Introduction

Objectives:

After completing this lab, you will be able to:
• Deploy Azure Sentinel as a platform for visualizing, investigating, and alerting on your customer's security big data
• Confidently use Sentinel to render your customer's visualized data truly actionable
• Respond to security incidents and Indicators of Compromise discovered with Sentinel
• Proactively hunt for misconfigurations and Indicators of Compromise with hunting queries

Prerequisites:

Before working on this lab, the following helps:
• Working knowledge of SIEM and SOAR technology
• Familiarity with common attack scenarios and techniques
• Familiarity with Azure Playbooks and automation
• Awareness of Kusto Query Language (KQL)

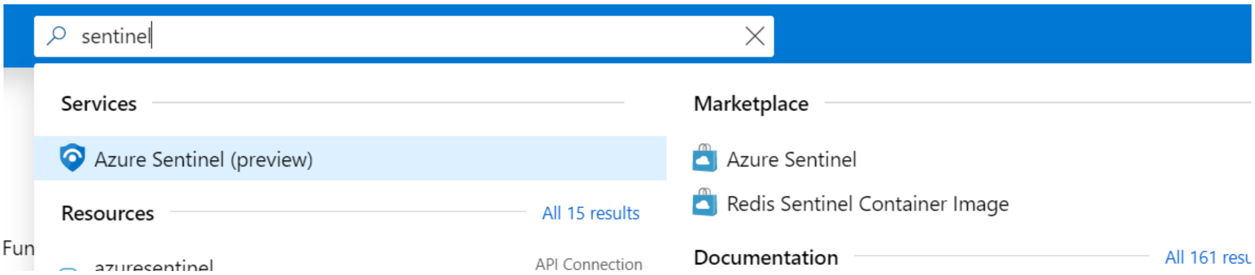
Introduction

Lab 01: Review Azure Sentinel Data Connectors

In this exercise you will be reviewing the data connectors of Azure Sentinel to confirm that data is already being collected in the Sentinel Workspace. Data connectors are critical to a setup of Azure Sentinel so that data can flow into the workspace. To save time, these data connectors have been configured in advance.

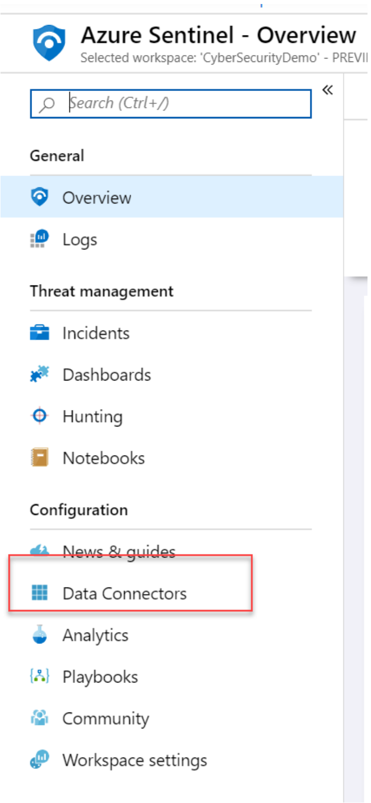
Task 1: Log on to Azure and Navigate to Sentinel
In this task, you will log into the Azure portal and navigate to Sentinel.

1. In a browser, navigate to https://portal.azure.com.
2. In the window that opens, click the account designed for you.
3. This brings you to the Azure Home. In the box at the top, search for **Sentinel**.
4. In the results, click on **Azure Sentinel**.



6. Click in the name of the Sentinel Workspace designed for you or create a new one

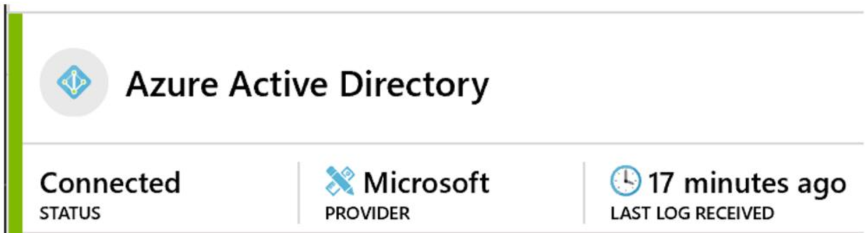
7. In the Navigation bar, select **Data Connectors**



8. Locate the data connectors **Azure Active Directory** and **Office365.** Connect them to your environment.

9. After, Click again **Azure Active Directory** connector and Click **Open connector page**

10. Confirm the status of the connector as **Connected**



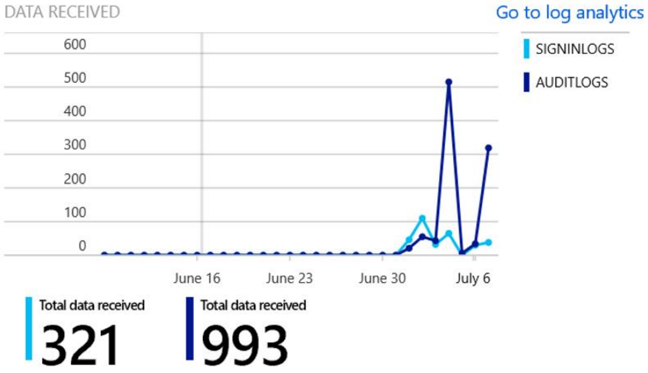
Azure Active Directory

Connected
STATUS

Microsoft
PROVIDER

17 minutes ago
LAST LOG RECEIVED

11. Confirm that there has been recent data received



Continue with the next lab

Lab 2: Setup Alerts (Analytics) and Respond to Incident in Azure Sentinel

In this section we will configure Alerts in Azure Sentinel to generate Incidents for our security team. During the review of these Incident we will build a Playbook to simplify the handling of these alerts.

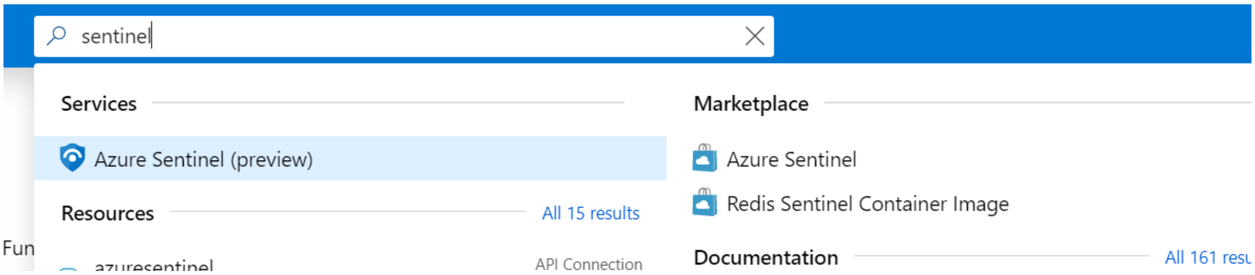
Task 1: Log on to the **My Apps portal** as a **Standard User**

In this task you will logon to the **My Apps portal** as a standard user to confirm the accounts functionality. Afterwards, you will send bad passwords to the account to simulate an unauthorized user attempting to gain access.

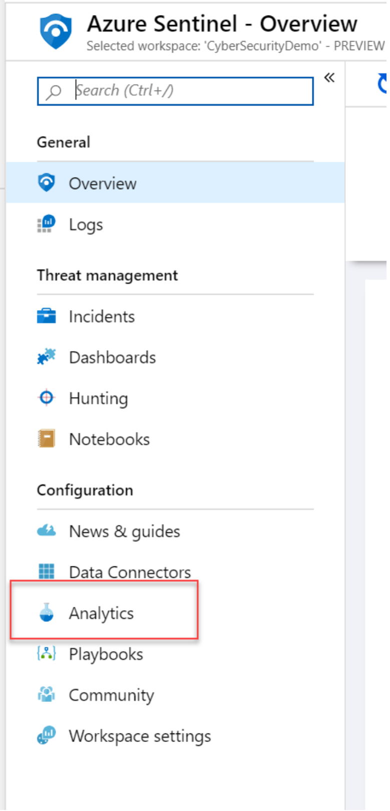
1. In a browser, navigate to https://myapps.microsoft.com
2. Enter your account information
3. Confirm you have access to the My Apps portal.
4. Close all instances of your browser and Navigate back to https://myapps.microsoft.com
5. This time, when logging on as your **user account** type in an **incorrect password**
6. Repeat this invalid logon attempt **5 times**

Task 2: Log on to Azure and Navigate to Sentinel
In this task you will log into the Azure portal and navigate to Sentinel.

1. In a browser, navigate to https://portal.azure.com.
2. In the window that opens, enter your account and click on Azure Sentinel



3. Click the name of your Sentinel Workspace.
4. Within the navigation bar, click **Analytics**



Task 2: Create Analytics Rule for Failed Logon

1. Click **Add**
   In the Name box type: **Failed Logon Alert**
2. Select Alert Severity **Medium**
3. In **Set Alert Query** copy the following query

```
SigninLogs
| where Status.errorCode == 50126 // Invalid Username or password
| where UserPrincipalName contains "standardXX"
| project TimeGenerated, Status.failureReason, UserPrincipalName, Status.errorCode, UserId, IPAddress
| sort by TimeGenerated desc
```

4. Under Entity Mapping, map the following entities

- Account -> UserId (Click **Add**)
- Host -> UserprincipalName (Click **Add**)
- IP address -> IPAddress (Click **Add**)

5. Set the Alert Trigger Operator to **Number of results greater than** and the threshold to 0.
6. Set the Alert **scheduling Frequency** to 5 minutes and Period to 30 minutes
7. Set Alert suppression to On and Suppress alerts for 30 minutes
8. Click **Create**.

Task 3: Create Analytics Rule for Admin Group Change

1. We will Create new Alert
-on the analytics main page click **Add**
In the Name box type **Admin Group Change**
2. Select Alert Severity **Medium**
3. In **Set Alert Query** copy the following query

```
let accttypes = dynamic(['Domain Admins', 'Enterprise Admins', 'Schema Admins', 'Administrators' ,
'Account Operators' , 'Backup Operators' , 'Print Operators' , 'Server Operators' , 'Domain Controllers' ,
'Read-only Domain Controllers' , 'Group Policy Creator Owners' , 'Cryptographic Operators']);
SecurityEvent
| where EventID in (4728, 4729, 4732, 4733, 4756, 4757)
| extend ChangeType = case(EventID in (4729, 4733, 4757), "Member Removed", "Member Added")
| parse EventData with *'"TargetUserName">'TargetUserName'<'*
| where TargetUserName in (accttypes)
| project ChangeType, ImpactedGroup=TargetUserName, ImpactedAccount=MemberName,
ChangeMaker=Account, DC=Computer
```

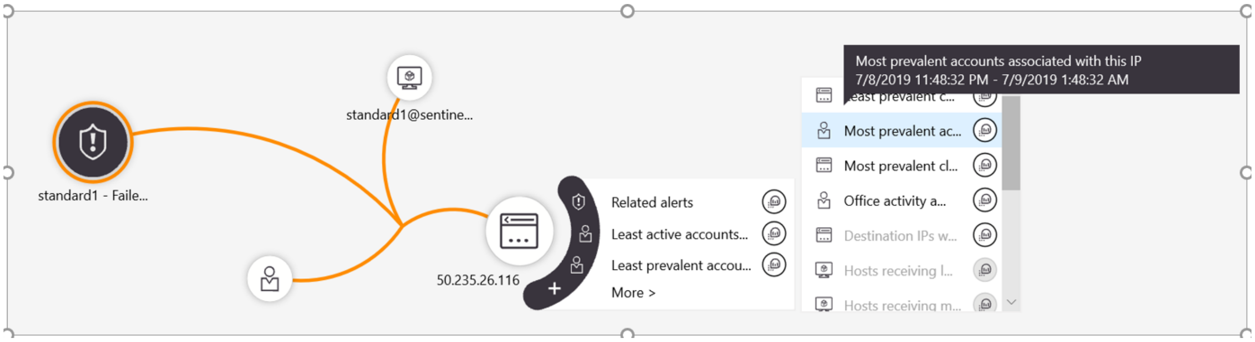
4. Under Entity Mapping, map the following entities

