errors. This is to be expected. Let this command/powershell script run in the background. Don't close the window. The command needs to generate log entries for some hours. You can proceed to the next task and next exercises while this script runs. The data created by this task will be used in the Threat Hunting lab later. This process will not create substantial amounts of data or processing.

22.0.2 Task 2: Attack Windows configured with Sysmon

In this task, you will perform attacks on a host with the Security Events connector configured and Sysmon configured.

1. Login to WIN2 virtual machine as Admin with the password: **Pa55w.rd**.
2. In the search of the task bar, enter *CMD*. Command Prompt will be displayed in the search results. Right-click on the Command Prompt and select **Run as Administrator**.
3. In the command prompt, enter the command in each row pressing Enter key after each row:

```
cd \
mkdir temp
cd \temp
```

4. Attack 1 - Copy and run this command:

```
REG ADD "HKCU\SOFTWARE\Microsoft\Windows\CurrentVersion\Run" /V "SOC Test" /t REG_SZ /F /D "C:\temp\sta
```

5. Attack 2 - Copy and run this command, enter the command in each row pressing Enter key after each row:

```
net user theusernametoadd /add
net user theusernametoadd ThePassword1!
net localgroup administrators theusernametoadd /add
```

22.1 Proceed to Exercise 6

23 Module 7 - Lab 1 - Exercise 6 - Create Detections

23.0.1 Task 1: Attack 1 Detection with Sysmon

In this task, you will create a detection for Attack 1 on the host with the Security Events connector and Sysmon installed.

The attack creates a registry key that runs on startup.

```
REG ADD "HKCU\SOFTWARE\Microsoft\Windows\CurrentVersion\Run" /V "SOC Test" /t REG_SZ /F /D "C:\temp\sta
```

1. Login to WIN1 virtual machine as Admin with the password: **Pa55w.rd**.
2. In the Edge browser, navigate to the Azure portal at https://portal.azure.com.
3. In the **Sign in** dialog box, copy and paste in the **Tenant Email** account for admin provided by your lab hosting provider and then select **Next**.
4. In the **Enter password** dialog box, copy and paste in the **Tenant Password** for admin provided by your lab hosting provider and then select **Sign in**.

5. In the Search bar of the Azure portal, type *Sentinel*, then select **Azure Sentinel**.
6. Select your Azure Sentinel Workspace you created earlier.
7. Select **Logs** from the General section.
8. First, you need to see where the data is stored. Since you just performed the attacks. Set the Log Time Range to **Last 24 hours**.
9. Run the following KQL Statement

```
search "temp\\startup.bat"
```

10. The results show for three different tables: DeviceProcessEvents DeviceRegistryEvents Event

    The Device* tables are from Defender for Endpoint (Data Connector - Microsoft 365 Defender). Event is from our Data Connector Security Events.

    Since we are receiving data from two different sources - Sysmon and Defender for Endpoint, we will need to build two KQL statements that could be unioned later. In our initial investigation, you will look at each separately.

11. Our first data source is Sysmon from Windows hosts. Run the following KQL Statement.

```
search in (Event) "temp\\startup.bat"
```

The results now only show for the Event table.

12. Expand the rows to see all the columns related to the record. A few of the fields like EventData and ParameterXml have multiple data items stored as structured data. This makes it difficult to query on specific fields.
13. Next, we have to build a KQL statement that parses the data from each row, allowing us to have meaningful fields. In the Azure Sentinel Community on GitHub, there are many examples of Parsers in the Parsers folder. Open another tab in your browser and navigate to: https://github.com/Azure/Azure-Sentinel
14. Select the **Parsers** folder, then **Sysmon** folder. You should be viewing: Azure-Sentinel/Parsers/Sysmon/Sysmon-v12.0.txt
15. Select the Sysmon-v12.0.txt file to view.

