# Analyzing RDP Brute Force Attacks with Microsoft Sentinel

By: Ritvik Indupuri

## 1. Project Summary

This document details the methodology and results of a cloud security lab environment designed to simulate and analyze brute force attacks. The project demonstrates the full threat detection and incident response lifecycle using Microsoft Sentinel, from initial log ingestion to sophisticated incident analysis.

## 2. Project Architecture

The architecture is a streamlined pipeline for security event analysis, built entirely within the Microsoft Azure cloud platform. The process begins with an attack vector and concludes with an actionable security incident.

- **Your Local Machine:** Initiates the simulated RDP brute force attack.
- **Azure Windows VM:** The target system that generates Windows Security Event Logs, specifically **Event ID 4625** for failed login attempts.
- **Data Connector (AMA):** The Azure Monitor Agent, which forwards all collected Security Event Logs to a central log repository.
- **Log Analytics Workspace:** The centralized log storage in Azure where all security events are aggregated.
- **Microsoft Sentinel:** The cloud-native SIEM platform that connects to the Log Analytics Workspace to run analytic rules against the collected data.
- **Analytic Rule:** A custom-developed scheduled query rule that detects a specific pattern of failed logins.
- **Incident Alert:** The final output of the pipeline—a correlated, actionable alert that signifies a detected security threat.

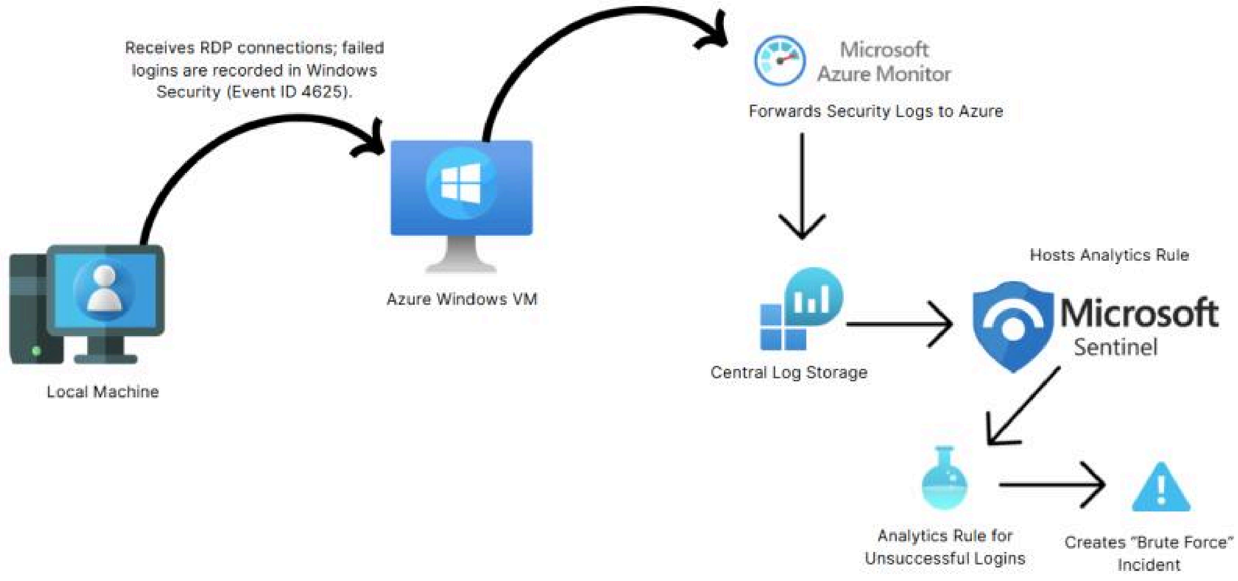

Receives RDP connections; failed logins are recorded in Windows Security (Event ID 4625).
Microsoft Azure Monitor
Forwards Security Logs to Azure
Azure Windows VM
Local Machine
Hosts Analytics Rule
Microsoft Sentinel
Central Log Storage
Analytics Rule for Unsuccessful Logins
Creates "Brute Force" Incident

*Figure 1: Project Architecture.*

*A visual representation of the end-to-end data flow, from the simulated attack on the Azure VM to the final incident alert in Microsoft Sentinel.*