- Account -> ImpactedAccount (Click Add)
- Host -> ImpactedGroup (Click Add)

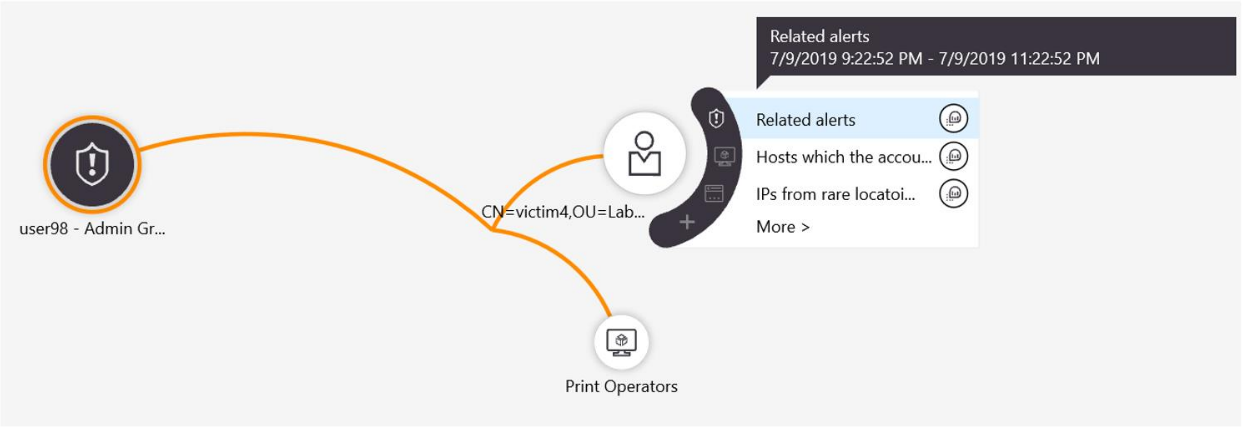
5. Set the Alert Trigger Operator to **Number of results greater than** and the threshold to 0
6. Set the Alert **scheduling Frequency** to 5 minutes and Period to 1 hour
7. Set **Alert suppression** to On and Suppress alerts for 1 hour

Task 4: Review your Incidents

1. Try on your tenant generate incidents related to both Analytics you've just created
2. Locate your Incidents that generated from the first rule that you created **Failed Logon**
3. Click on your Incidents and Click the **Investigate** button
4. Within the Investigation window hover over the IP Address node and select **More -> Most prevalent accounts associated with this IP** to bring in additional data related to that IP



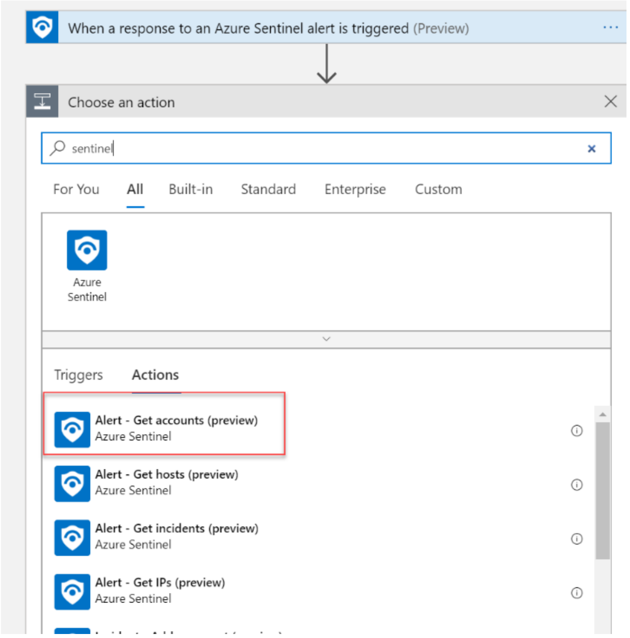
6. Then, hover once again over the IP address and click **Related alerts** to see other Sentinel alerts based on the same IP Address to help identify other problems related to this bad logon
7. Navigate back to **Azure Sentinel** -> **Incidents** and locate the Incident **Admin Group Change**
8. Click on your Incident and Click the **Investigate** button
9. Within the investigation window you can see the **User that was added to the group**, and the Group that they were added to.
10. In this example bellow, the **Related alerts** shows additional alerts related to this Admin group change involving this account.



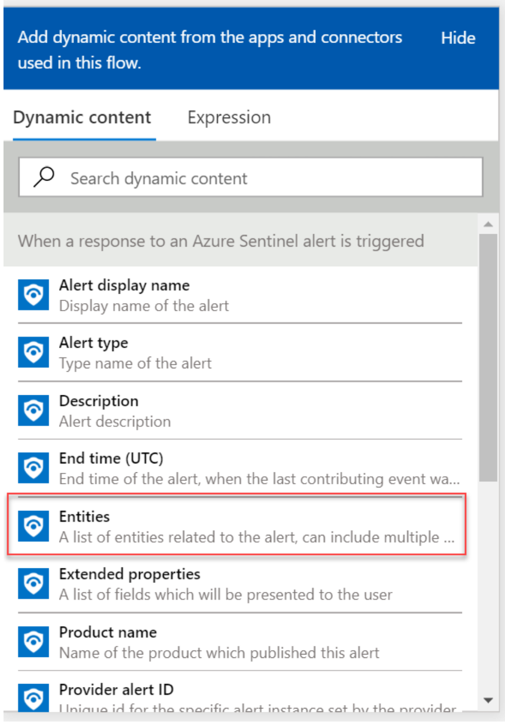
Task 5: Create a playbook to respond to your Incidents

In this playbook we will be adding a user into a group based on Incidents. This group is already linked to a conditional access policy which will block user logons to all cloud applications.

1. Within Azure Sentinel, Navigate to **Playbooks**
2. Click **Add Playbook**
3. In the Name field type **Block-With-Conditional-Access-Policy**
4. Under Resource Group, Select your own or otherwise **create a new one**
5. Click **Create**
6. In the Logic Apps Designer, Click **Blank Logic App**
7. In the text box **Search connectors and triggers** type **Sentinel** and click on the icon for **Azure Sentinel**.
8. Select **When a response to an Azure Sentinel alert is triggered**
9. Click **New Step**
10. Search for **Sentinel** and click on the **Azure Sentinel** icon and click **Get Accounts (preview)**


When a response to an Azure Sentinel alert is triggered (Preview)
Choose an action
sentinel
For You
All
Built-in
Standard
Enterprise
Custom
Azure Sentinel
Triggers
Actions
Alert - Get accounts (preview)
Azure Sentinel
Alert - Get hosts (preview)
Azure Sentinel
Alert - Get incidents (preview)
Azure Sentinel
Alert - Get IPs (preview)
Azure Sentinel

11. Click the text box **Entities List** and from the flyout, select **Entities**



Add dynamic content from the apps and connectors used in this flow.
Hide
Dynamic content
Expression
Search dynamic content

When a response to an Azure Sentinel alert is triggered

Alert display name
Display name of the alert

Alert type
Type name of the alert

Description
Alert description

End time (UTC)
End time of the alert, when the last contributing event wa...

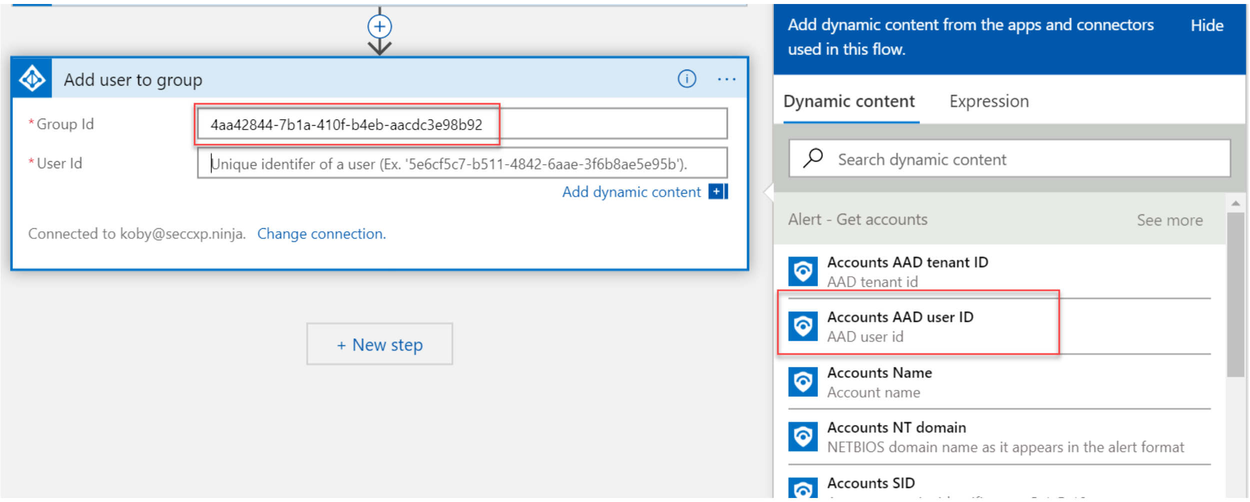
Entities
A list of entities related to the alert, can include multiple ...

Extended properties
A list of fields which will be presented to the user

Product name
Name of the product which published this alert

Provider alert ID
Unique id for the specific alert instance set by the provider

12. Click **New Step** and search for **Azure AD**
13. Select **Add user to group**
14. Under **Group ID**, enter 4aa42844-7b1a-410f-b4eb-aacdc3e98b92 *(This number is already mapped to an exisiting group)*
15. Under **User ID,** select **Accounts AAD user ID** from the flyout



* Group Id: 4aa42844-7b1a-410f-b4eb-aacdc3e98b92

* User Id: Unique identifer of a user (Ex. '5e6cf5c7-b511-4842-6aae-3f6b8ae5e95b').

Connected to koby@seccxp.ninja.

Alert - Get accounts

Accounts AAD tenant ID
AAD tenant id

Accounts AAD user ID
AAD user id

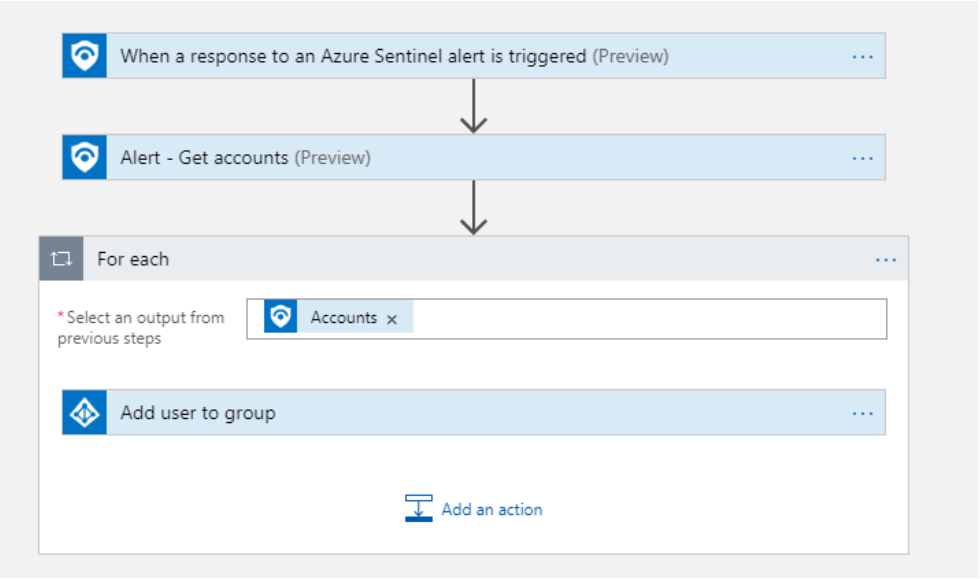
Accounts Name
Account name

Accounts NT domain
NETBIOS domain name as it appears in the alert format

Accounts SID

16. Click **Save**

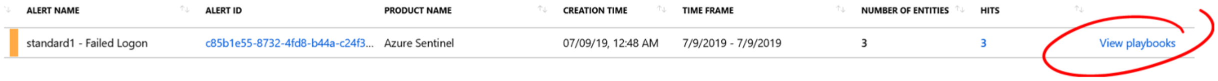
- This is how it should look


When a response to an Azure Sentinel alert is triggered (Preview)
Alert - Get accounts (Preview)
For each
Add user to group
Add an action