At the top of the file, you see a Let statement querying the Event table and storing to a variable named EventData.

```
let EventData = Event
| where Source == "Microsoft-Windows-Sysmon"
| extend RenderedDescription = tostring(split(RenderedDescription, ":")[0])
| project TimeGenerated, Source, EventID, Computer, UserName, EventData, RenderedDescription
| extend EvData = parse_xml(EventData)
| extend EventDetail = EvData.DataItem.EventData.Data
| project-away EventData, EvData  ;
```

Further down in the file, you see another let statement looking at EventID == 13 and using the EventData variable as input.

```
let SYSMON_REG_SETVALUE_13=()
{
    let processEvents = EventData
    | where EventID == 13
    | extend RuleName = EventDetail.[0].["#text"], EventType = EventDetail.[1].["#text"], UtcTime = Eve
    ProcessId = EventDetail.[4].["#text"], Image = EventDetail.[5].["#text"], TargetObject = EventDetai
    | project-away EventDetail  ;
    processEvents;

};
```

This looks like a good start.

16. You use the above statement to create your own KQL statement to display all Registry Key Set Value rows. Run the following KQL query:

```
Event
| where Source == "Microsoft-Windows-Sysmon"
| where EventID == 13
| extend RenderedDescription = tostring(split(RenderedDescription, ":")[0])
| project TimeGenerated, Source, EventID, Computer, UserName, EventData, RenderedDescription
| extend EvData = parse_xml(EventData)
| extend EventDetail = EvData.DataItem.EventData.Data
| project-away EventData, EvData
| extend RuleName = EventDetail.[0].["#text"], EventType = EventDetail.[1].["#text"], UtcTime = EventDe
    ProcessId = EventDetail.[4].["#text"], Image = EventDetail.[5].["#text"], TargetObject = EventDetai
    | project-away EventDetail
```

17. You could continue to build your detection rule from here, but this KQL statement looks like it could be reused in other detection rule's KQL statements.

```
In the Log window, select **Save**, then **Save**.
In the Save flyout, enter the following:

Name: Event_Reg_SetValue
Save as: Function
Function Alias: Event_Reg_SetValue
Category: Sysmon
```

18. Open a new Log Query Tab. Then run the following KQL Statement:

```
Event_Reg_SetValue
```

Depending on the current data collection, you could receive many rows. This is expected. Our next task is to filter to our specific scenario.

19. Run the following KQL Statement:

```
Event_Reg_SetValue | search "startup.bat"
```

This returns our specific record that we can now review the data to see what we can change to identify rows.

20. From our Threat Intelligence, we know that the Threat Actor is using reg.exe to add the registry key. The directory is c:\temp. The startup.bat can be a different name. Run the following script

```
Event_Reg_SetValue
| where Image contains "reg.exe"
```

This is a good start. Next, you need to return results only for c:\temp directory.

21. Next, enter the following KQL statement:

```
Event_Reg_SetValue
| where Image contains "reg.exe"
| where Details startswith "C:\\TEMP"
```

This looks like a good detection rule.

22. It is important to help the Security Operations Analyst by providing as much context about the alert as you can. This includes projecting Entities for use in the investigation graph. Run the following query:

```
Event_Reg_SetValue
| where Image contains "reg.exe"
| where Details startswith "C:\\TEMP"
| extend timestamp = TimeGenerated, HostCustomEntity = Computer, AccountCustomEntity = UserName
```

23. Now that you have a good detection rule, in the Log window with the query, select the **New alert rule** in the Command Bar, and select **Create Azure Sentinel alert**.

24. This starts our Analytics rule wizard. For the General Tab enter:

    Name: Sysmon Startup RegKey

    Description: Sysmon Startup Regkey in c:\temp

    Tactics: Persistence

    Severity: High

Select **Next : Set rule logic**.

25. On the **Set rule logic** tab, the **Rule query and Map entities** should already be populated.

26. For Query scheduling set the following:

- Run Query every: 5 minutes
- Look data from the last: 1 Day

Note We are purposely generating many incidents for the same data. This enables the Lab to use these alerts.

27. Leave the rest of the options to the defaults. Select **Next : Incident settings** button.

28. For the Incident settings set the following:

- Incident settings: Enabled
- Alert grouping: Disabled

Select **Next : Automated response** button.

