## Scenario 3: Potential Impossible Travel 9/20/2025

**Explanation**:

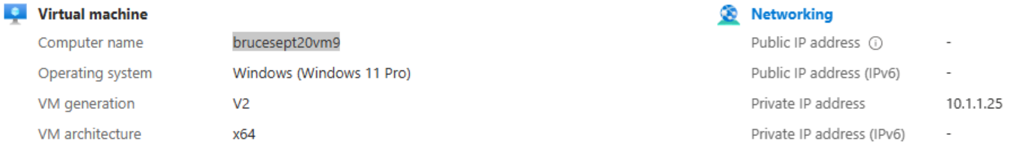
Sometimes corporations have policies against working outside of designated geographic regions, account sharing (this should be standard), or use of non-corporate VPNs. The following scenario will be used to detect unusual logon behavior by creating an incident if a user's login patterns are too erratic. “Too erratic” can be defined as logging in from multiple geographic regions within a given time period.

Whenever a user logs into Azure or authenticates with their main Azure account, logs will be created in the “SigninLogs” table, which is being forwarded to the Log Analytics Workspace being used by Microsoft Sentinel, our SIEM. Within Sentinel, we will define an alert to trigger whenever a user logs into more than one location in a 7 day time period. Not all triggers will be true positives, but it will give us a chance to investigate.

**Preliminary Steps:**

- Create a Virtual Machine.
- Onboard the Virtual Machine to Microsoft Defender for Endpoint (MDE).
- Open Sentinel in: https://portal.azure.com/ to create the Schedule Query Rule in: Sentinel → Analytics → Schedule Query Rule

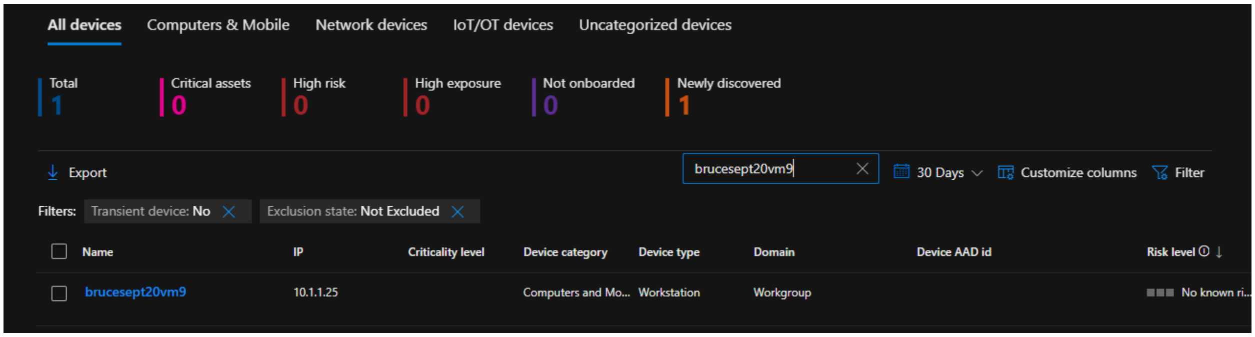
Virtual Machine Name: brucesept20vm9
Virtual Machine Operating System: Windows 11



| Virtual machine | |
|---|---|
| Computer name | brucesept20vm9 |
| Operating system | Windows (Windows 11 Pro) |
| VM generation | V2 |
| VM architecture | x64 |

| Networking | |
|---|---|
| Public IP address | - |
| Public IP address (IPv6) | - |
| Private IP address | 10.1.1.25 |
| Private IP address (IPv6) | - |

Screenshot of Virtual Machine Onboarded to Microsoft Defender for Endpoint (MDE)



Total 1 | Critical assets 0 | High risk 0 | High exposure 0 | Not onboarded 0 | Newly discovered 1

brucesept20vm9

Filters: Transient device: No | Exclusion state: Not Excluded

| Name | IP | Criticality level | Device category | Device type | Domain | Device AAD id | Risk level |
|---|---|---|---|---|---|---|---|
| brucesept20vm9 | 10.1.1.25 | | Computers and Mo... | Workstation | Workgroup | | No known ri... |

## Part 1: Create Alert Rule (Potential Impossible Travel)

Design a Sentinel Scheduled Query Rule within Log Analytics that will discover when a user logs in to more than a certain number of locations within a given time period.