*Select an output from previous steps: Accounts

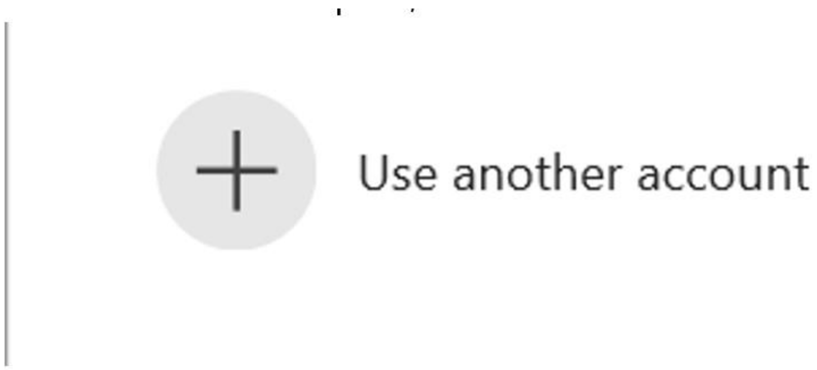
Task 6: Use your new playbook to respond to an Incidents

1. Navigate back to **Sentinel -> Incidents** in the Azure Portal
2. Click on your Incidents **Failed Logon**
3. Click **View full details**
4. Under the Incidents, click **View playbooks**

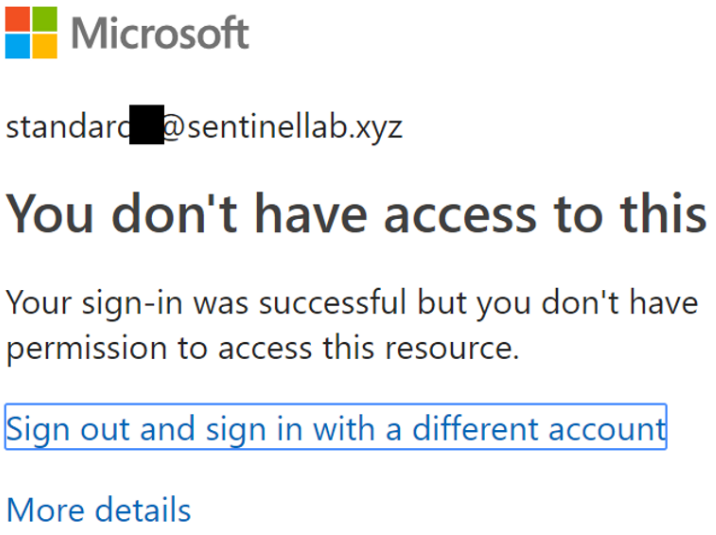


ALERT NAME	ALERT ID	PRODUCT NAME	CREATION TIME	TIME FRAME	NUMBER OF ENTITIES	HITS	
standard1 - Failed Logon	c85b1e55-8732-4fd8-b44a-c24f3...	Azure Sentinel	07/09/19, 12:48 AM	7/9/2019 - 7/9/2019	3	3	View playbooks

5. From the list of Playbooks locate **Block-With-Conditional-Access-Policy** and click **Run**
6. Close all browser windows and open a new browser
7. Navigate to https://myapps.microsoft.com
8. In the window that opens, click **Use another account**.



9. Enter the account information of a standar, regular user, like: standardXX@sentinellab.xyz (this account should be created first!)
10. You will now see that access to the my apps portal has been blocked due to conditional access


Microsoft

standard[redacted]@sentinellab.xyz

You don't have access to this

Your sign-in was successful but you don't have permission to access this resource.

Lab 3: Proactively Investigate Potential Threats, Misconfigurations, and Suspicious Activities Visually

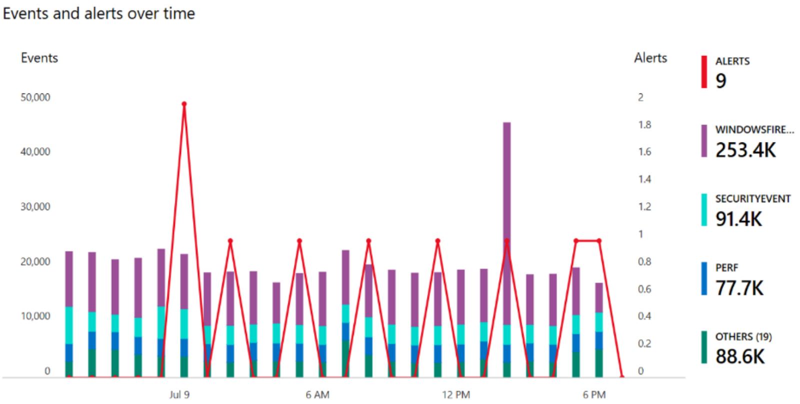
In this exercise, you will learn how to access Azure Sentinel and how to use a Dashboard to view data visually. You will also learn how to examine the underlying Kusto queries, and to change them. For this lab, Data connectors have already been added and appropriately configured as necessary.

Task 1: Log on to Azure and Navigate to Sentinel In this task, you will log into the Azure portal and navigate to Sentinel.

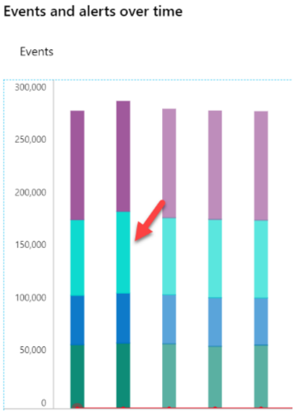
1. In a browser, navigate to https://portal.azure.com and log in Sentinel dashboard with your account
2. Take a moment to examine the Sentinel **Overview** page. This page gives you a bird's eye view of your estate's data.

Task 2: Navigate the Overview Page and Understand Basic Kusto Queries*

1. Look closely at the pane entitled **Events and alerts over time**. This pane provides a bar graph representing alerts, Windows Firewall Events, Security Events, Performance data, and other data that has been recently gathered.



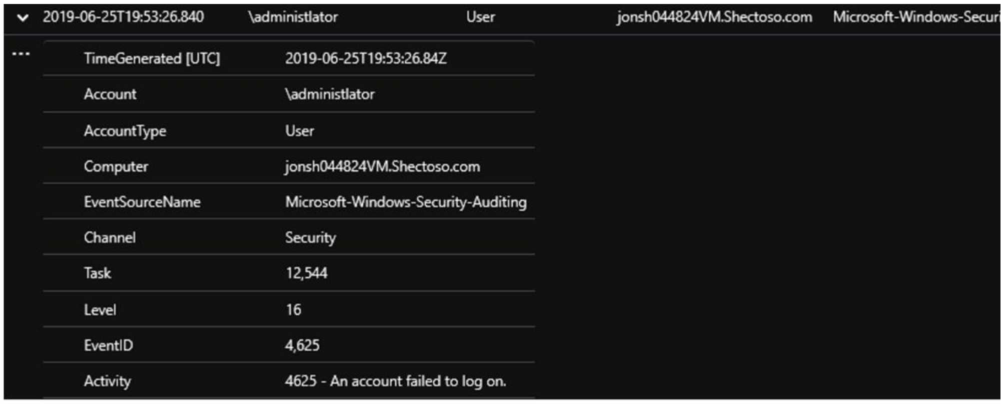
6. Return to the Sentinel Overview page by clicking **Azure Sentinel – Overview** at the top.
7. Click one of the **Security Event** areas of one of the bars – it appears in light green(The Bar color may change).



9. This query is slightly more complex and will look something like this *(the date and time fields will be different)*:

```
union SecurityEvent
| where TimeGenerated >= datetime(2019-08-25T16:19:04.063Z) and TimeGenerated < datetime(2019-08-25T16:19:04.063Z) + 1h
```

10. This query is asking for a **SecurityEvent** (from a Windows security log) during a specified date and time.
11. Click on one of the chevrons, preferably for an **AccountType** of **User** and not a built-in account like NT AUTHORITY.



```
⌄ 2019-06-25T19:53:26.840    \administlator    User    jonsh044824VM.Shectoso.com    Microsoft-Windows-Secur
```

TimeGenerated [UTC]	2019-06-25T19:53:26.84Z
Account	\administlator
AccountType	User
Computer	jonsh044824VM.Shectoso.com
EventSourceName	Microsoft-Windows-Security-Auditing
Channel	Security
Task	12,544
Level	16
EventID	4,625
Activity	4625 - An account failed to log on.

In this example, the administrator account has failed to log on. This could be very telling as an early indicator of attack. Why is someone logging on (and failing) using the administrator account?

12. Return to the Sentinel Overview page.

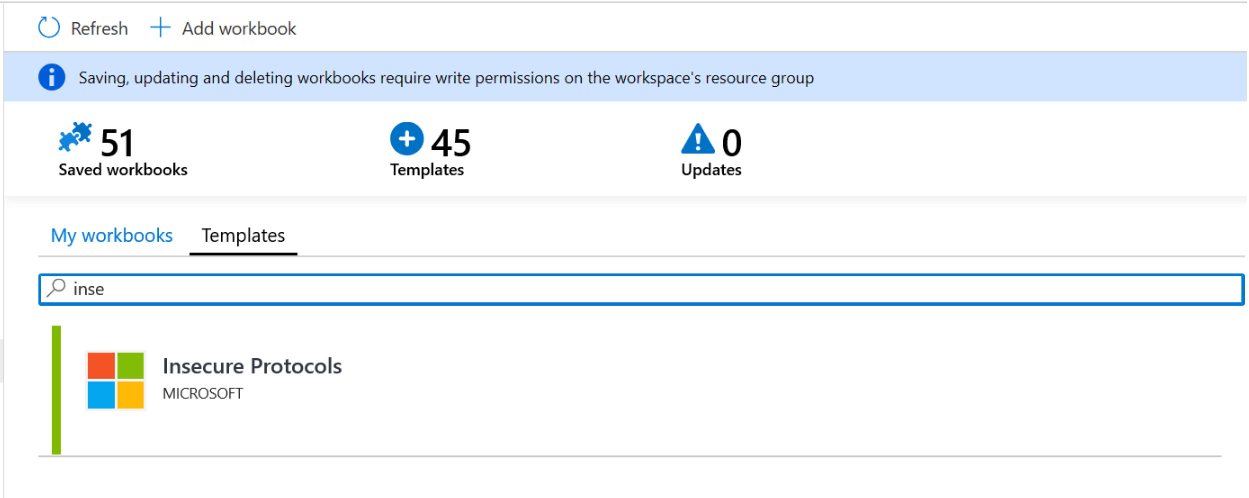
Task 3: Utilize Dashboards to Proactively Investigate Potential Threats, Misconfigurations, and Suspicious Activities

Step 1: Learn to navigate the Workbooks (Dashboards) frame

In this task, you will learn to proactively investigate using Sentinel Dashboards. Sentinel Workbook provide visualized data from your connected data sources to help render them proactively actionable.

