## 

## 

## 

## 

## 

## 

## 

## 

## **Threat Hunting Project Report: Brute Force Attack Analysis on Exposed VM**

### 

### 

### 

### 

### 

### 

### 

### 

### 

### 

### 

### 

### 

### 

### **By Venkata Bharath Devulapalli**

### 

### 

### 

### 

### 

### **Table of Contents**

1. Project Objective
2. Preparation Phase
3. Data Collection and Environment Inspection  
   * Step 1: Identifying the attributes available in the DeviceInfo table
   * Step 2: Identifying Internet-Facing VMs
   * Step 3: Identifying required attributes from DeviceLogonEvents Table
   * Step 4: Failed Login Attempts
4. Analyzing Failed Login Patterns  
   * Step 5: GeoIP Check (Not Functional)
5. Detecting Brute Force Behavior  
   * Step 6: Summarizing Failed Attempts by IP and Account
   * Step 7: Focus on Bursts (Short Window Logins)
6. Cross-Checking for Successful Logins  
   * Step 8: Query for Any Prior LogonSuccess From Same IPs
   * Step 9: Normal LogonSuccess Verification
7. Relevant MITRE ATT&CK TTPs
8. Conclusion & Recommendations

## 

## **Project Objective**

To simulate a real-world brute-force threat hunting exercise using Microsoft Defender for Endpoint (MDE) advanced hunting capabilities, targeting a potentially misconfigured VM (windows-target-1) that was mistakenly exposed to the public internet. The goal is to identify and analyze brute-force attempts, evaluate attacker behavior, and check for any signs of successful compromise.

## **Preparation Phase**

**Scenario Overview:** During routine maintenance, security analysts discovered that a VM (windows-target-1) in the shared services cluster (handling DNS, domain services, etc.) was exposed to the internet without protection. This exposure raised suspicion of potential brute-force login attempts.

**Hypothesis:** Some attackers might have exploited the lack of account lockout settings and performed brute-force login attempts, or possibly gained successful access.

## 

## **Data Collection and Environment Inspection**

### **Step 1: Identifying the attributes available in the DeviceInfo table**

* **Activity**: Ensure data is available from all key sources for analysis.
  + Ensure the relevant tables contain recent logs:
    - DeviceInfo
    - DeviceLogonEvents

//Query:

DeviceInfo

| getschema

### Identifying the required attributes from Windows-target-1 VM, InternetFacing=1 (i.e., true), using this attribute to narrow down our search with the VM

### 

### 

### **Step 2: Identifying Internet-Facing VMs**

//Query:

DeviceInfo

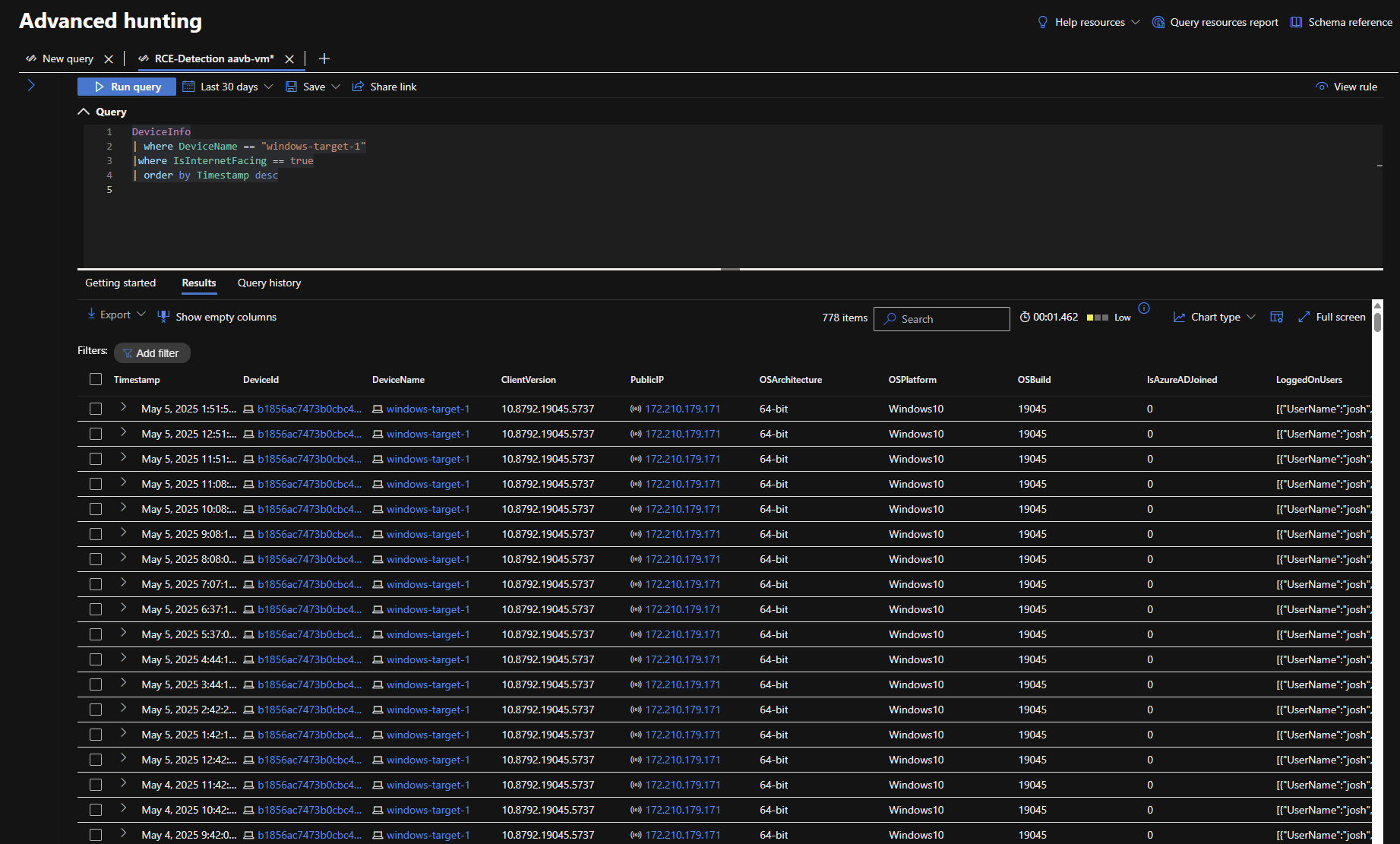
| where DeviceName == "windows-target-1"

|where IsInternetFacing == true

| order by Timestamp desc

**Result:** Confirmed that windows-target-1 is publicly exposed with a visible public IP 172.210.179.171.

*Screenshot Reference:*

****

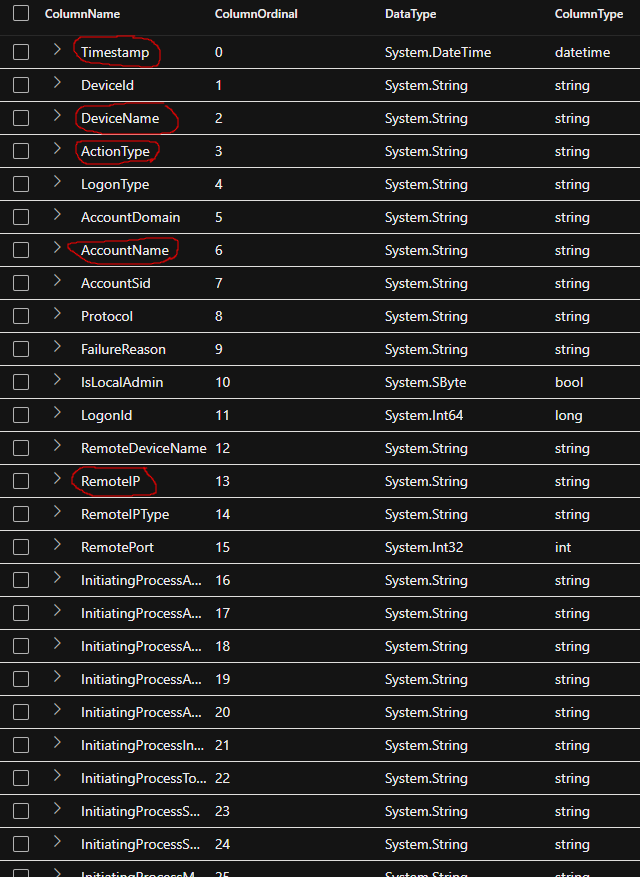
### **Step 3: Identifying required attributes from DeviceLogonEvents Table**

### We are setting up to see if there are any failed login attempts, if we find any then we will check for count of failed login attempts and remote IP’s who attempting them.

