Lab #13: **Automatic Alert Import**

Purpose:

- We'll be automatically importing several custom rules into Microsoft Sentinel.

Tasks:

1. **Import several Sentinel Analytics rules**
    - Download the raw JSON file
    - Import the JSON file
2. **Inspect the "Brute Force SUCCESS" alert**
    - Inspect the alert - Attempt #1
    - Inspect the alert - Attempt #2

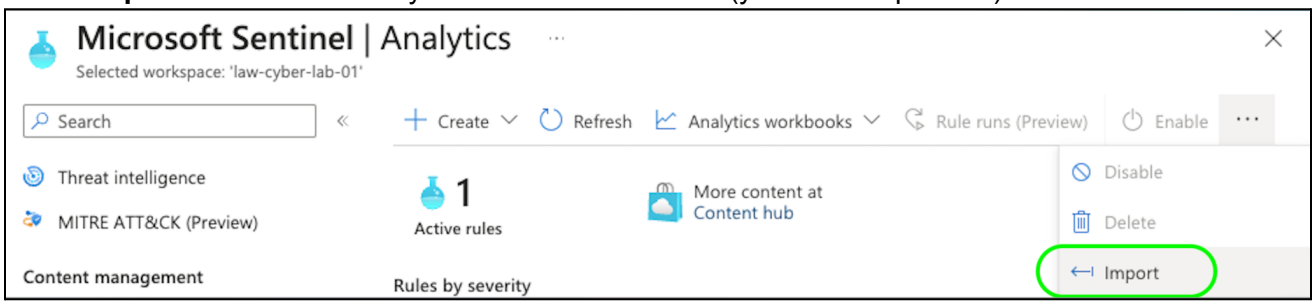
Task 1: **Import several Sentinel Analytics rules**

Download the JSON file:

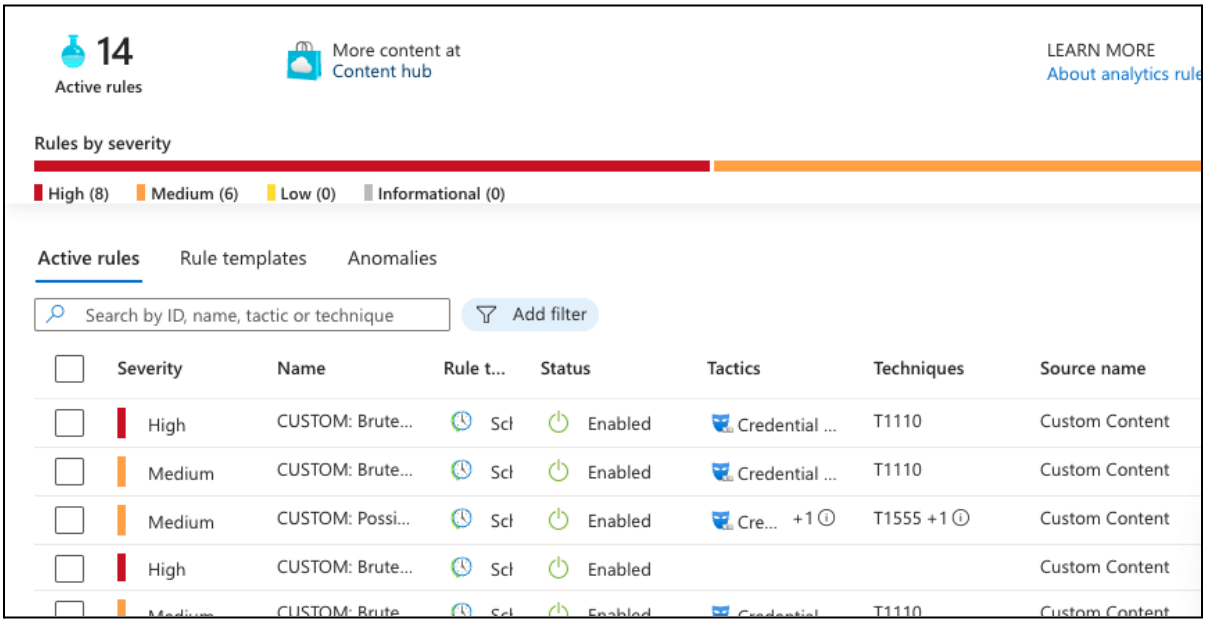
1. Open this GitHub page to view the custom rule file: https://github.com/erichmair/Azure-SOC-Honeynet-Project/blob/main/Sentinel-Analytics-Rules%20/Sentinel-Analytics-Rules(KQL%20Alert%20Queries).json
2. Now, download the raw file (save it to your local Desktop folder).

Import the JSON file:

1. **Azure** portal > **Sentinel** > select your workspace > **Analytics**
2. Select **Import** > select the newly downloaded JSON file (your Desktop folder).