1. From the Sentinel Overview page, click on **Workbooks**.

2. On the next blade that appears, click Template.



Refresh + Add workbook

Saving, updating and deleting workbooks require write permissions on the workspace's resource group

51 Saved workbooks

45 Templates

0 Updates

My workbooks Templates

inse

Insecure Protocols
MICROSOFT

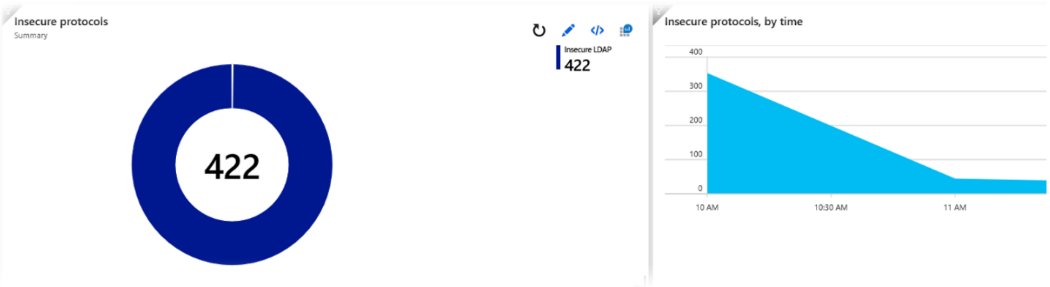
3. Take a moment to browse the list of built-in Workbooks such as **Azure Activity, Azure AD Sign-in logs, Office 365**, and third-party Workbooks for products from vendors like Symantec and Palo Alto. It is important to understand that **Workbooks** are typically used with **Data connectors**.

4. For this exercise, we'll be using the **Insecure Protocols** built-in dashboard.
5. From the **Workbooks** blade, click on **Templates**.

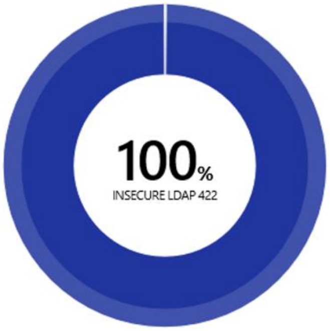
6. Click on **Insecure Protocols**. In the lower right-hand corner, click **Create* (or Save)** in order to bring up a new the dashboard.
7. The **Insecure Protocols** dashboard visually represents Security log data from on-premise or IaaS Active Directory Domain Controllers. Specifically, it represents data detailing which insecure protocols are currently in use and their data flows throughout the estate.
8. Take a moment to browse the dashboard. This dashboard addresses the problem of organizations that wish to remove insecure protocols (like NTLMv1, SMB1, wDigest and weak Kerberos ciphers) but are unable to do so for fear of breaking critical business systems. They need to remediate the sources before they can disable the protocols. And for that they need data visualization.

Step 2: Dive deeper in the Insecure Protocols Dashboard to Investigate Security Data

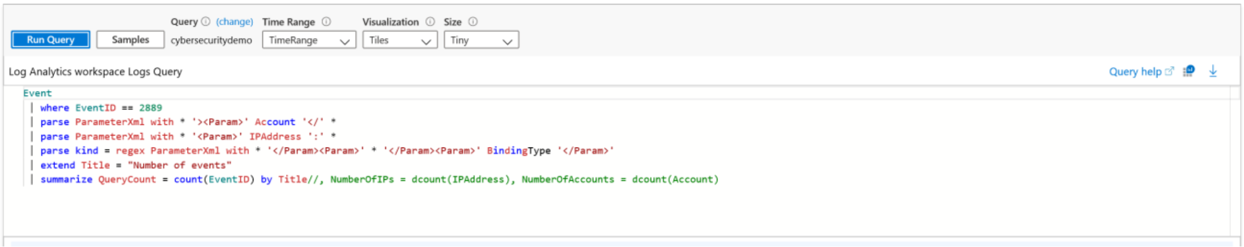
1. Let's take a look at the overview graph titled **Insecure Protocols**.



2. On the pie chart, we can immediately see something important. Out of all the Insecure Protocols that could be running in this estate, only Insecure LDAP shows up. That tell us that, so long as this trend continues, we could safely turn off other protocols such as NTLMv1 and SMB1. See as example bellow: 100% of the Insecure Protocols in the environment are **Unsigned LDAP**.



3. We can also drop down to the Insecure LDAP section of the page (directly below the Overview) for more insightful data representations.
4. On the **Insecure Protocols** workbook, scroll down to **Unsigned LDA**P**. This was the most active Insecure Protocol in the estate, so it makes sense to examine it closely.
5. Note that we can drill down in a number of ways – by **Client, Time**, and by **Details**.
6. Sometimes we will need to assist our customers in delving deeper into a particular data set or sets. In order to do that, we will often want to drill down closer to the actual data collected by Sentinel (Log Analytics on the back end). Let's do this for this visualizer, **Unsigned LDAP** by details.
7. Click on the Edit button, right bellow **Unsigned LDAP** text
8. This brings up the Kusto Query Language (KQL) query that is used to populate the visualizer. Peruse the query.

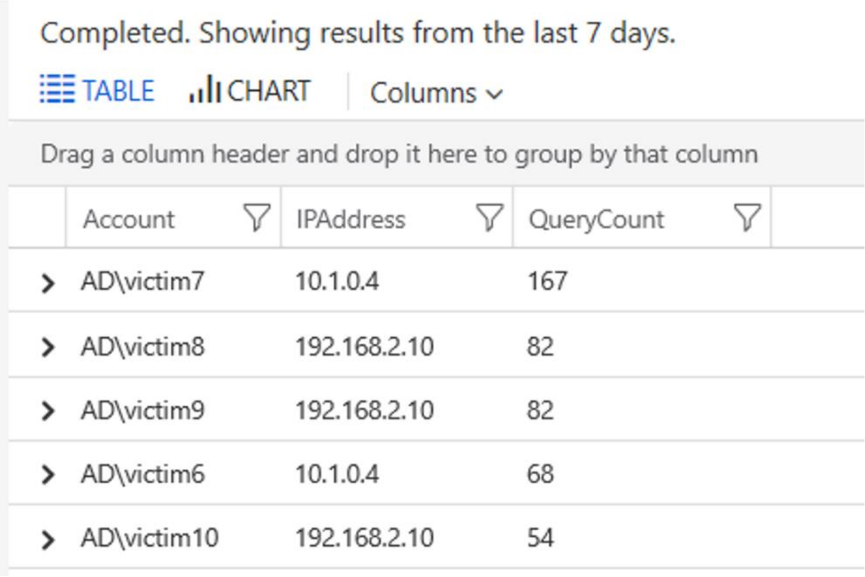


Query (change) cybersecuritydemo
Time Range TimeRange
Visualization Tiles
Size Tiny

Log Analytics workspace Logs Query

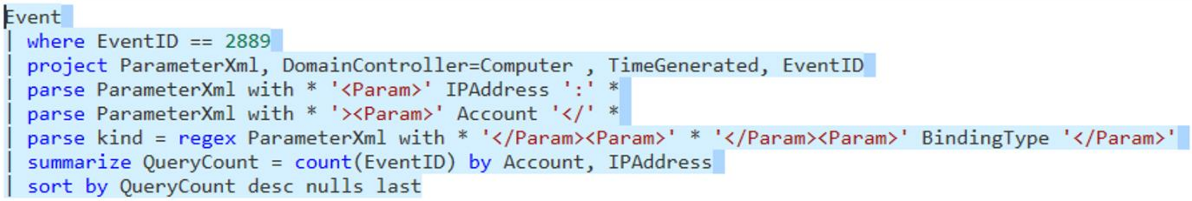
```
Event
 | where EventID == 2889
 | parse ParameterXml with * '><Param>' Account '</' *
 | parse ParameterXml with * '<Param>' IPAddress ':' *
 | parse kind = regex ParameterXml with * '</Param><Param>' * '</Param><Param>' BindingType '</Param>'
 | extend Title = "Number of events"
 | summarize QueryCount = count(EventID) by Title//, NumberOfIPs = dcount(IPAddress), NumberOfAccounts = dcount(Account)
```

9. This query is asking for logs with EventID 2889 (unsigned LDAP bind) and is doing some extractions as well as RegEx matching to put the data in an acceptable format. It is summarizing the data by count (Account, IPAddress) and sorting it in descending order. We can see this in the data that is returned.


Completed. Showing results from the last 7 days.
TABLE
CHART
Columns
Drag a column header and drop it here to group by that column

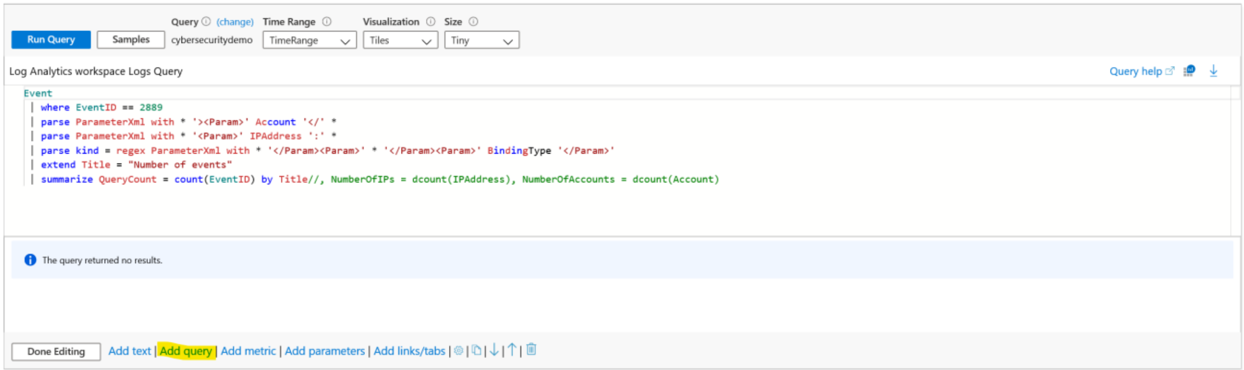
Account	IPAddress	QueryCount
AD\victim7	10.1.0.4	167
AD\victim8	192.168.2.10	82
AD\victim9	192.168.2.10	82
AD\victim6	10.1.0.4	68
AD\victim10	192.168.2.10	54

10. We see now that we have an actionable set of data. We know the accounts and IPAddresses that are issuing Insecure LDAP binds. We know where we need to look in order to remediate the problem and ultimately to disallow Unsigned LDAP throughout our estate.
11. Let's say we needed to go a bit deeper. For example, maybe we need to know the Domain Controller being targeted by the Insecure LDAP binds. We could easily do that in the following steps.
12. Highlight the whole query, right click and **Copy.**



```
Event
| where EventID == 2889
| project ParameterXml, DomainController=Computer , TimeGenerated, EventID
| parse ParameterXml with * '<Param>' IPAddress ':' *
| parse ParameterXml with * '><Param>' Account '</' *
| parse kind = regex ParameterXml with * '</Param><Param>' * '</Param><Param>' BindingType '</Param>'
| summarize QueryCount = count(EventID) by Account, IPAddress
| sort by QueryCount desc nulls last
```

At the bottom, click the **Add query** to open a new Query window.



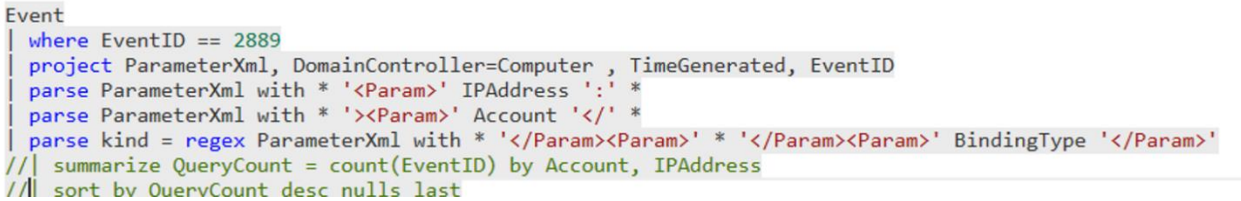
```
Event
| where EventID == 2889
| parse ParameterXml with * '><Param>' Account '</' *
| parse ParameterXml with * '<Param>' IPAddress ':' *
| parse kind = regex ParameterXml with * '</Param><Param>' * '</Param><Param>' BindingType '</Param>'
| extend Title = "Number of events"
| summarize QueryCount = count(EventID) by Title//, NumberOfIPs = dcount(IPAddress), NumberOfAccounts = dcount(Account)
```

The query returned no results.