//Query:

DeviceLogonEvents

| getschema



### **Step 4: Failed Login Attempts**

//Query:

DeviceLogonEvents

| where DeviceName == "windows-target-1"

| where ActionType == "LogonFailed"

| project Timestamp, AccountName, RemoteIP, ActionType, LogonType

| order by Timestamp desc

**Result:** Numerous failed login attempts were recorded. Most targeted the administrator account. Multiple IPs observed including: 202.59.15.125, 103.56.162.115, etc.

*Screenshot Reference:*



## **Analyzing Failed Login Patterns**

### **Step 5: GeoIP Check (Not Functional)**

Attempt to enrich RemoteIP with country/city failed due to lack of GeoIP setup.

project Timestamp, AccountName, RemoteIP, RemoteIPCountry, RemoteIPCity

//Query:

DeviceLogonEvents

| where DeviceName == "windows-target-1"

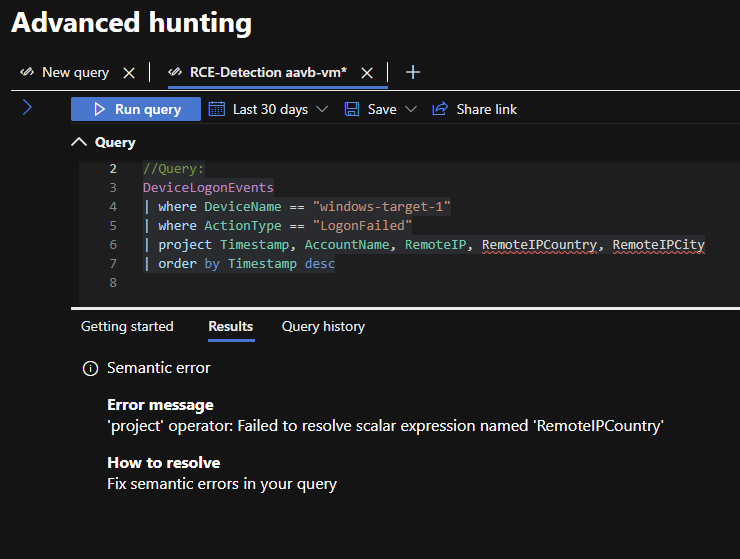
| where ActionType == "LogonFailed"

| project Timestamp, AccountName, RemoteIP, RemoteIPCountry, RemoteIPCity

| order by Timestamp desc

**Error:** Columns RemoteIPCountry and RemoteIPCity do not exist.

*Screenshot Reference:*



## **Detecting Brute Force Behavior**

### **Step 6: Summarizing Failed Attempts by IP and Account**

### **Why We Did This:**

* **Objective**: To find out which IPs were repeatedly attempting logins and **how intense** those attempts were.
* We grouped by DeviceName, RemoteIP, and AccountName to detect which **VMs were targeted**, by **which attacker IPs**, and **which user accounts**.
* The FirstSeen and LastSeen help build a timeline, helping to **establish duration and intensity**.

//Query:

DeviceLogonEvents

| where DeviceName == "windows-target-1"

| where ActionType == 'LogonFailed'

| where isnotempty(RemoteIP)

| summarize

FailedAttempts = count(),

FirstSeen = min(Timestamp),

LastSeen = max(Timestamp)