**Query Rule Designed in Sentinel**: BrucePotential Impossible Travel
**Script used in Alert Rule**:

Description:

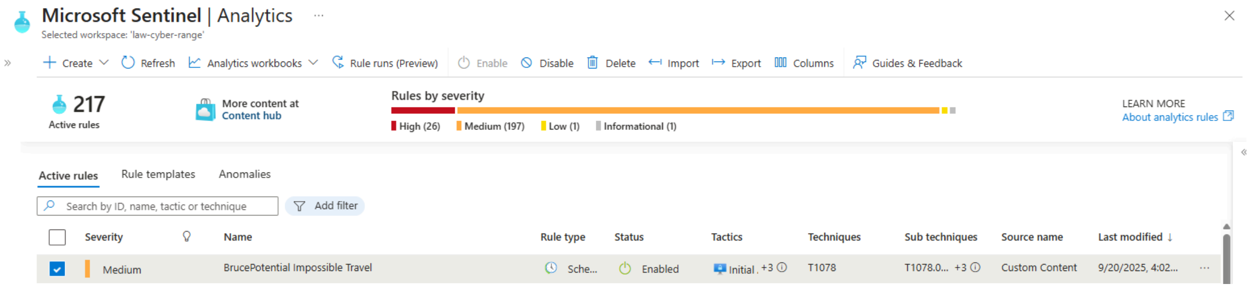
Will discover when a user logs in to more than a certain number of locations within a given time period; for example, trigger if a user logs into 2 different geographic regions within a 7 day time period.

```
// Locate Instances of Potential Impossible Travel
let TimePeriodThreshold = timespan(7d); // Change to how far back you want to look
let NumberOfDifferentLocationsAllowed = 1;
SigninLogs
| where TimeGenerated > ago(TimePeriodThreshold)
| summarize Count = count() by UserPrincipalName, UserId, City =
tostring(parse_json(LocationDetails).city), State = tostring(parse_json(LocationDetails).state),
Country = tostring(parse_json(LocationDetails).countryOrRegion)
| project UserPrincipalName, UserId, City, State, Country
| summarize PotentialImpossibleTravelInstances = count() by UserPrincipalName, UserId
| where PotentialImpossibleTravelInstances > NumberOfDifferentLocationsAllowed
```

**MITRE ATT&CK Framework:**

T1078 —
Defense Evasion
Initial Access
Persistence
Privilege Escalation
Valid Accounts (technique) — use of legitimate credentials to access systems.

**Alert Rule Creation Time**: 9/20/25, 04:02 PM



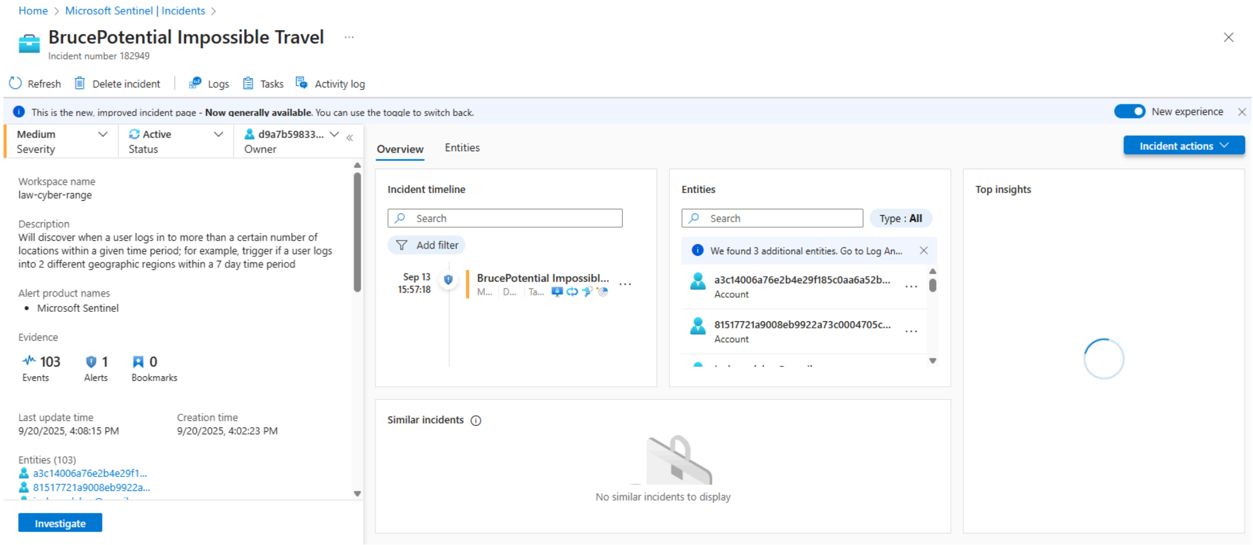
| ☐ | Severity | Name | Rule type | Status | Tactics | Techniques | Sub techniques | Source name | Last modified ↓ |
|---|---|---|---|---|---|---|---|---|---|
| ☑ | Medium | BrucePotential Impossible Travel | Sche... | Enabled | Initial , +3 | T1078 | T1078.0... +3 | Custom Content | 9/20/2025, 4:02... |