Done Editing
Add text | Add query | Add metric | Add parameters | Add links/tabs

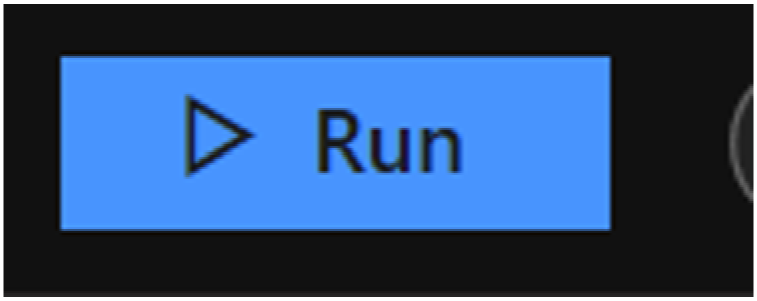
13. In the window that opens, Paste the copied query.

a. Comment the last two lines using backslashes, like this:

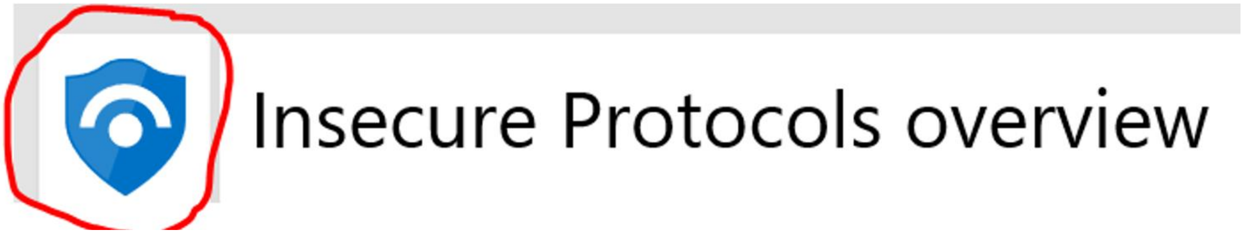


```
Event
| where EventID == 2889
| project ParameterXml, DomainController=Computer , TimeGenerated, EventID
| parse ParameterXml with * '<Param>' IPAddress ':' *
| parse ParameterXml with * '><Param>' Account '</' *
| parse kind = regex ParameterXml with * '</Param><Param>' * '</Param><Param>' BindingType '</Param>'
//| summarize QueryCount = count(EventID) by Account, IPAddress
//| sort by QueryCount desc nulls last
```

14. Change the **Time range** to **Last 7 days** and click Run



15. This returns the results that are not grouped – closer to what you might think of as "raw" log data. Click one of the chevrons to the left side of the screen in order to examine one of these logs.
16. This type of data can help fill in the missing pieces to our story. Depending on which log you clicked, you can gain the missing information.
17. Return to your initial Insecure Protocols workbook.

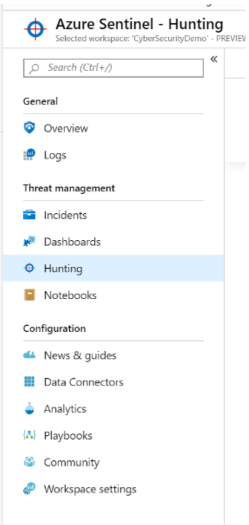


18. This returns us to the Azure Sentinel homepage.

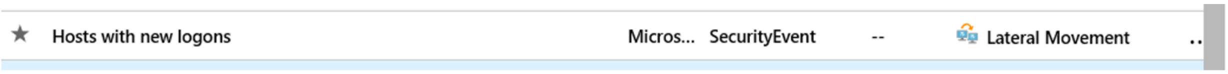
Exercise 4: Proactively Investigate Potential Threats, Misconfigurations, and Suspicious Activities with KQL Queries

Task 1: Navigate to Sentinel Hunting and Run all queries
As in Task 1, we will be proactively hunting for Indicators of Attack in Sentinel. In that task we were doing so visually through the use of dashboards. Here, we will use built-in KQL queries to identify signs of potential problems.

1. From the Azure Sentinel homepage, click on **Hunting.**
2. In the pane that opens, click on Hunting.

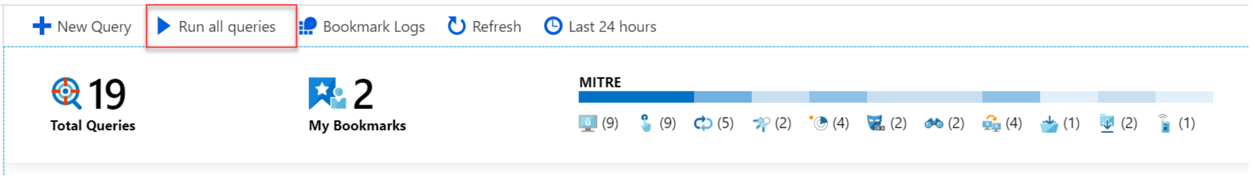


3. Examine the Queries in the middle pane. These are built-in KQL queries provided to help you hunt for adversaries in your environment. Pay attention to the description and also special attention to the Tactics column. You may learn more about a particular tactic by mousing over it for a description. This column helps you understand which part of the attacker kill chain you may insert yourself in with the query. For example, by examining Hosts with new logons, you may be able to detect Lateral Movement in the environment.



★ Hosts with new logons	Micros...	SecurityEvent	--	Lateral Movement	..

4. By running all the built-in queries, you have a quick indicator of specific areas of concern. These may form a ready-made map of where your time should be spent hunting in any given point in time.
5. Click **Run all queries.**



6. Look at the Results column. Note that several of these queries have returned results. These require further investigation. You will want to help your customers pay special investigative attention to the results of queries like New processes observed in last 24 hours, and uncommon processes – bottom 5%. These can be indicators of compromise that are often missed.

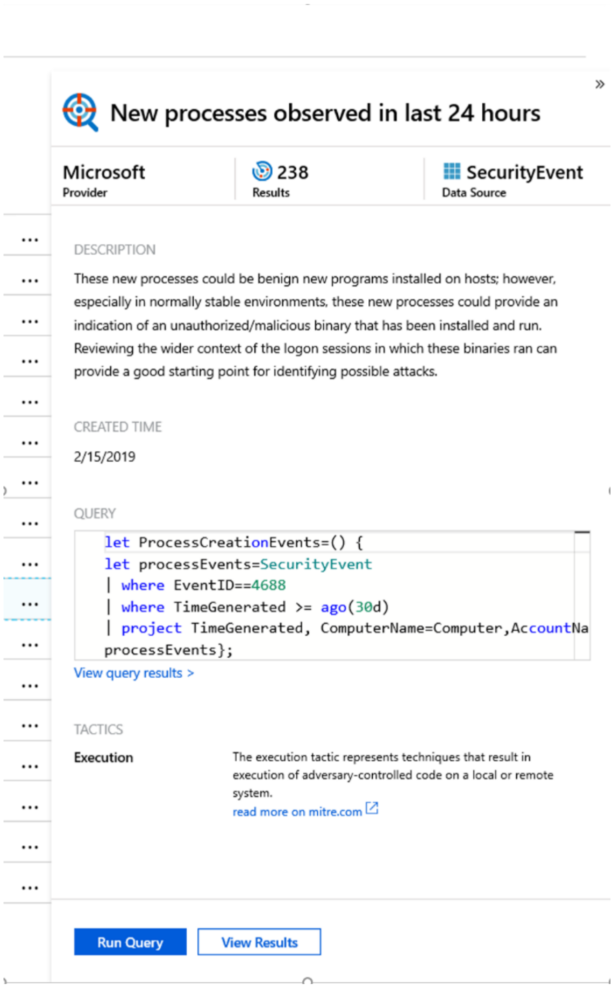
Task 2. Run Several Single Hunting Queries and Examine the Output in Detail

In this Task, we will run several single queries to hunt for indicators of compromise.

Step 1: Run the New processes observed Query and Examine the Output

1. Click on the query New processes observed in the last 24 hours.

2. This brings up information about this Hunting query on the right side of the screen.



New processes observed in last 24 hours

Microsoft
Provider

238
Results

SecurityEvent
Data Source

DESCRIPTION

These new processes could be benign new programs installed on hosts; however, especially in normally stable environments, these new processes could provide an indication of an unauthorized/malicious binary that has been installed and run. Reviewing the wider context of the logon sessions in which these binaries ran can provide a good starting point for identifying possible attacks.

CREATED TIME

2/15/2019

QUERY

```
let ProcessCreationEvents=() {
let processEvents=SecurityEvent
| where EventID==4688
| where TimeGenerated >= ago(30d)
| project TimeGenerated, ComputerName=Computer,AccountNa
processEvents};
```

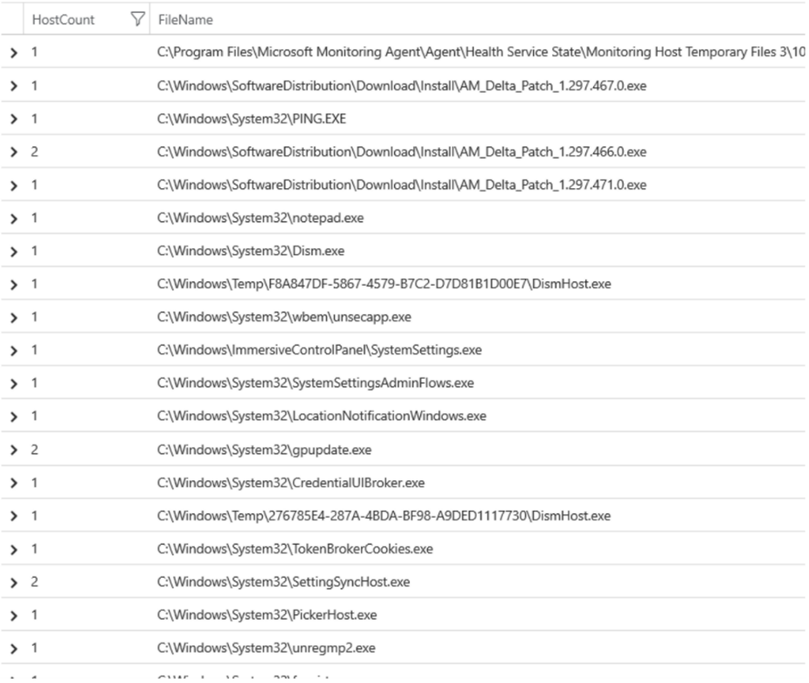
View query results >

TACTICS

Execution

The execution tactic represents techniques that result in execution of adversary-controlled code on a local or remote system.
read more on mitre.com

3. **Optional:** for background information, you may click on read more on mitre.com. This is an excellent knowledge base of adversary tactics and techniques that have been documented in the wild.
4. Click **Run Query** and then click **View Results**. This launches the Logs page, pre-populates the KQL query, and displays the results. Follows bellow one example of such results given by this **Hunting**