3. You should now see all of your newly imported rules. All should be enabled.

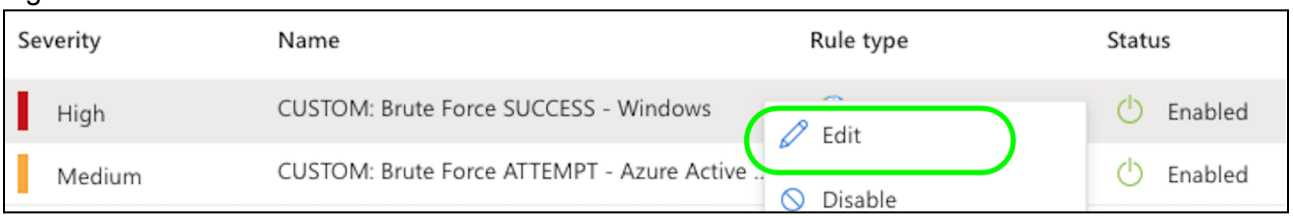


Task 2: Inspect the “Brute Force SUCCESS” alert

***Note**: The “CUSTOM: Brute Force SUCCESS - Windows” alert will return query results if somebody manages to successfully brute-force into our environment.*

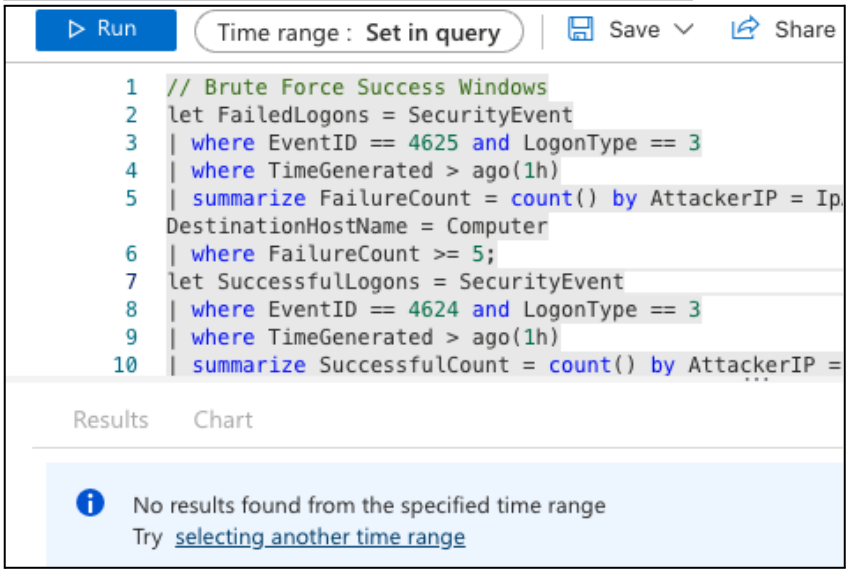
Inspect the alert - Attempt #1:

1. **Azure** portal > **Sentinel** > select your workspace > **Analytics** >
2. Right-click the “***CUSTOM: Brute Force SUCCESS - Windows***” rule > select **Edit**.



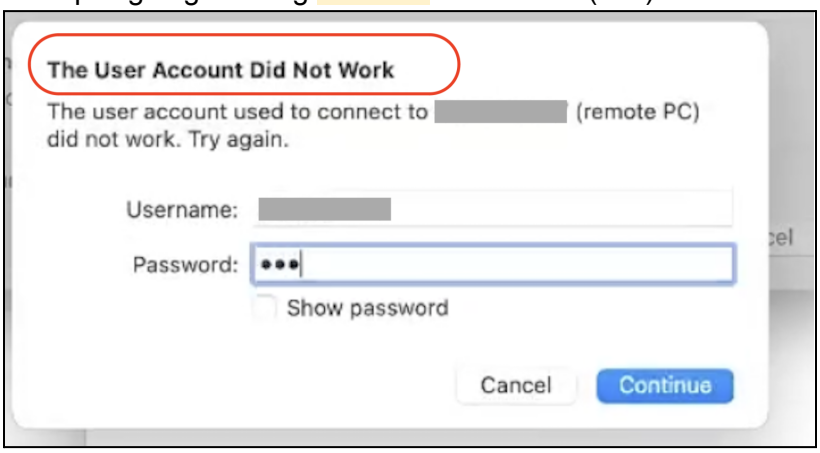
3. In the **Set Rule Logic** tab, copy the query that is in the **Rule Query** box.
4. Go to **Log Analytics workspace** > paste the query into the query box > **Run**

***Note**: No results should appear yet. This is normal.*



Inspect the alert - Attempt #2:

1. Open the **Microsoft Remote Desktop app** > attempt to sign into **windows-vm**
   a. Attempt signing in using incorrect credentials (10x).



   b. Now, sign in 1x using correct credentials.

> ***Note***: *These sign-in attempts should've now generated query results.*

2. Go back to Logs Analytics workspace > Logs > **Run** the query again.

```
// Brute Force Success Windows
let FailedLogons = SecurityEvent
| where EventID == 4625 and LogonType == 3
| where TimeGenerated > ago(1h)
| summarize FailureCount = count() by AttackerIP = IpAddress, EventID, Activity, LogonType, DestinationHostName =
| where FailureCount >= 5;
let SuccessfulLogons = SecurityEvent
| where EventID == 4624 and LogonType == 3
| where TimeGenerated > ago(1h)
| summarize SuccessfulCount = count() by AttackerIP = IpAddress, LogonType, DestinationHostName = Computer, Authen
SuccessfulLogons
| join kind = inner FailedLogons on DestinationHostName, AttackerIP, LogonType
| project AuthenticationSuccessTime, AttackerIP, DestinationHostName, FailureCount, SuccessfulCount
```

Results Chart

AuthenticationSuccessTime [Local Time]	AttackerIP	DestinationHostName	FailureCount
9/8/2023, 3:50:16.602 PM	[redacted]	windows-vm	23

End:

- We enabled SIEM rules in Microsoft Sentinel. In future labs, we'll keep our VMs running to generate logs for us to eventually analyze.