## Part 2: Trigger Alert to Create Incident

In order to generate the necessary logs for us to detect this activity, simply Remote Desktop Access our Virtual Machine, open up an Internet Browser, and then log into azure (https://portal.azure.com) from within your VM. This will trigger a new logon event in some random city on the East Coast (east us 2) somewhere.

This Alert has been triggered: 9/20/2025 04:08 PM

This Incident has been assigned to me, and the Status has been changed to “Active.”



Home > Microsoft Sentinel | Incidents >

BrucePotential Impossible Travel

Incident number 182949

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

Severity: Medium

Status: Active

Owner: d9a7b59833...

Workspace name
law-cyber-range

Description
Will discover when a user logs in to more than a certain number of locations within a given time period; for example, trigger if a user logs into 2 different geographic regions within a 7 day time period

Alert product names
- Microsoft Sentinel

Evidence
103 Events
1 Alerts
0 Bookmarks

Last update time
9/20/2025, 4:08:15 PM

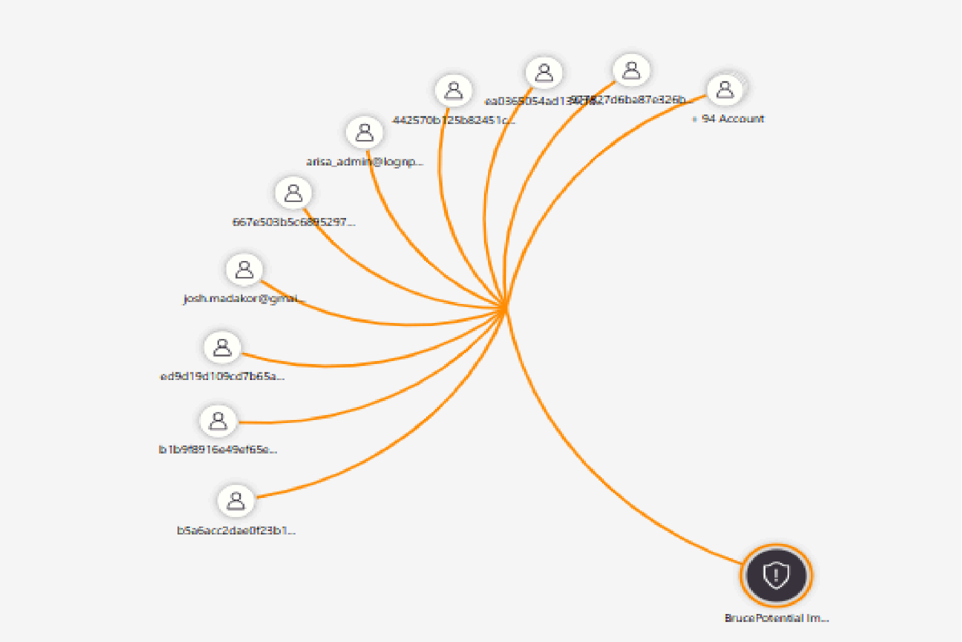
Creation time
9/20/2025, 4:02:23 PM

Entities (103)
a3c14006a76e2b4e29f1...
81517721a9008eb9922a...