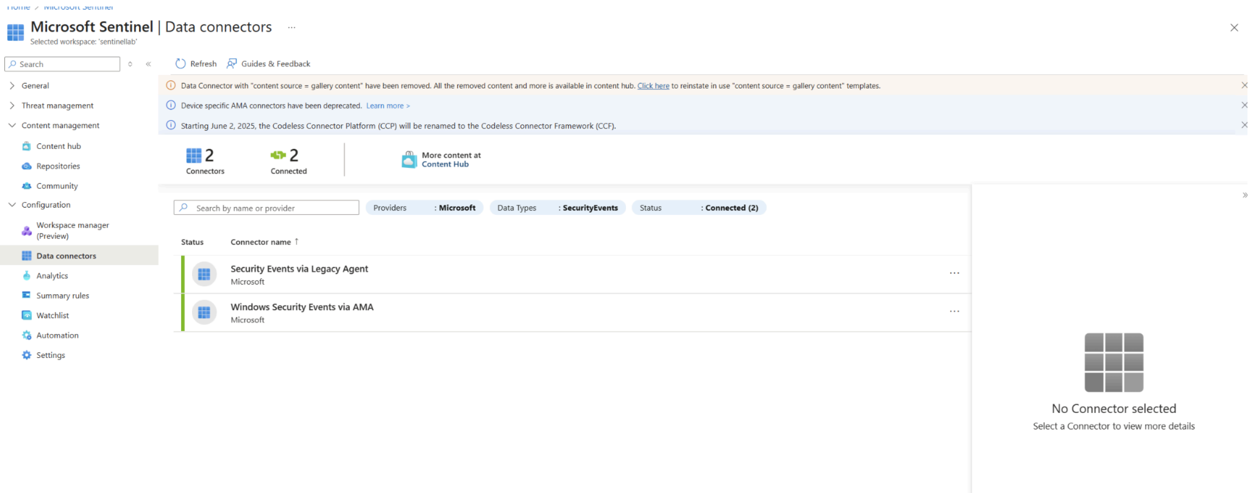
## 3. Tools and Technologies

- **Cloud Platform:** Microsoft Azure
- **SIEM:** Microsoft Sentinel
- **Virtualization:** Windows Virtual Machine
- **Log Collection:** Azure Monitor Agent (AMA)
- **Query Language:** Kusto Query Language (KQL)

## 4. Methodology and Key KQL Query

The project was executed in four key phases, from environment setup to incident closure.

**Phase I: Environment Setup** The first step involved configuring the **Windows Security Events via AMA** data connector. This ensured that all relevant security logs from the Azure VM were successfully ingested into Microsoft Sentinel, providing the necessary data for analysis.



Microsoft Sentinel | Data connectors

Selected workspace: 'sentinellab'

Data Connector with "content source = gallery content" have been removed. All the removed content and more is available in content hub. Click here to reinstate in use "content source = gallery content" templates.

Device specific AMA connectors have been deprecated. Learn more >

Starting June 2, 2025, the Codeless Connector Platform (CCP) will be renamed to the Codeless Connector Framework (CCF).

2 Connectors

2 Connected

Providers : Microsoft | Data Types : SecurityEvents | Status : Connected (2)

| Status | Connector name ↑ |
|---|---|
| | Security Events via Legacy Agent<br>Microsoft |
| | Windows Security Events via AMA<br>Microsoft |

No Connector selected

Select a Connector to view more details

*Figure 2: Data Connector Configuration.*

*This image shows the successful connection of the Windows Security Events via AMA connector, which is the foundation of the project's data pipeline.*

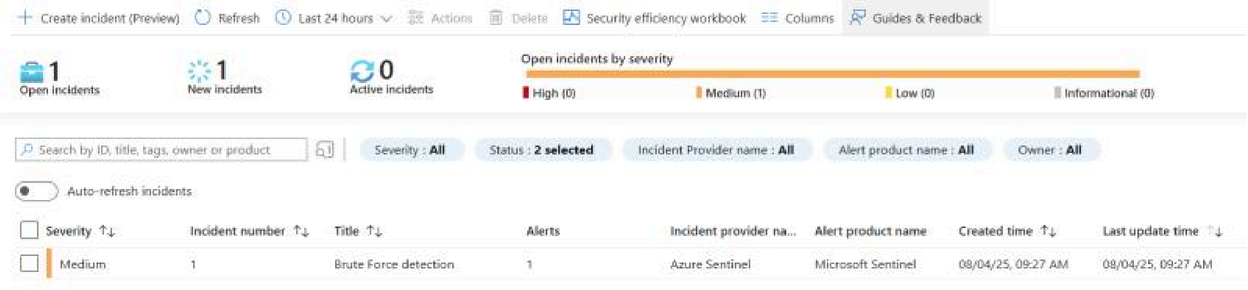
**Phase II: Custom Analytic Rule Development** A scheduled query rule was created to specifically detect a brute force pattern. This custom rule used KQL to analyze the incoming security logs for a specific event ID and pattern.