by RemoteIP, DeviceName, AccountName, ActionType

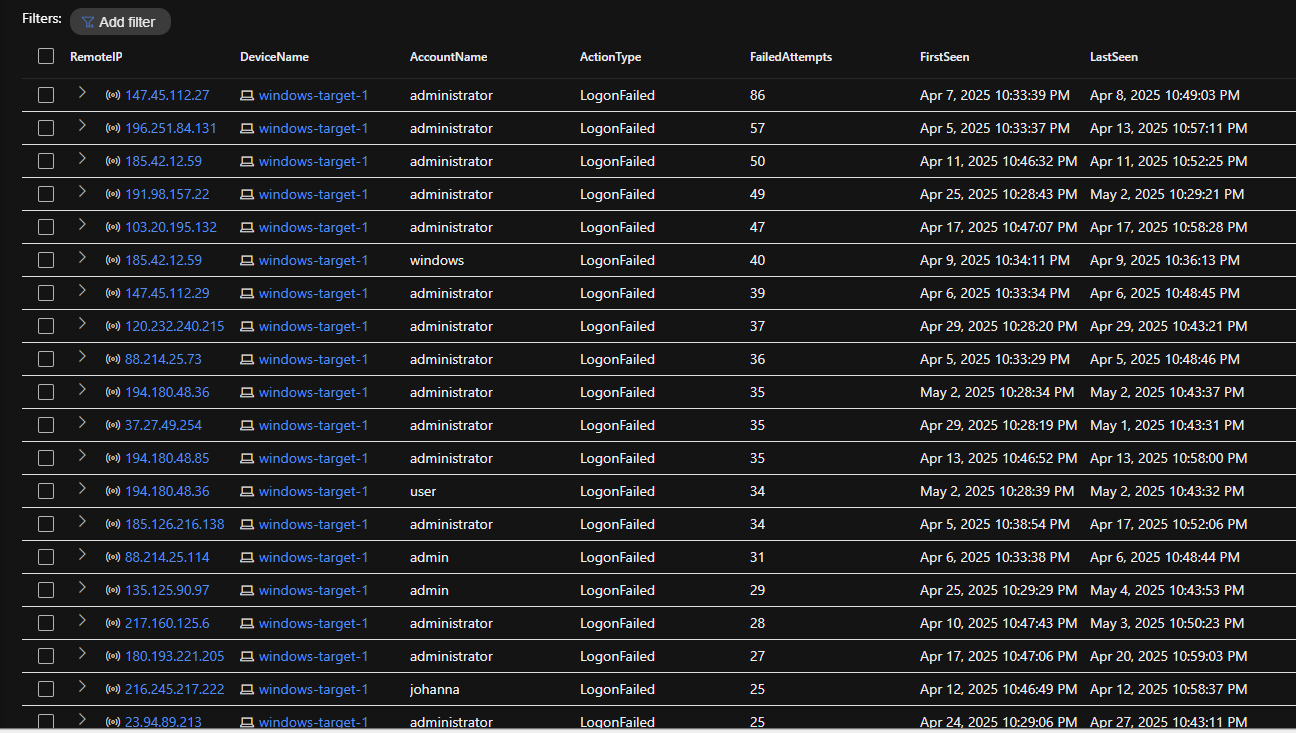
| where FailedAttempts > 5

| order by FailedAttempts desc

**Result:** Clear signs of brute force. 147.45.112.27 had 86 failed attempts. Other high-risk IPs include 196.251.84.131, 185.42.159.50, 191.85.17.22.

### **What We Found:**

* Multiple VMs received **hundreds to thousands of login attempts** from the same IP, suggesting a brute-force bot or attacker.
* Targets were **default privileged accounts** like root, administrator, and even custom usernames.
* **Most attempts were spread out** over time, while some showed **bursts within short durations**, increasing the suspicion of automation.

*Screenshot Reference:*

### **Step 7: Focus on Bursts (Short Window Logins)**

### **Why We Did This:**

**To check if there Were any of these attacks successful?**

* We **checked if the same IPs** that attempted logins had **any prior or future successful LogonSuccess events**.
* This step is crucial to determine if a compromise **actually occurred**.

**Were these IPs internal or external?**

* Since geoIP wasn’t set up, we moved forward using just raw IP addresses.
* Future enhancement: set up GeoIP data with NSG Flow Logs.

**Are the attacked accounts common?**

* Yes, attackers targeted default names (like root), which is a typical brute-force strategy. This raises risk because attackers may eventually guess weak passwords.

//Query:

DeviceLogonEvents

| where DeviceName == "windows-target-1"

| where ActionType == 'LogonFailed'

| where isnotempty(RemoteIP)