Incident timeline
Sep 13 15:57:18 BrucePotential Impossibl...
M... D... Ta...

Entities
We found 3 additional entities. Go to Log An...
a3c14006a76e2b4e29f185c0aa6a52b...
Account
81517721a9008eb9922a73c0004705c...
Account

Similar incidents
No similar incidents to display



---

**Part 3: Work Incident**

Now we will work our incident to completion and close it out, in accordance with the NIST 800-61: Incident Response Lifecycle.

**Detection and Analysis**

Gather relevant evidence and assess impact.

It has been observed that many different instances that have now triggered our Potential Impossible Travel Alert rule exist, and we have evidence of this activity.

Here is an example:

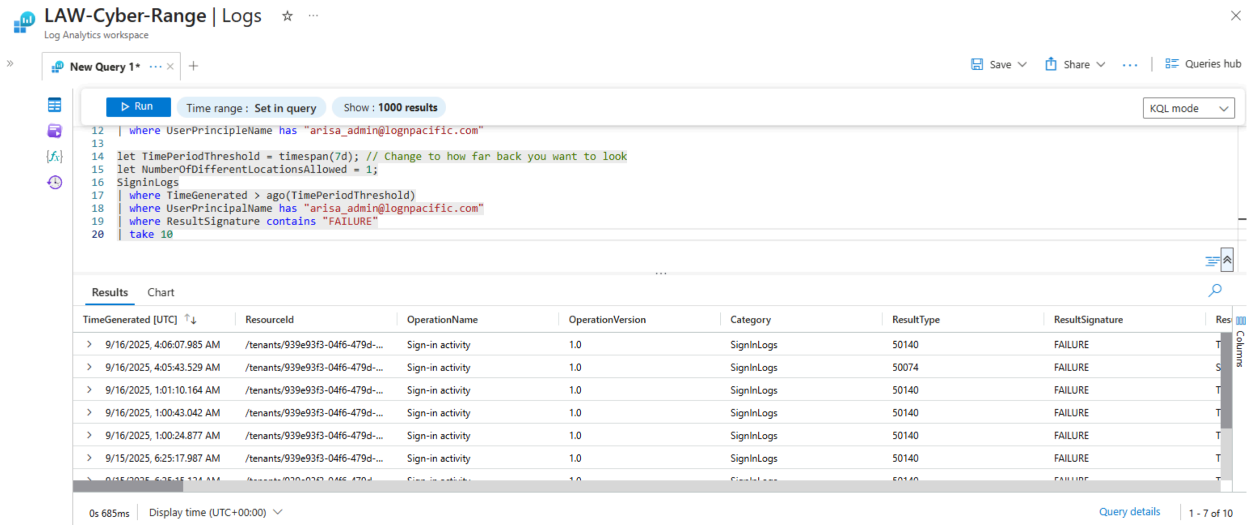
arisa_admin@lognpacific.com (4 instances)

---

```
let TimePeriodThreshold = timespan(7d); // Change to how far back you want to look
let NumberOfDifferentLocationsAllowed = 1;
SigninLogs
```

```
| where TimeGenerated > ago(TimePeriodThreshold)
| where UserPrincipalName has "arisa_admin@lognpacific.com"
| where ResultSignature has "FAILURE"
| take 10
```