**The KQL Query used was:**

```
SecurityEvent
| where EventID == 4625
| summarize count() by IpAddress, bin(TimeGenerated, 5m)
| where count_ > 5
| extend Computer = computer, IpAddress = IpAddress, Account = Account
```

- `| where EventID == 4625`: Filters the logs to include only failed login attempts.
- `| summarize count() by IpAddress, bin(TimeGenerated, 5m)`: Correlates events by counting them for each unique IP address within a 5-minute time window.
- `| where count_ > 5`: Sets the detection logic to alert only when there are more than 5 failed attempts within that 5-minute window.
- `| extend Computer = computer, IpAddress = IpAddress, Account = Account`: Adds relevant details from the log to the final alert output.

**Phase III: Attack Simulation and Detection** An RDP brute force attack was simulated against the Azure VM. The analytic rule successfully triggered a security incident, confirming the detection logic was working.


+ Create incident (Preview)
Refresh
Last 24 hours
Actions
Delete
Security efficiency workbook
Columns
Guides & Feedback
Search by ID, title, tags, owner or product
Auto-refresh incidents

1 Open incidents
1 New incidents
0 Active incidents

Open incidents by severity
High (0) Medium (1) Low (0) Informational (0)

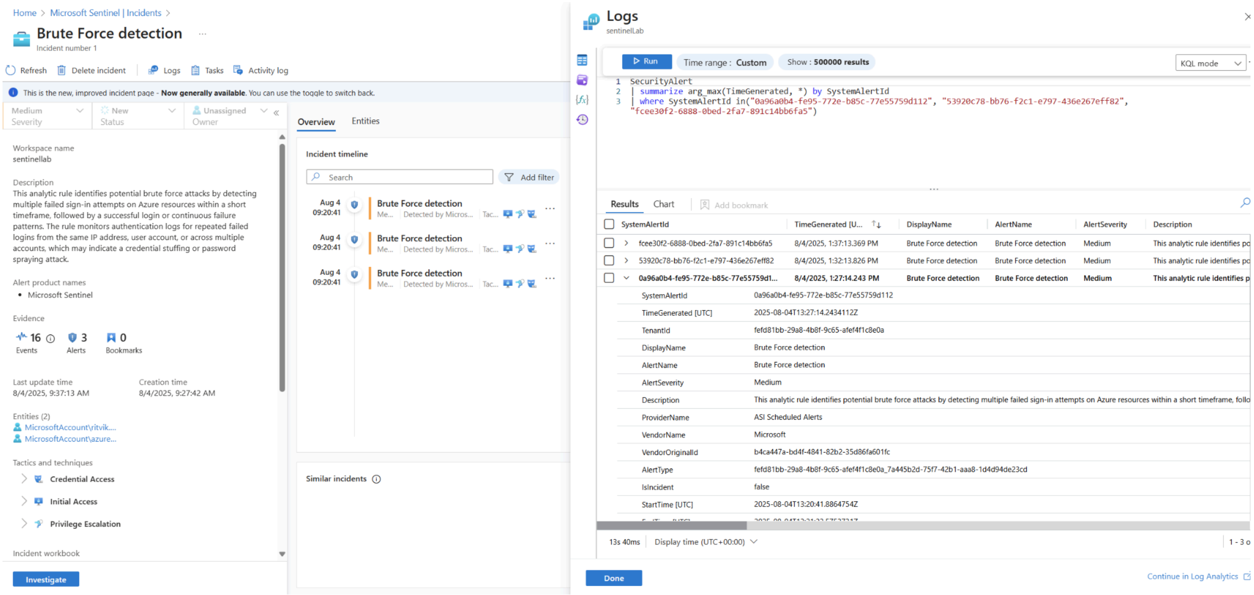
Severity : **All** Status : **2 selected** Incident Provider name : **All** Alert product name : **All** Owner : **All**

| □ Severity ↑↓ | Incident number ↑↓ | Title ↑↓ | Alerts | Incident provider na... | Alert product name | Created time ↑↓ | Last update time ↑↓ |
|---|---|---|---|---|---|---|---|
| □ Medium | 1 | Brute Force detection | 1 | Azure Sentinel | Microsoft Sentinel | 08/04/25, 09:27 AM | 08/04/25, 09:27 AM |

*Figure 3: Incident Detection.*

*The primary project outcome: a medium-severity 'Brute Force detection' incident has been successfully generated and is ready for investigation.*

**Phase IV: Incident Investigation** The generated incident was investigated by a security analyst. The `Logs` section of the incident was used to run ad-hoc KQL queries and examine the raw log data, validating the alert and confirming the details of the attack.