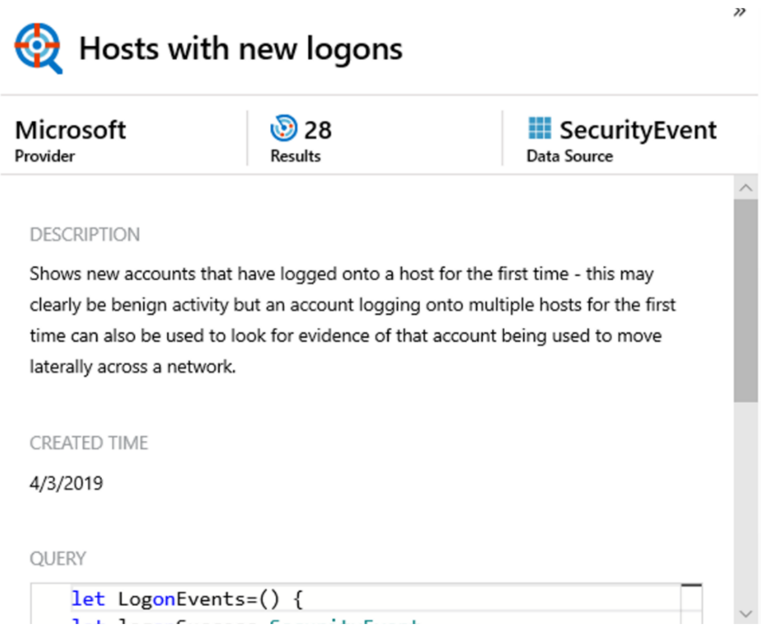


HostCount	FileName
1	C:\Program Files\Microsoft Monitoring Agent\Agent\Health Service State\Monitoring Host Temporary Files 3\10
1	C:\Windows\SoftwareDistribution\Download\Install\AM_Delta_Patch_1.297.467.0.exe
1	C:\Windows\System32\PING.EXE
2	C:\Windows\SoftwareDistribution\Download\Install\AM_Delta_Patch_1.297.466.0.exe
1	C:\Windows\SoftwareDistribution\Download\Install\AM_Delta_Patch_1.297.471.0.exe
1	C:\Windows\System32\notepad.exe
1	C:\Windows\System32\Dism.exe
1	C:\Windows\Temp\F8A847DF-5867-4579-B7C2-D7D81B1D00E7\DismHost.exe
1	C:\Windows\System32\wbem\unsecapp.exe
1	C:\Windows\ImmersiveControlPanel\SystemSettings.exe
1	C:\Windows\System32\SystemSettingsAdminFlows.exe
1	C:\Windows\System32\LocationNotificationWindows.exe
2	C:\Windows\System32\gpupdate.exe
1	C:\Windows\System32\CredentialUIBroker.exe
1	C:\Windows\Temp\276785E4-287A-4BDA-BF98-A9DED1117730\DismHost.exe
1	C:\Windows\System32\TokenBrokerCookies.exe
2	C:\Windows\System32\SettingSyncHost.exe
1	C:\Windows\System32\PickerHost.exe
1	C:\Windows\System32\unregmp2.exe
1	C:\Windows\System32\[illegible]

5. Briefly examine the table of results. Notice that it features **HostCount** and **FileName** columns of information. From the results, we can see that there have been a number of new processes launched across the estate in the last 24 hours.
6. This may be a sign of normal, benign programs being run. However, in normally stable environments, these new processes could provide an indication of an unauthorized/malicious binary that has been installed and run. Depending on the environment, Certainly, we need to investigate further.

Step 2: Run the Hosts with new logons Query and Examine the Output

1. Click on the query **Hosts with new logons**.
2. This brings up information about this Hunting query on the right side of the screen



Hosts with new logons

Microsoft
Provider

28
Results

SecurityEvent
Data Source

DESCRIPTION

Shows new accounts that have logged onto a host for the first time - this may clearly be benign activity but an account logging onto multiple hosts for the first time can also be used to look for evidence of that account being used to move laterally across a network.

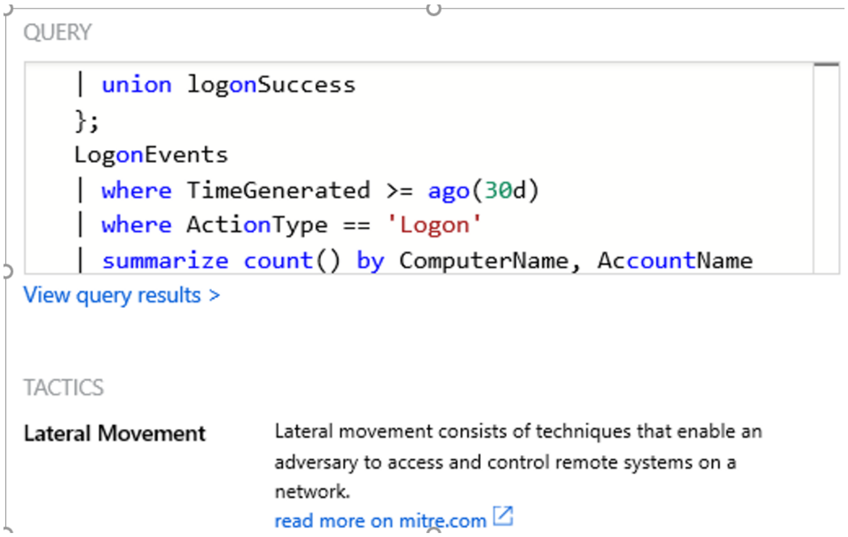
CREATED TIME

4/3/2019

QUERY

```
   let LogonEvents=() {
```

3. Notice the description.
4. Scroll down to view the query and the tactics.



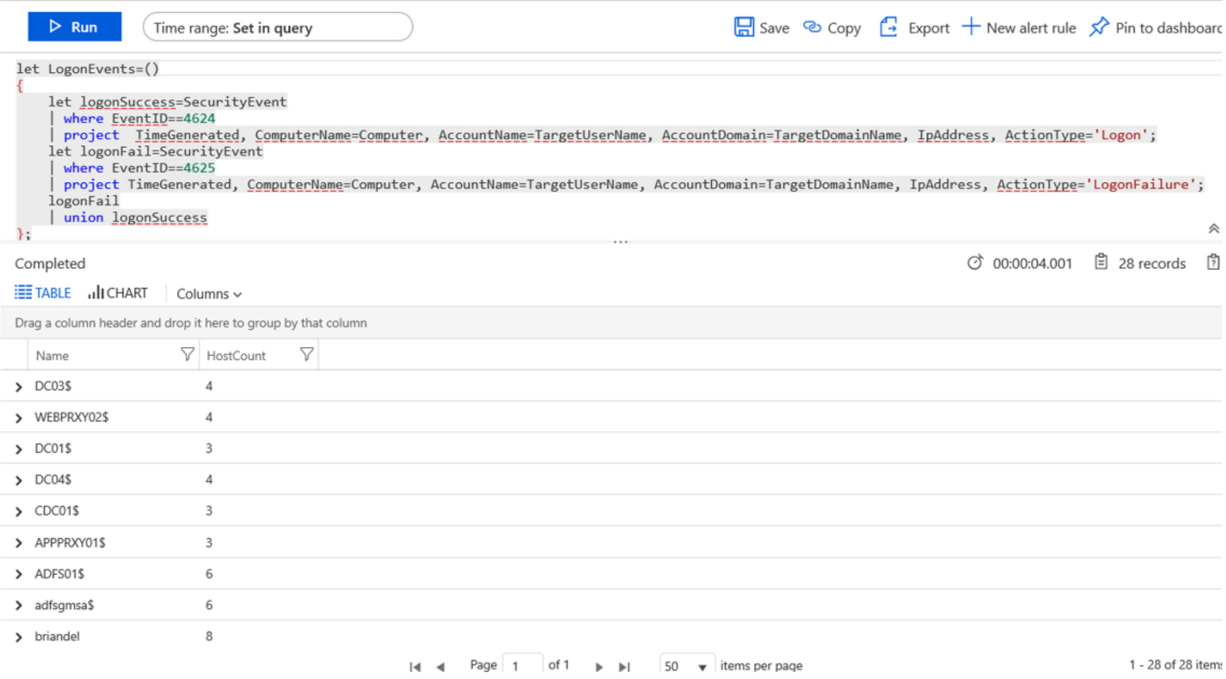
QUERY

```
    | union logonSuccess
    };
    LogonEvents
    | where TimeGenerated >= ago(30d)
    | where ActionType == 'Logon'
    | summarize count() by ComputerName, AccountName
```

TACTICS

Lateral Movement Lateral movement consists of techniques that enable an adversary to access and control remote systems on a network.

5. **Optional**: for background information, you may click on read more on mitre.com. This is a knowledge base of adversary tactics and techniques that have been documented in the wild.
6. Click **Run Query** and then click **View Results.** This launches the Logs page, pre-populates the KQL query, and displays the results, as you can see in this example bellow:



```
let LogonEvents=()
{
    let logonSuccess=SecurityEvent
    | where EventID==4624
    | project  TimeGenerated, ComputerName=Computer, AccountName=TargetUserName, AccountDomain=TargetDomainName, IpAddress, ActionType='Logon';
    let logonFail=SecurityEvent
    | where EventID==4625
    | project TimeGenerated, ComputerName=Computer, AccountName=TargetUserName, AccountDomain=TargetDomainName, IpAddress, ActionType='LogonFailure';
    logonFail
    | union logonSuccess
};
```

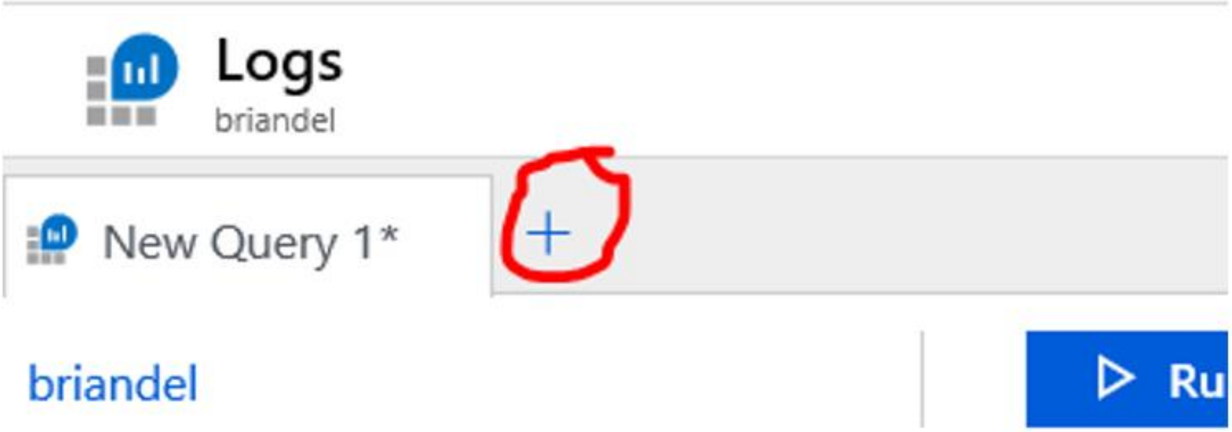
Completed
00:00:04.001
28 records
TABLE
CHART
Columns
Drag a column header and drop it here to group by that column

Name	HostCount
DC03$	4
WEBPRXY02$	4
DC01$	3
DC04$	4
CDC01$	3
APPPRXY01$	3
ADFS01$	6
adfsgmsa$	6
briandel	8

7. Briefly examine the query. The query is asking for successful and failed logon events, creating a union, setting some time parameters, joining several tables, and setting some output parameters.
8. Briefly examine the table of results. Notice that it features the projected attributes Name and Hostcount. From the results, we can see that a subset of computers have indeed received recent (new) logon activity.
9. Your output may look slightly different, but right away we might notice some suspicious activity. Multiple domain controllers (DCs) have received new logons. This bears further investigation. New logons to DCs are typically rare; DCs also represent high-value attack targets.