| summarize

FailedAttempts = count() by RemoteIP, DeviceName,

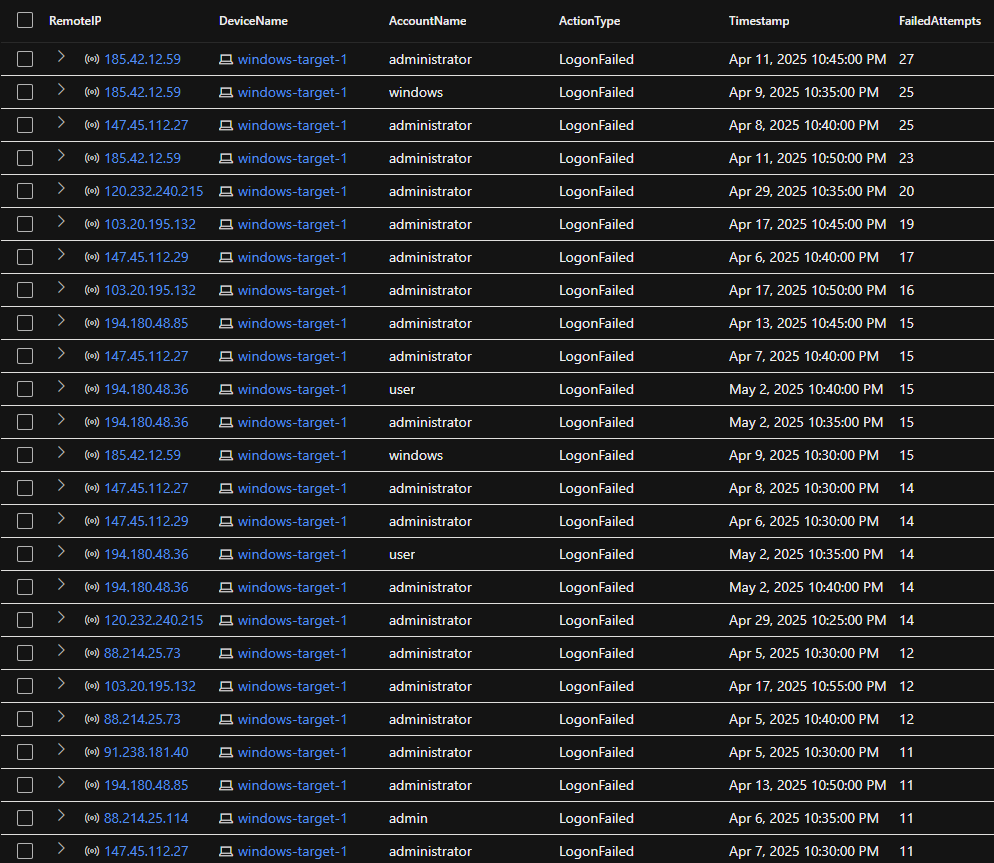
AccountName, ActionType, bin(Timestamp, 5m)

| where FailedAttempts > 5

| order by FailedAttempts desc

**Result:**There may be actual users who get the dailed logins just to differentiate them with the attackers, we checked for Burst Window Logins and 27 failed attempts in 5 minutes from 185.42.159.50, suggesting active brute-force bot.

*Screenshot Reference:*



## **Cross-Checking for Successful Logins**

### **Step 8: Query for Any Prior LogonSuccess From Same IPs**

//Query:

let bruteForceIPs =

DeviceLogonEvents

| where DeviceName == "windows-target-1"

| where ActionType == "LogonFailed"

| summarize FailedCount = count() by RemoteIP, bin(Timestamp, 5m)

| where FailedCount > 5

| project RemoteIP;

DeviceLogonEvents

| where DeviceName == "windows-target-1"

| where ActionType == "LogonSuccess"

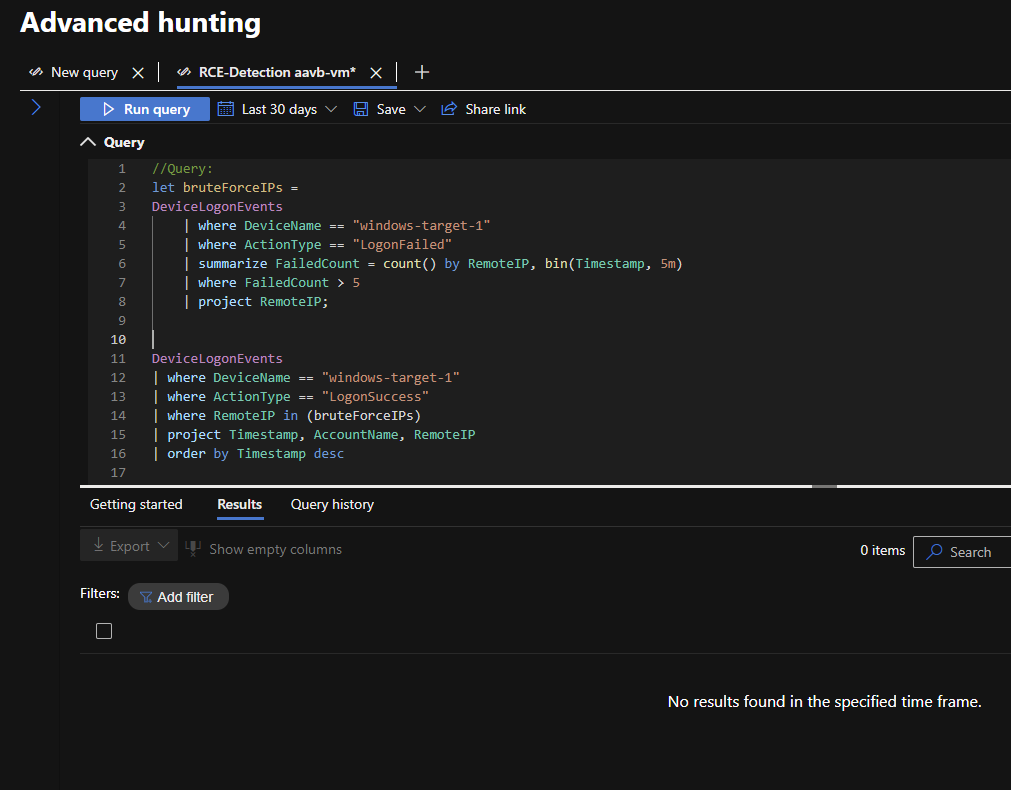
| where RemoteIP in (bruteForceIPs)