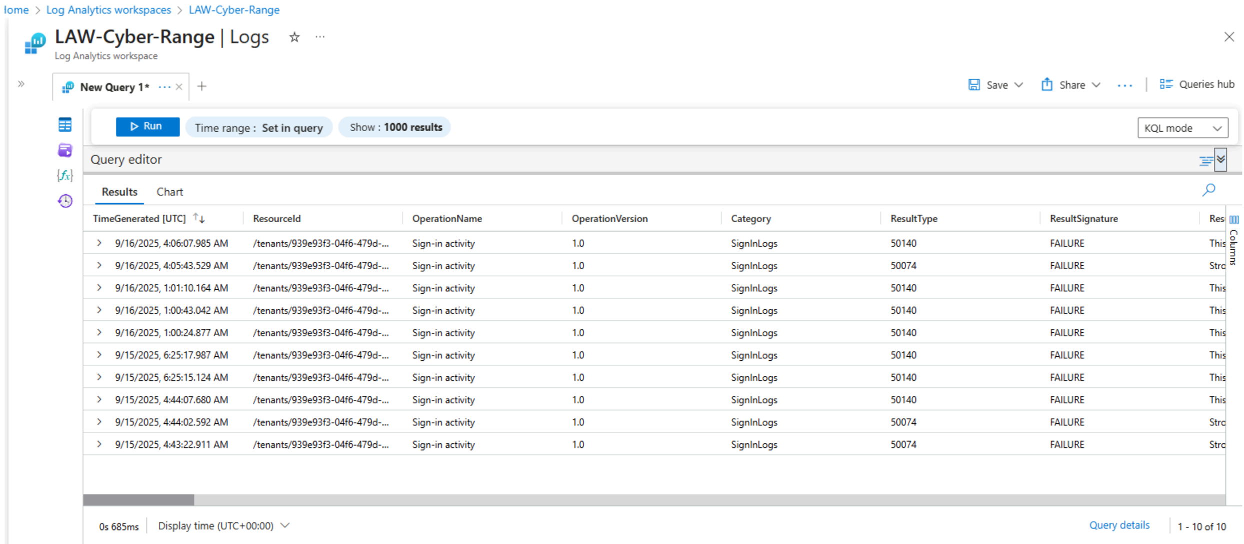
These screenshots show the return from the query above, showing 10 instances of ResultSignature are returned as FAILURE.



```
| where UserPrincipleName has "arisa_admin@lognpacific.com"

let TimePeriodThreshold = timespan(7d); // Change to how far back you want to look
let NumberOfDifferentLocationsAllowed = 1;
SigninLogs
| where TimeGenerated > ago(TimePeriodThreshold)
| where UserPrincipalName has "arisa_admin@lognpacific.com"
| where ResultSignature contains "FAILURE"
| take 10
```

| TimeGenerated [UTC] | ResourceId | OperationName | OperationVersion | Category | ResultType | ResultSignature | Res |
|---|---|---|---|---|---|---|---|
| 9/16/2025, 4:06:07.985 AM | /tenants/939e93f3-04f6-479d-... | Sign-in activity | 1.0 | SignInLogs | 50140 | FAILURE | T |
| 9/16/2025, 4:05:43.529 AM | /tenants/939e93f3-04f6-479d-... | Sign-in activity | 1.0 | SignInLogs | 50074 | FAILURE | S |
| 9/16/2025, 1:01:10.164 AM | /tenants/939e93f3-04f6-479d-... | Sign-in activity | 1.0 | SignInLogs | 50140 | FAILURE | T |
| 9/16/2025, 1:00:43.042 AM | /tenants/939e93f3-04f6-479d-... | Sign-in activity | 1.0 | SignInLogs | 50140 | FAILURE | T |
| 9/16/2025, 1:00:24.877 AM | /tenants/939e93f3-04f6-479d-... | Sign-in activity | 1.0 | SignInLogs | 50140 | FAILURE | T |
| 9/15/2025, 6:25:17.987 AM | /tenants/939e93f3-04f6-479d-... | Sign-in activity | 1.0 | SignInLogs | 50140 | FAILURE | T |