29. For the Automated response tab set the following:

- Select Post-Message-Teams.

Select **Next : Review** button.

30. On the Review tab, select the **Create** button.

23.0.2 Task 2: Attack 1 Detection with Defender for Endpoint

In this task, you will create a detection for Attack 1 on the host with the Microsoft Defender for Endpoint configured.

The attack creates a registry key that runs on startup.

```
REG ADD "HKCU\SOFTWARE\Microsoft\Windows\CurrentVersion\Run" /V "SOC Test" /t REG_SZ /F /D "C:\temp\sta
```

1. In the Azure Sentinel portal, Select **Logs** from the General section.

2. First, you need to see where the data is stored. Since you just performed the attacks.

   Set the Log Time Range to Last 24 hours.

3. Run the following KQL Statement:

```
search "temp\\startup.bat"
```

4. The results show for three different tables: DeviceProcessEvents DeviceRegistryEvents Event

   The Device* tables are from Defender for Endpoint (Data Connector - Microsoft 365 Defender). Event is from our Data Connector Security Events.

   Since we are receiving data from two different sources - Sysmon and Defender for Endpoint. We will need to build two KQL statements that could be unioned later. But our initial investigation, you will look at each separately.

5. This detection will focus on data from Defender for Endpoint. Run the following KQL Statement:

```
search in (Device*) "temp\\startup.bat"
```

6. The table - DeviceRegistryEvents looks to have the data already normalized and easy for us to query. Expand the rows to see all the columns related to the record.

7. From our Threat Intelligence, we know that the Threat Actor is using reg.exe to add the registry key. The directory is c:\temp. The startup.bat can be a different name. Enter this KQL statement:

```
DeviceRegistryEvents
| where ActionType == "RegistryValueSet"
| where InitiatingProcessFileName == "reg.exe"
| where RegistryValueData startswith "c:\\temp"
```

This looks like a good detection rule.

8. It is important to help the Security Operations Center Analyst by providing as much context about the alert as you can. This includes projecting Entities for use in the investigation graph.

```
DeviceRegistryEvents
| where ActionType == "RegistryValueSet"
| where InitiatingProcessFileName == "reg.exe"
| where RegistryValueData startswith "c:\\temp"
| extend timestamp = TimeGenerated, HostCustomEntity = DeviceName, AccountCustomEntity = InitiatingProc
```

9. Now that you have a good detection rule, in the Log window with the query, select the **New alert rule** in the Command Bar. Then select **Create Azure Sentinel alert**.
10. This starts our Analytics rule wizard. For the General Tab, enter:

    Name: D4E Startup RegKey

    Description: D4E Startup Regkey in c:\temp

    Tactics: Persistence

    Severity: High
11. Select **Next : Set rule logic** button.
12. On the Set rule logic tab, the Rule query and Map entities should already be populated.
13. For Query scheduling set the following:

- Run Query every: 5 minutes
- Look data from the last: 1 Days

Note We are purposely generating many incidents for the same data. This enables the Lab to use these alerts.

14. Leave the rest of the options to the defaults. Select **Next : Incident settings**:
15. For the Incident settings set the following:

- Incident settings: Enabled
- Alert grouping: Disabled

Select **Next : Automated response**:

16. For the Automated response tab set the following:

- Select Post-Message-Teams.
- Select **Next: Review**.

17. On the Review and create tab, select **Create**.

23.0.3 Task 3: Attack 2 Detection with SecurityEvent

In this task, you will create a detection for Attack 2 on the host with the Security Events connector and Sysmon installed.

The attack creates a new user and adds the user to the local administrators.

```
net user theusernametoadd /add
net user theusernametoadd ThePassword1!
net localgroup administrators theusernametoadd /add
```

1. Select **Logs** from the General section of the Azure Sentinel portal.
2. First, you need to see where the data is stored. Since you just performed the attacks.