Task 3: Craft and Execute a Hunting Query to Investigate the New DC Logons

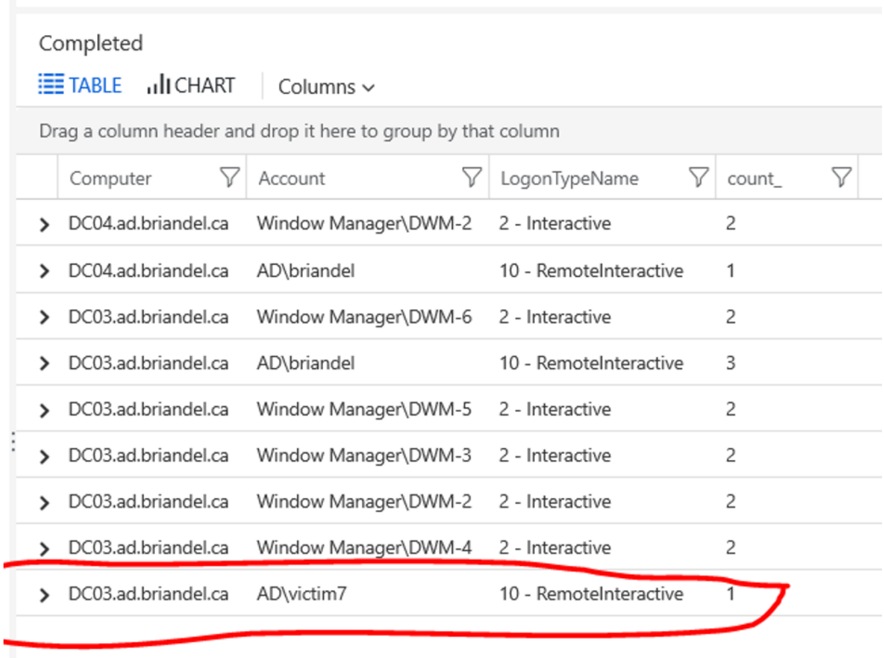
1. In this Task, we want to investigate the accounts that have logged onto the DCs in the last day.
2. Click the **Plus** (+) sign next to New Query 1.



3. In the new query window, copy the text of this query to investigate the accounts that have logged onto the DCs:

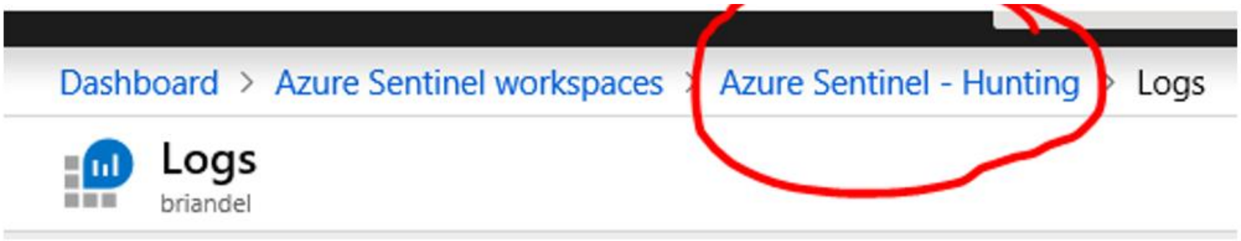
```
SecurityEvent
| where EventID == 4624
| where TimeGenerated >= ago(3d)
| where Computer startswith "DC01" or Computer startswith "DC03" or Computer startswith "DC04"
| where Account !endswith "$" and Account != @"NT AUTHORITY\LOCAL SERVICE" and Account !=
@"NT AUTHORITY\SYSTEM"
| summarize count() by Computer, Account, LogonTypeName
```

4. Examine the output. Here you would hunt for any Accounts that you would not expect to log onto DCs. For example, an account that did not belong to a member of the IT staff (and specifically an Active Directory administrator) would throw a red flag to take action. You could also additionally hunt for discrepancies in logon times, for example, by modifying the query in various ways.
5. In the query results, do you see any accounts that you would flag with your customer? In the example below, we see an account called victim7 having logged into DC03. That is definitely an indicator of compromise. We would need, in this case, to take action against both the account and potentially inside of Active Directory. We could even be in an incident response type of scenario.


Completed
TABLE
CHART
Columns
Drag a column header and drop it here to group by that column

Computer	Account	LogonTypeName	count_
DC04.ad.briandel.ca	Window Manager\DWM-2	2 - Interactive	2
DC04.ad.briandel.ca	AD\briandel	10 - RemoteInteractive	1
DC03.ad.briandel.ca	Window Manager\DWM-6	2 - Interactive	2
DC03.ad.briandel.ca	AD\briandel	10 - RemoteInteractive	3
DC03.ad.briandel.ca	Window Manager\DWM-5	2 - Interactive	2
DC03.ad.briandel.ca	Window Manager\DWM-3	2 - Interactive	2
DC03.ad.briandel.ca	Window Manager\DWM-2	2 - Interactive	2
DC03.ad.briandel.ca	Window Manager\DWM-4	2 - Interactive	2
DC03.ad.briandel.ca	AD\victim7	10 - RemoteInteractive	1

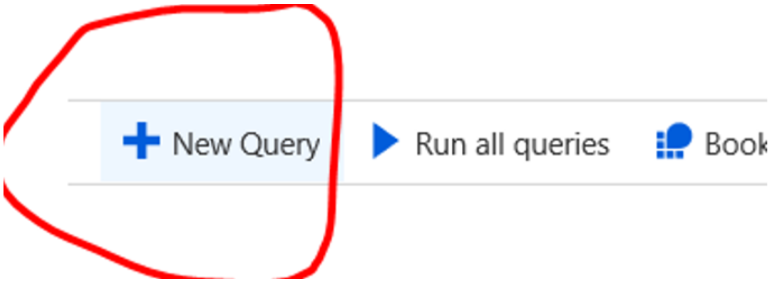
6. At the top of the page, click **Azure Sentinel – Hunting** to return to the Hunting pane.


Dashboard > Azure Sentinel workspaces > Azure Sentinel - Hunting > Logs
Logs
briandel

Task 4: Craft and Execute a Unique Hunting Query to Investigate a Distinct Security Concern

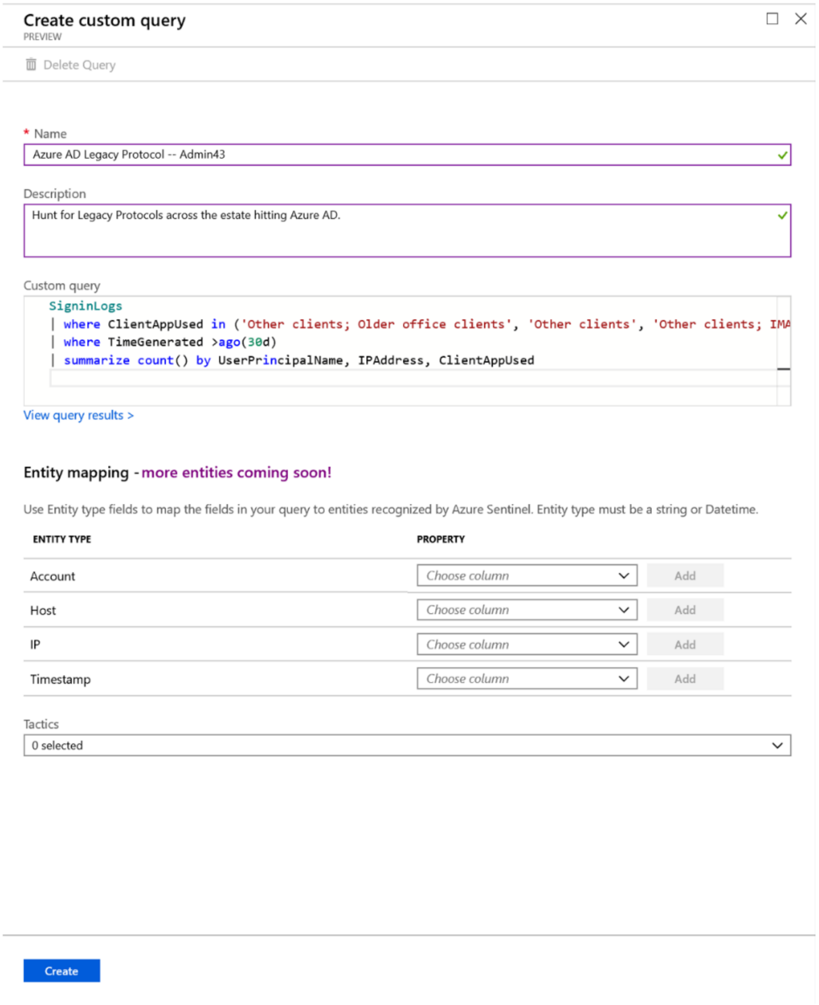
You will craft a query to hunt for the use of Legacy Protocols against Azure AD. One source of legacy protocols is often, but not exclusively, the use of mail browsers accessing Office 365/Exchange Online. In reality, you could make this a workbook, or an extension of the Insecure Protocols dashboard. But in the interest of time, here you will craft this activity as a Hunting Query.

1. From within the **Hunting** section of the portal, Click on the + **New Query** button



2. On the Create custom query page, Fill in the following information
   Name: Azure AD Legacy Protocol
   Description: Hunt for Legacy Protocols across the estate hitting Azure AD.
   Custom query:

```
SigninLogs
| where ClientAppUsed in ('Other clients; Older office clients', 'Other clients', 'Other clients; IMAP',
'Other clients; POP', 'Other clients; SMTP')
| where TimeGenerated >ago(30d)
| summarize count() by UserPrincipalName, IPAddress, ClientAppUsed
```



Create custom query
PREVIEW

* Name
Azure AD Legacy Protocol -- Admin43

Description
Hunt for Legacy Protocols across the estate hitting Azure AD.

Custom query

```
SigninLogs
| where ClientAppUsed in ('Other clients; Older office clients', 'Other clients', 'Other clients; IMA
| where TimeGenerated >ago(30d)
| summarize count() by UserPrincipalName, IPAddress, ClientAppUsed
```

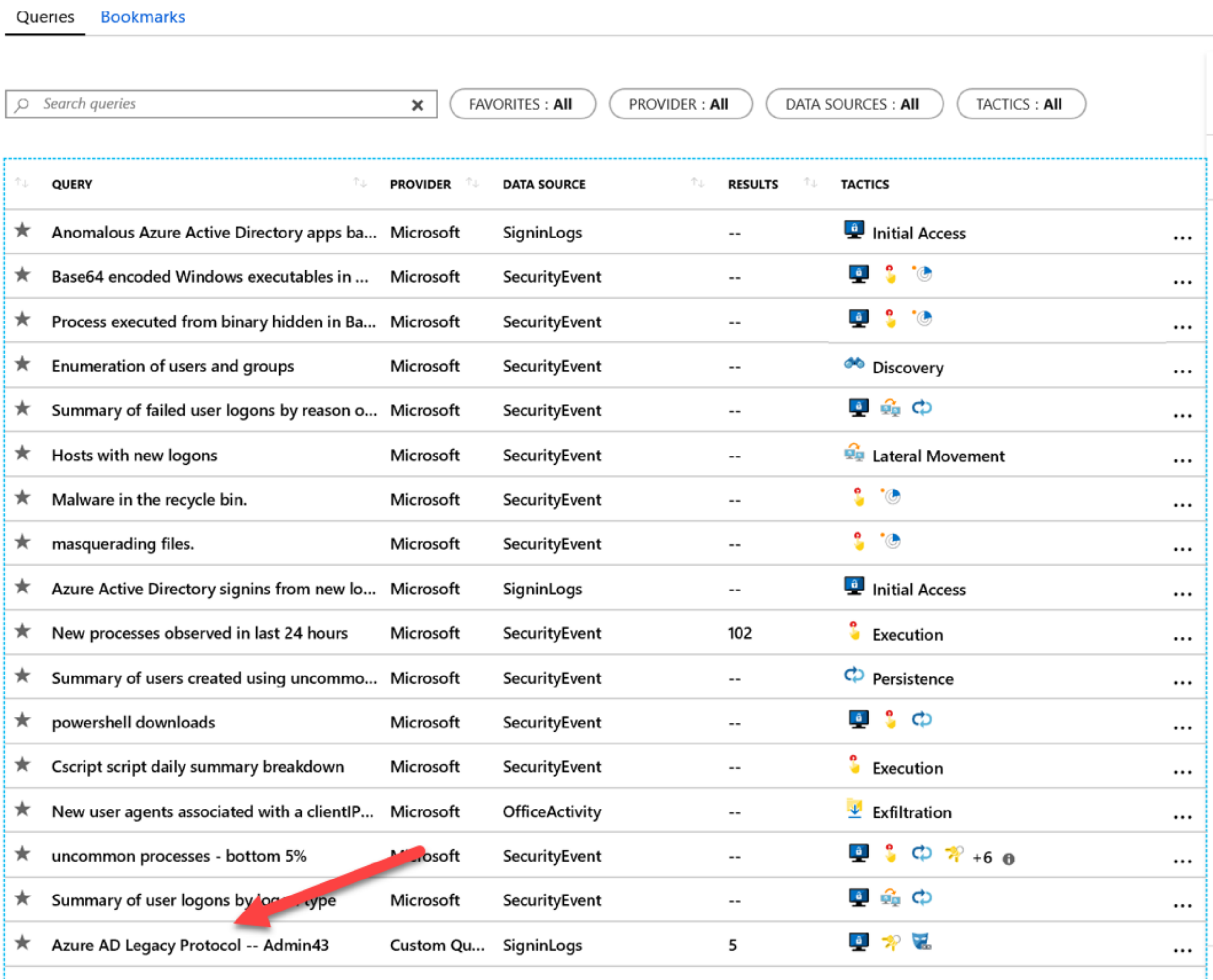
Entity mapping - more entities coming soon!