Brute Force detection

Incident number 1

This is the new, improved incident page - **Now generally available**. You can use the toggle to switch back.

Severity: Medium | Status: New | Owner: Unassigned

Workspace name
sentinellab

Description
This analytic rule identifies potential brute force attacks by detecting multiple failed sign-in attempts on Azure resources within a short timeframe, followed by a successful login or continuous failure patterns. The rule monitors authentication logs for repeated failed logins from the same IP address, user account, or across multiple accounts, which may indicate a credential stuffing or password spraying attack.

Alert product names
- Microsoft Sentinel

Evidence
16 Events | 3 Alerts | 0 Bookmarks

Last update time
8/4/2025, 9:37:13 AM

Creation time
8/4/2025, 9:27:42 AM

Entities (2)
MicrosoftAccount\ritvik....
MicrosoftAccount\azure...

Tactics and techniques
- Credential Access
- Initial Access
- Privilege Escalation

Incident workbook

Incident timeline

| | |
|---|---|
| Aug 4 09:20:41 | Brute Force detection<br>Me... Detected by Micros... Tac... |
| Aug 4 09:20:41 | Brute Force detection<br>Me... Detected by Micros... Tac... |
| Aug 4 09:20:41 | Brute Force detection<br>Me... Detected by Micros... Tac... |

Similar incidents

Logs
sentinelLab

Time range : Custom | Show : **500000 results**

```
SecurityAlert
| summarize arg_max(TimeGenerated, *) by SystemAlertId
| where SystemAlertId in("0a96a0b4-fe95-772e-b85c-77e55759d112", "53920c78-bb76-f2c1-e797-436e267eff82",
"fcee30f2-6888-0bed-2fa7-891c14bb6fa5")
```

| SystemAlertId | TimeGenerated [U... | DisplayName | AlertName | AlertSeverity | Description |
|---|---|---|---|---|---|
| fcee30f2-6888-0bed-2fa7-891c14bb6fa5 | 8/4/2025, 1:37:13.369 PM | Brute Force detection | Brute Force detection | Medium | This analytic rule identifies po |
| 53920c78-bb76-f2c1-e797-436e267eff82 | 8/4/2025, 1:32:13.826 PM | Brute Force detection | Brute Force detection | Medium | This analytic rule identifies po |
| **0a96a0b4-fe95-772e-b85c-77e55759d1...** | **8/4/2025, 1:27:14.243 PM** | **Brute Force detection** | **Brute Force detection** | **Medium** | **This analytic rule identifies p** |

| | |
|---|---|
| SystemAlertId | 0a96a0b4-fe95-772e-b85c-77e55759d112 |
| TimeGenerated [UTC] | 2025-08-04T13:27:14.2434112Z |
| TenantId | fefd81bb-29a8-4b8f-9c65-afef4f1c8e0a |
| DisplayName | Brute Force detection |
| AlertName | Brute Force detection |
| AlertSeverity | Medium |
| Description | This analytic rule identifies potential brute force attacks by detecting multiple failed sign-in attempts on Azure resources within a short timeframe, follo |
| ProviderName | ASI Scheduled Alerts |
| VendorName | Microsoft |
| VendorOriginalId | b4ca447a-bd4f-4841-82b2-35d86fa601fc |
| AlertType | fefd81bb-29a8-4b8f-9c65-afef4f1c8e0a_7a445b2d-75f7-42b1-aaa8-1d4d94de23cd |
| IsIncident | false |
| StartTime [UTC] | 2025-08-04T13:20:41.8864754Z |

13s 40ms | Display time (UTC+00:00) | 1 - 3 o

*Figure 4: Incident Investigation and KQL.*

*This image demonstrates the use of a KQL query to analyze the raw logs, allowing for a deeper understanding of the brute force attempts.*

## 5. Results and Conclusion

The project successfully demonstrated the end-to-end functionality of a cloud-native SIEM. The key outcome was the successful detection of a single medium-severity brute force incident. The project provided critical, hands-on experience in:

- SIEM deployment and configuration in a cloud environment.
- Developing and deploying custom detection logic using KQL.
- Performing incident triage and log analysis.
- Understanding the crucial difference between raw events and correlated alerts.