   Set the Log Time Range to Last 24 hours.
3. Run the following KQL Statement:

```
search "administrators"
```

4. The results show the following tables: Event SecurityEvent
5. Our first data source is SecurityEvent. Time to research what event ID Windows uses to identify adding a member to a privileged group. The following EventID and Event are what we are looking for:

4732 - A member was added to a security-enabled local group.

Running the following script:

```
SecurityEvent
| where EventID == "4732"
| where TargetAccount == "Builtin\\Administrators"
```

6. Expand the rows to see all the columns related to the record. The user name we are looking for doesn't show. The issue is that instead of storing the user name, the security identifier (SID) is stored. The following KQL will try to match the SID to populate the TargetUserName that was added to the Administrators group.

```
SecurityEvent
| where EventID == "4732"
| where TargetAccount == "Builtin\\Administrators"
| extend Acct = MemberSid, MachId = SourceComputerId
| join kind=leftouter (
     SecurityEvent
     | summarize count() by TargetSid, SourceComputerId, TargetUserName
     | project Acct1 = TargetSid, MachId1 = SourceComputerId, UserName1 = TargetUserName
) on $left.MachId == $right.MachId1, $left.Acct == $right.Acct1
```

This looks like a good detection rule.

Note: This KQL might not return the expected results because of the small dataset used in the lab.

7. It is important to help the Security Operations Analyst by providing as much context about the alert as you can. This includes projecting Entities for use in the investigation graph. Run the following script:

```
SecurityEvent
| where EventID == "4732"
| where TargetAccount == "Builtin\\Administrators"
| extend Acct = MemberSid, MachId = SourceComputerId
| join kind=leftouter (
     SecurityEvent
     | summarize count() by TargetSid, SourceComputerId, TargetUserName
     | project Acct1 = TargetSid, MachId1 = SourceComputerId, UserName1 = TargetUserName
) on $left.MachId == $right.MachId1, $left.Acct == $right.Acct1
| extend timestamp = TimeGenerated, HostCustomEntity = Computer, AccountCustomEntity = UserName1
```

8. Now that you have a good detection rule, in the Log window with the query, select **New alert rule** in the Command Bar, then select **Create Azure Sentinel alert**.
9. This starts our Analytics rule wizard. For the General Tab, enter:

- Name: SecurityEvents Local Administrators User Add
- Description: SecurityEvents Local Administrators User Add

- Tactics: Privilege Escalation
- Severity: High

Select **Next : Set rule logic** button.

10. On the Set rule logic tab, the Rule query and Map entities should already be populated.
11. For Query scheduling set the following:

- Run Query every: 5 minutes
- Look data from the last: 1 Day

Note We are purposely generating many incidents for the same data. This enables the Lab to use these alerts.

12. Leave the rest of the options to the defaults. Select **Next : Incident settings**:
13. For the Incident settings set the following:

- Incident settings: Enabled
- Alert grouping: Disabled
- Select **Next: Automated response**

14. For the Automated response tab set the following:

- Select **Post-Message-Teams**.

Select **Next : Review** button.

15. On the Review tab, select **Create**.

23.1 Proceed to Exercise 7

24 Module 7 - Lab 1 - Exercise 7 - Investigate Incidents

24.0.1 Task 1: Investigate an incident.

In this task, you will investigate an incident.

1. Log in to WIN1 virtual machine as Admin with the password: **Pa55w.rd**.
2. In the **Sign in** dialog box, copy and paste in the **Tenant Email** account provided by your lab hosting provider and then select **Next**.
3. In the **Enter password** dialog box, copy and paste in the **Tenant Password** provided by your lab hosting provider and then select **Sign in**.
4. In the Search bar of the Azure portal, type *Sentinel*, then select **Azure Sentinel**.
5. Select your Azure Sentinel Workspace you created earlier.
6. Select the **Incidents** page.
7. Review the list of Incidents

Note: The analytical rules are generating alerts and incidents on the same specific log entry. This is done to generate more alerts and incidents to be utilized in the lab.