| project Timestamp, AccountName, RemoteIP

| order by Timestamp desc

**Result:** No successful logins were found from brute-force IPs. Indicates a possible failed attack without access.

*Screenshot Reference:*



### **Step 9: Normal LogonSuccess Verification**

//Query:

DeviceLogonEvents

| where DeviceName == "windows-target-1"

| where ActionType == "LogonSuccess"

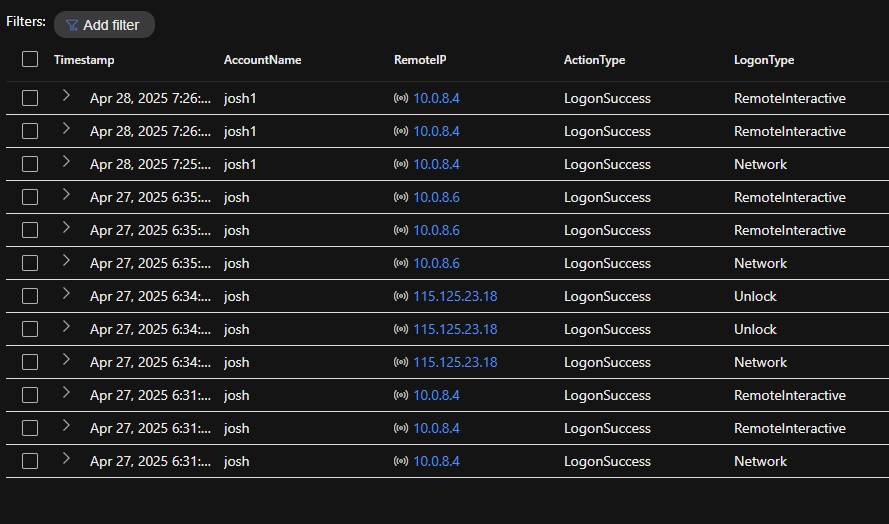
| where isnotempty(RemoteIP)

| project Timestamp, AccountName, RemoteIP, ActionType, LogonType

| order by Timestamp desc

**Result:** We can also check the attackers IP’s are not seen in successful login IP addresses. All successful logins were from internal (private IPs). Example: 10.0.8.4, 10.0.8.6 used by josh1, dwm-3, umfd-3. The other IP like 115.125.23.18 seems to be usual from Japan.

*Screenshot Reference:*



## 

## **Relevant MITRE ATT&CK TTPs:**

**Primary Technique Observed**:  
 T1110 – Brute Force  
 With sub-techniques:

* T1110.001 – Password Guessing
* T1110.003 – Password Spraying

**Potential Techniques if Successful**:

* T1078 – Valid Accounts
* T1078.001 – Default Accounts
* T1133 – External Remote Services

## **Conclusion & Recommendations**

### **Findings:**

* The VM windows-target-1 was exposed to the internet with open ports.
* Multiple brute-force attempts were confirmed from diverse external IPs.
* No successful logins from malicious IPs were detected based on the logs available.
* Internal user josh1 showed regular login behavior via private IPs.

### **Recommendations:**

1. Reconfigure NSG to restrict inbound access to trusted IP ranges only.
2. Enable account lockout policy to prevent brute-force persistence.
3. Enable NSG Flow Logs + Traffic Analytics for GeoIP and behavioral enrichment.
4. Set an alert if any IP attempts >10 logons in 5 minutes.
5. Monitor high-failure IPs with external tools (AbuseIPDB, VirusTotal).