0s 685ms | Display time (UTC+00:00) | Query details | 1 - 7 of 10



| TimeGenerated [UTC] | ResourceId | OperationName | OperationVersion | Category | ResultType | ResultSignature | Res |
|---|---|---|---|---|---|---|---|
| 9/16/2025, 4:06:07.985 AM | /tenants/939e93f3-04f6-479d-... | Sign-in activity | 1.0 | SignInLogs | 50140 | FAILURE | This |
| 9/16/2025, 4:05:43.529 AM | /tenants/939e93f3-04f6-479d-... | Sign-in activity | 1.0 | SignInLogs | 50074 | FAILURE | Stro |
| 9/16/2025, 1:01:10.164 AM | /tenants/939e93f3-04f6-479d-... | Sign-in activity | 1.0 | SignInLogs | 50140 | FAILURE | This |
| 9/16/2025, 1:00:43.042 AM | /tenants/939e93f3-04f6-479d-... | Sign-in activity | 1.0 | SignInLogs | 50140 | FAILURE | This |
| 9/16/2025, 1:00:24.877 AM | /tenants/939e93f3-04f6-479d-... | Sign-in activity | 1.0 | SignInLogs | 50140 | FAILURE | This |
| 9/15/2025, 6:25:17.987 AM | /tenants/939e93f3-04f6-479d-... | Sign-in activity | 1.0 | SignInLogs | 50140 | FAILURE | This |
| 9/15/2025, 6:25:15.124 AM | /tenants/939e93f3-04f6-479d-... | Sign-in activity | 1.0 | SignInLogs | 50140 | FAILURE | This |
| 9/15/2025, 4:44:07.680 AM | /tenants/939e93f3-04f6-479d-... | Sign-in activity | 1.0 | SignInLogs | 50140 | FAILURE | This |
| 9/15/2025, 4:44:02.592 AM | /tenants/939e93f3-04f6-479d-... | Sign-in activity | 1.0 | SignInLogs | 50074 | FAILURE | Stro |
| 9/15/2025, 4:43:22.911 AM | /tenants/939e93f3-04f6-479d-... | Sign-in activity | 1.0 | SignInLogs | 50074 | FAILURE | Stro |

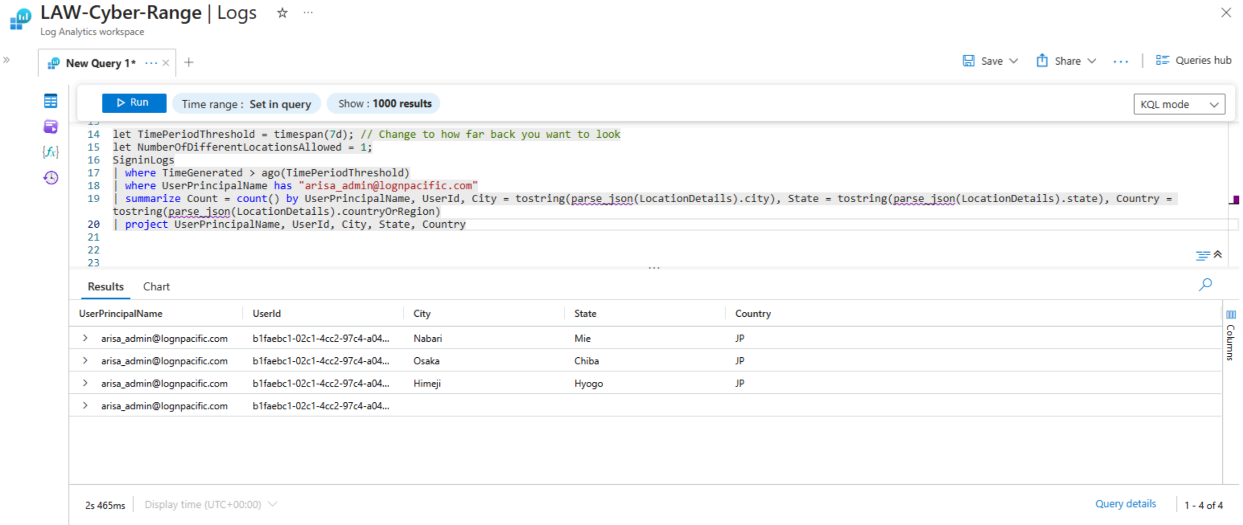
0s 685ms | Display time (UTC+00:00) | Query details | 1 - 10 of 10

---

This next addition allows us to see the City, State, Country, UserID, and UserPrincipleName.