8. Select a Sysmon Startup RegKey incident.
9. Select **View full details** button.
10. On the left side of the page, change the Status to **Active** and then select **Apply**.
11. In the Tag area, select + and add a tag named **RegKey** and select **Ok**.
12. On the right side of the page, select the tab **Comments**.
13. Enter in the Comments: *I will research this. *
14. Select the **Comments** button to submit the new comment.
15. Select the **Entities** tab and review.
16. Select the **Alerts** tab.

Note: For the alert shown, notice to the far right there is an option for View Playbooks. This allows for the manual execution of a playbook.

17. Select the **Investigate** button.
18. Select the **Sysmon Startup RegKey Alert** graphic.
19. Select **Timeline** and review.
20. Select **Info** and review.
21. Select **Entities** and review.
22. Select **Insights** and review.
23. Select the **System Account** graphic.
24. Select **Timeline** and review.
25. Select **Info** and review.
26. Select **Entities** and review.
27. Select **Insights** and review.
28. Select the Base20E Host graphic (your WIN1 device name may vary depending on how it was deployed by your lab hoster).
29. Select **Timeline** and review.
30. Select **Info** and review.
31. Select **Entities** and review.
32. Select **Insights** and review.
33. Select the **Alert** in the graph. A menu should appear around the icon. Select related alerts.
34. Explore related Alerts.

24.1 Proceed to Exercise 8

25 Module 7 - Lab 1 - Exercise 8 - Create workbooks

25.1 Lab scenario

You're a Security Operations Analyst working at a company that implemented Azure Sentinel. You must design workbooks with advanced visualizations.

25.1.1 Task 1: Explore Workbooks.

In this task, you will explore the configuration of a workbook.

1. Login to WIN1 virtual machine as Admin with the password: **Pa55w.rd**.
2. In the Edge browser, navigate to the Azure portal at https://portal.azure.com.
3. In the **Sign in** dialog box, copy and paste in the **Tenant Email** account provided by your lab hosting provider and then select **Next**.
4. In the **Enter password** dialog box, copy and paste in the **Tenant Password** provided by your lab hosting provider and then select **Sign in**.
5. In the Search bar of the Azure portal, type *Sentinel*, then select **Azure Sentinel**.
6. Select your Azure Sentinel Workspace.
7. Select **Workbooks**.
8. Select **Identity and Access**, then select **View template**.
9. Review Workbook.
10. Select **Workbooks**
11. In the Templates tab search for and select **Azure AD Sign-on logs**, then select **Save**.

12. Select the location that is appropriate for your location. The appropriate location usually defaults.
13. Select **View saved workbook**.
14. Select **Edit** in command bar.
15. In the Sign-in Location area, select **Edit** at the bottom of the grid.

25.1.1.1 To format columns, the Column setting panel provides customization options, do the following:

16. Select **Column Settings** in the Query Command bar.
17. Select the column **Failure Count|Interrupt Count**.
18. Review the settings, including the Column renderer and Color palette.
19. Select the column **Trend**.
20. Review the settings, including the Column renderer and Color palette.
21. Select **cancel**.

25.1.1.2 To have one tile/grid control filter the results in another tile/grid do the following:

22. Select **Advanced Settings** tab in the Query.
23. Review the When items are selected, export parameters. Notice the LocationDetail field is selected.
24. Select **Done Editing** at the bottom of the query.
25. Select **Edit** for the Device Sign-in details table on the right side of the screen.
26. In the query, locate "LocationDetails". The query is using the parameter exported from the other query to filter results.
27. Select **Done Editing** for the query.
28. Select **Done Editing** for the workbook.

25.1.2 Task 2: Create a Workbook.

In this task, you will create a new workbook with advanced visualizations.