Use Entity type fields to map the fields in your query to entities recognized by Azure Sentinel. Entity type must be a string or Datetime.

ENTITY TYPE	PROPERTY	
Account	Choose column	Add
Host	Choose column	Add
IP	Choose column	Add
Timestamp	Choose column	Add

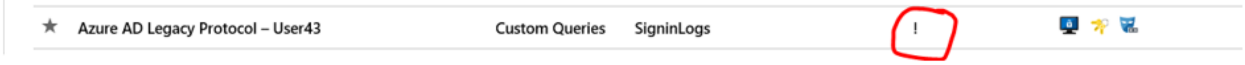
Tactics
0 selected

In this example, my user name is **Admin43**.
3. Leave the **Entity mapping** fields blank.
4. Add the Tactics that you think best match the attacker tactics you are hunting for. For example, **Initial Access, Privilege Escalation and Credential Access** are all risks of using Legacy Protocols.
5. Click **Create.**
6. On the main Hunting page, examine your new query.



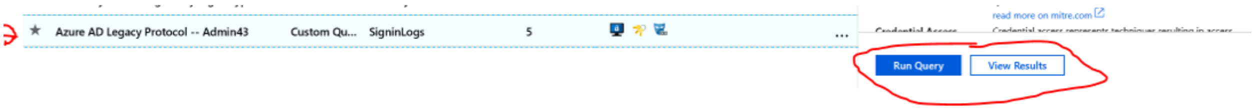
QUERY	PROVIDER	DATA SOURCE	RESULTS	TACTICS
Anomalous Azure Active Directory apps ba...	Microsoft	SigninLogs	--	Initial Access
Base64 encoded Windows executables in ...	Microsoft	SecurityEvent	--	
Process executed from binary hidden in Ba...	Microsoft	SecurityEvent	--	
Enumeration of users and groups	Microsoft	SecurityEvent	--	Discovery
Summary of failed user logons by reason o...	Microsoft	SecurityEvent	--	
Hosts with new logons	Microsoft	SecurityEvent	--	Lateral Movement
Malware in the recycle bin.	Microsoft	SecurityEvent	--	
masquerading files.	Microsoft	SecurityEvent	--	
Azure Active Directory signins from new lo...	Microsoft	SigninLogs	--	Initial Access
New processes observed in last 24 hours	Microsoft	SecurityEvent	102	Execution
Summary of users created using uncommo...	Microsoft	SecurityEvent	--	Persistence
powershell downloads	Microsoft	SecurityEvent	--	
Cscript script daily summary breakdown	Microsoft	SecurityEvent	--	Execution
New user agents associated with a clientIP...	Microsoft	OfficeActivity	--	Exfiltration
uncommon processes - bottom 5%	Microsoft	SecurityEvent	--	+6
Summary of user logons by logon type	Microsoft	SecurityEvent	--	
Azure AD Legacy Protocol -- Admin43	Custom Qu...	SigninLogs	5	

7. Click **Run all queries** to include your new query in the overall Hunting experience.
8. Note: If you receive an exclamation point for the result, do not worry. This is an intermittent known issue and should be ignored for now.



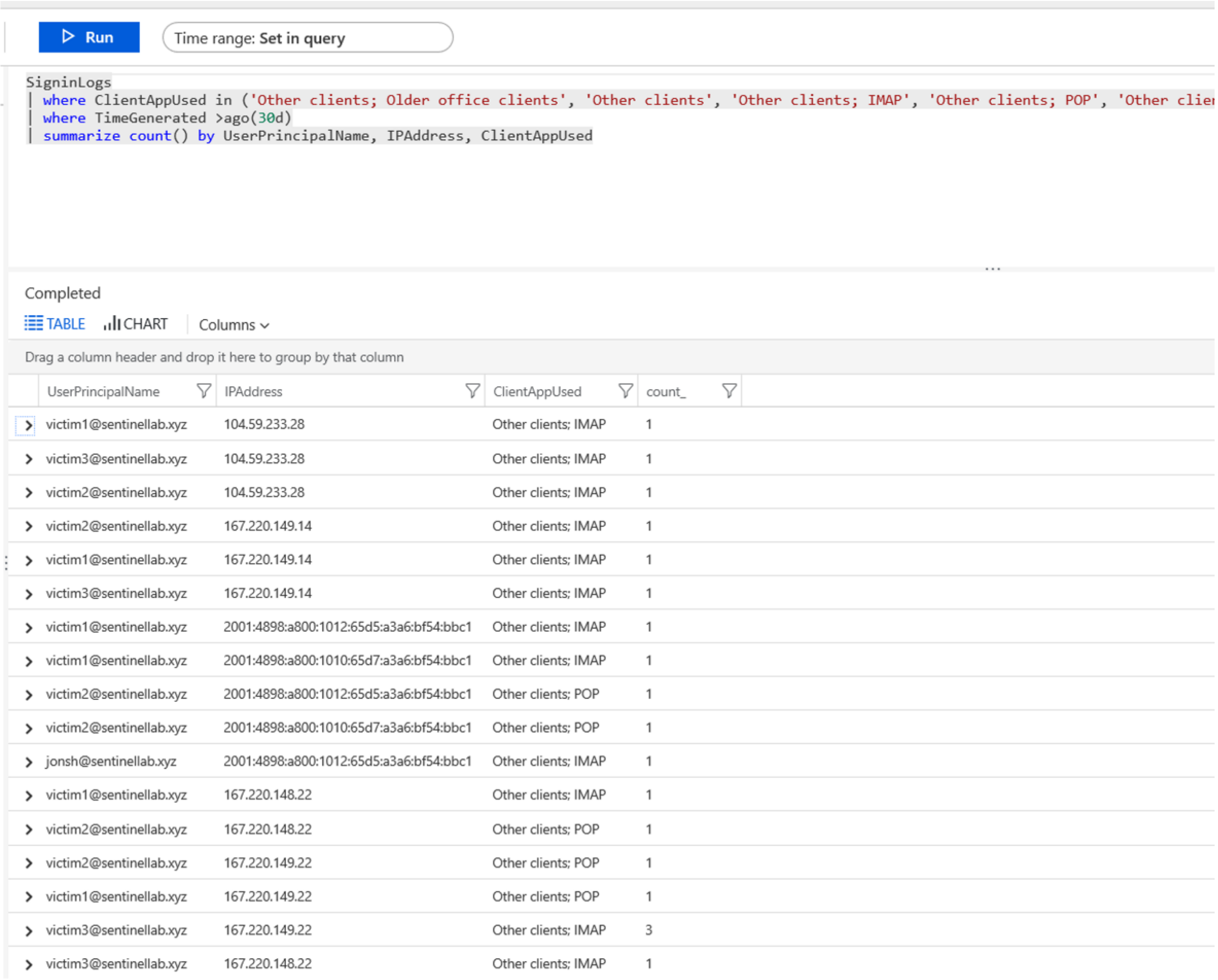
Azure AD Legacy Protocol – User43	Custom Queries	SigninLogs	!

10. Click on your new query, click **Run Query** and then click **View Results**.



Azure AD Legacy Protocol -- Admin43	Custom Qu...	SigninLogs	5

11. On the following page, examine the output. You are now able to Hunt for Users and IP Addresses that are using Legacy Protocols to access Azure AD; you also know the actual protocol used (IMAP, POP, etc.). Finally, the advantage of creating a Hunting Query is that anyone in your company can now easily reuse your work to hunt for misconfigurations or indicators of attack.



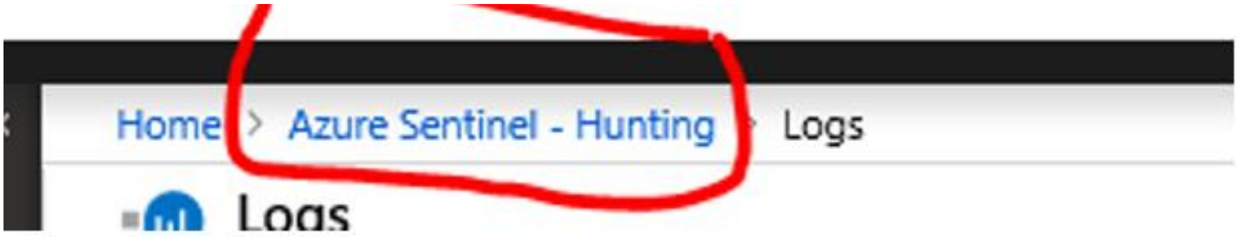
Time range: Set in query

```
SigninLogs
| where ClientAppUsed in ('Other clients; Older office clients', 'Other clients', 'Other clients; IMAP', 'Other clients; POP', 'Other clie
| where TimeGenerated >ago(30d)
| summarize count() by UserPrincipalName, IPAddress, ClientAppUsed
```

Completed
TABLE
CHART
Columns
Drag a column header and drop it here to group by that column

UserPrincipalName	IPAddress	ClientAppUsed	count_
victim1@sentinellab.xyz	104.59.233.28	Other clients; IMAP	1
victim3@sentinellab.xyz	104.59.233.28	Other clients; IMAP	1
victim2@sentinellab.xyz	104.59.233.28	Other clients; IMAP	1
victim2@sentinellab.xyz	167.220.149.14	Other clients; IMAP	1
victim1@sentinellab.xyz	167.220.149.14	Other clients; IMAP	1
victim3@sentinellab.xyz	167.220.149.14	Other clients; IMAP	1
victim1@sentinellab.xyz	2001:4898:a800:1012:65d5:a3a6:bf54:bbc1	Other clients; IMAP	1
victim1@sentinellab.xyz	2001:4898:a800:1010:65d7:a3a6:bf54:bbc1	Other clients; IMAP	1
victim2@sentinellab.xyz	2001:4898:a800:1012:65d5:a3a6:bf54:bbc1	Other clients; POP	1
victim2@sentinellab.xyz	2001:4898:a800:1010:65d7:a3a6:bf54:bbc1	Other clients; POP	1
jonsh@sentinellab.xyz	2001:4898:a800:1012:65d5:a3a6:bf54:bbc1	Other clients; IMAP	1
victim1@sentinellab.xyz	167.220.148.22	Other clients; IMAP	1
victim2@sentinellab.xyz	167.220.148.22	Other clients; POP	1
victim2@sentinellab.xyz	167.220.149.22	Other clients; POP	1
victim1@sentinellab.xyz	167.220.149.22	Other clients; POP	1
victim3@sentinellab.xyz	167.220.149.22	Other clients; IMAP	3
victim3@sentinellab.xyz	167.220.148.22	Other clients; IMAP	1

12. At the top, click **Azure Sentinel – Hunting.**


Home
Azure Sentinel - Hunting
Logs
Logs