1. Select **Workbooks** in the Azure Sentinel portal.
2. Select **Add workbook**
3. Select **Edit**

25.1.2.1 Edit Header text:

4. Change *New workbook* to *My workbook*.
5. Select **Done Editing**.
6. Select **Edit** for the only visible graph.
7. Review the KQL statement that provides a union of counts across multiple tables.
8. Select the **Done Editing**.
9. Select **...** then select **Add**, then select **Add query**.
10. Enter *SecurityEvent*, then select **Run Query**.
11. Change the Timerange to **Last 3 days**.
12. Change the Visualization to different options and see the results.
13. Change the Visualization to **Time chart**.
14. Select **Style** from the Query tab.
15. Select the **Make this item a custom width** box.

16. Set the Percent width to **75** and Max Width to **75**.
17. Select **Advanced Settings** from the Query tab.
18. Select **Enable time range brushing** box.
19. Enter *demoparam* for **Export selected time range as parameter**.
20. Select **Done Editing**.
21. On the displayed grid, click once, hold, and drag. This will display a selected range.
22. Select **Add**, then **Add query**.

Enter the following KQL command for the query:

```
SecurityEvent
```

23. For Time Range, select **demoparam**.
24. Change the Visualization to **Grid**.
25. Select the **Style** tab.
26. Select **Make this item a custom width**.
27. Change percentage width to **25** and maximum width to **25**.

Done Editing for the Query

28. Select **Done Editing for the Workbook**.
29. Select **Save** and select **Save** again if prompted.
30. Select **Workbooks** in the Azure Sentinel portal.
31. Select the **My workbooks** tab.
32. Select the workbook you just created.
33. Select view saved workbook.

Note: Remember to try the timeslice by dragging on the grid.

25.2 You have completed the lab.

26 Module 8 - Lab 1 - Exercise 1 - Perform Threat Hunting in Azure Sentinel

26.1 Lab scenario

You're a Security Operations Analyst working at a company that implemented Azure Sentinel. You have received threat intelligence about a Command and Control (C2) technique. You need to perform a hunt and watch for the threat.

Note The log data used in the lab was created in a previous module in the course.

Note Because you already experienced the process of exploring data in a previous module, the lab provides a KQL statement to start with.

26.1.1 Task 1: Create a hunting query

In this task, you will create a hunting query, bookmark a result, and create a Livestream.

1. Log in to WIN1 virtual machine as Admin with the password: **Pa55w.rd**.
2. In the Edge browser, navigate to the Azure portal at https://portal.azure.com.
3. In the **Sign in** dialog box, copy and paste in the **Tenant Email** account provided by your lab hosting provider and then select **Next**.
4. In the **Enter password** dialog box, copy and paste in the **Tenant Password** provided by your lab hosting provider and then select **Sign in**.
5. In the Search bar of the Azure portal, type *Sentinel*, then select **Azure Sentinel**.

6. Select your Azure Sentinel Workspace.
7. Select **Logs**
8. Enter the following KQL Statement in the New Query 1 space:

```
let lookback = 2d;
DeviceEvents
| where TimeGenerated >= ago(lookback)
| where ActionType == "DnsQueryResponse"
| extend c2 = substring(tostring(AdditionalFields.DnsQueryString),0,indexof(tostring(AdditionalFields.D
| where c2 startswith "sub"
| summarize count() by bin(TimeGenerated, 3m), c2
| where count_ > 5
| render timechart
```

9. The goal of this statement is to provide a visualization to check for a C2 beaconing out on a consistent basis. Take time to adjust the 3m setting to 30s and more. Change the count_ > 5 setting to other threshold counts to witness the impact.
10. You have now identified DNS requests that are beaconing to a C2 server. Next, determine which devices are beaconing. Enter the following KQL Statement:

```
let lookback = 2d;
DeviceEvents
| where TimeGenerated >= ago(lookback)
| where ActionType == "DnsQueryResponse"
| extend c2 = substring(tostring(AdditionalFields.DnsQueryString),0,indexof(tostring(AdditionalFields.D
| where c2 startswith "sub"
| summarize cnt=count() by bin(TimeGenerated, 5m), c2, DeviceName
| where cnt > 15
```

Note The generate log data is only from one device.

11. Select the **Hunting** page in the Threat Management area of the Azure Sentinel portal.
12. Select **New Query** from the command bar.
13. For the Query enter the following KQL statement:

```
let lookback = 2d;
DeviceEvents
| where TimeGenerated >= ago(lookback)
| where ActionType == "DnsQueryResponse"
| extend c2 = substring(tostring(AdditionalFields.DnsQueryString),0,indexof(tostring(AdditionalFields.D
| where c2 startswith "sub"
| summarize cnt=count() by bin(TimeGenerated, 5m), c2, DeviceName
| where cnt > 15
```

14. For the Name enter type *C2 Hunt*
15. For the Entity Mapping enter:

    For the Host select **DeviceName** and then select **Add**. For the Timestamp select **TimeGenerated** and then select **Add**.
16. Select **Create**.
17. In the Azure Sentinel | Hunting blade search for the query you just created in the list, *C2 Hunt*.
18. Select **C2 Hunt** in the list.
19. Select the **Run Query** button on the right side of the page.
20. The result count is displayed at the top of the flyout.
21. Select **View Results**.
22. Select the first row in the results.
23. Select **Add bookmark**.

24. Select **Create** in the pane that appears.
25. Return to the Hunting page in the Azure Sentinel portal.
26. Select the **Bookmarks** tab.
27. Select the bookmark in the results list.
28. Select **Investigate** in the flyout pane.
29. Explore the Investigation graph.
30. Return to the Hunting page in the Azure Sentinel portal.
31. Select the **Queries** tab
32. Select the **C2 Hunt** query.
33. Select the **...** at the end of the row to open the context menu.
34. Select **Add to livestream**.

27 Proceed to Exercise 2

28 Module 8 - Lab 1 - Exercise 2 - Threat Hunting using Notebooks with Azure Sentinel

28.1 Lab scenario

You're a Security Operations Analyst working at a company that implemented Azure Sentinel. You need to explore the benefits of threat hunting with Azure Sentinel Notebooks.

28.1.1 Task 1: Explore Notebooks

In this task, you will explore using notebooks in Azure Sentinel.

1. Log in to WIN1 virtual machine as Admin with the password: **Pa55w.rd**.
2. In the Edge browser, navigate to the Azure portal at https://portal.azure.com.
3. In the **Sign in** dialog box, copy and paste in the **Tenant Email** account provided by your lab hosting provider and then select **Next**.
4. In the **Enter password** dialog box, copy and paste in the **Tenant Password** provided by your lab hosting provider and then select **Sign in**.
5. In the Search bar of the Azure portal, type *Sentinel*, then select **Azure Sentinel**.
6. Select your Azure Sentinel Workspace.
7. In the Azure Sentinel Workspace, select **Notebooks**.
8. Next, you need to select an AzureML Workspace. Select **Create new AML workspace**.
9. In the Subscription box, select your subscription.
10. Select **Create new** for the Resource group and choose a name for your new resource group.
11. In the Workspace details section do the following:

- Give your workspace a unique name.
- Choose your Region (it should default with a reasonable option)
- Keep the default Storage account, Key vault, and Application insights information.
- The Container registry option can remain as **None**.

12. At the bottom of the page, select **Review + create**. Then on the next page, select **Create**.

Note: It may take a few moments to deploy the workspace.

13. After the deployment is finished. Return the Azure Sentinel portal.
14. Select **Notebooks**.

15. Select **A Getting Started Guide For Azure Sentinel ML Notebooks**, then select **Save notebook**. In the pop-up for the name of your notebook let default and select **OK**.
16. Select the **Launch notebook** button.
17. Next to the **Compute:** instance selector at the top of the screen, select the + symbol for **New Compute**.
18. Choose your compute settings. Then select **Next**.
19. Name your Compute instance and select the **Create** button at the bottom of the screen. This may take a few minutes.
20. Once the Compute has been created, in the top right of the notebook, select a kernel to use.
21. Follow the Getting Started tutorial.

Note If you cannot complete the steps above to access the notebook, you can view it on its GitHub page instead. See the notebook file here:Azure Sentinal Notebooks on GitHub

28.2 You have completed the lab.