The line | where ResultSignature has “FAILURE” must be removed.

```
let TimePeriodThreshold = timespan(7d); // Change to how far back you want to look
let NumberOfDifferentLocationsAllowed = 1;
SigninLogs
| where TimeGenerated > ago(TimePeriodThreshold)
| where UserPrincipalName has "arisa_admin@lognpacific.com"
| summarize Count = count() by UserPrincipalName, UserId, City =
tostring(parse_json(LocationDetails).city), State = tostring(parse_json(LocationDetails).state),
Country = tostring(parse_json(LocationDetails).countryOrRegion)
| project UserPrincipalName, UserId, City, State, Country
```



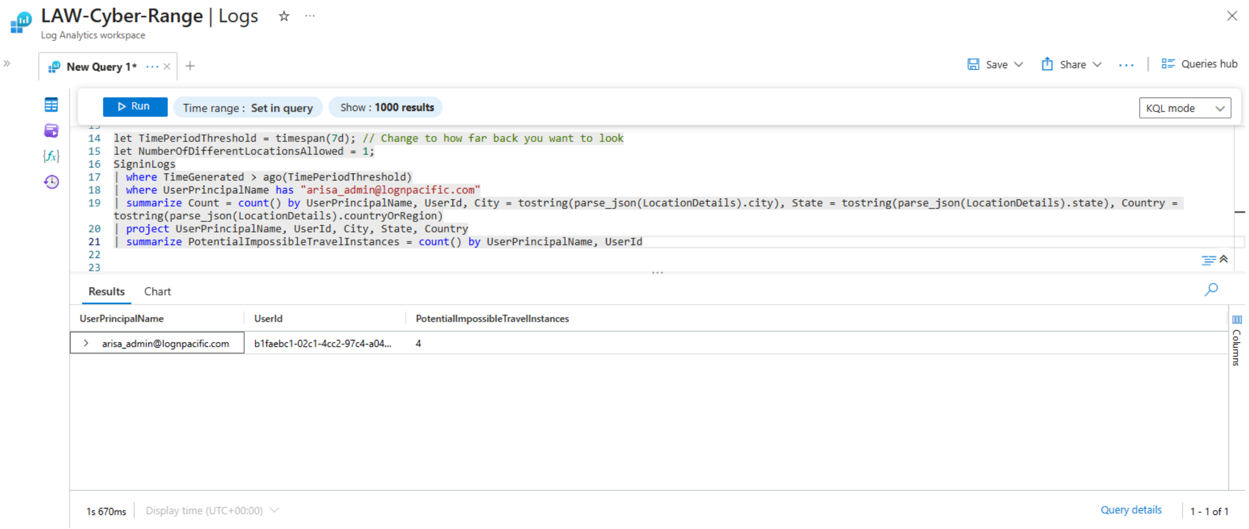
```
let TimePeriodThreshold = timespan(7d); // Change to how far back you want to look
let NumberOfDifferentLocationsAllowed = 1;
SigninLogs
| where TimeGenerated > ago(TimePeriodThreshold)
| where UserPrincipalName has "arisa_admin@lognpacific.com"
| summarize Count = count() by UserPrincipalName, UserId, City = tostring(parse_json(LocationDetails).city), State = tostring(parse_json(LocationDetails).state), Country =
tostring(parse_json(LocationDetails).countryOrRegion)
| project UserPrincipalName, UserId, City, State, Country
```

| UserPrincipalName | UserId | City | State | Country |
|---|---|---|---|---|
| arisa_admin@lognpacific.com | b1faebc1-02c1-4cc2-97c4-a04... | Nabari | Mie | JP |
| arisa_admin@lognpacific.com | b1faebc1-02c1-4cc2-97c4-a04... | Osaka | Chiba | JP |
| arisa_admin@lognpacific.com | b1faebc1-02c1-4cc2-97c4-a04... | Himeji | Hyogo | JP |
| arisa_admin@lognpacific.com | b1faebc1-02c1-4cc2-97c4-a04... | | | |

We have now taken the script that is within the Alert Rule and broken it down, utilized variations to identify where suspicious activity is evident, and returned back to where we had originally found 4 instances of Potential Impossible Travel.

Adding in this next line we have this information narrowed down to just one entry.

```
let TimePeriodThreshold = timespan(7d); // Change to how far back you want to look
let NumberOfDifferentLocationsAllowed = 1;
SigninLogs
| where TimeGenerated > ago(TimePeriodThreshold)
| where UserPrincipalName has "arisa_admin@lognpacific.com"
| summarize Count = count() by UserPrincipalName, UserId, City =
tostring(parse_json(LocationDetails).city), State = tostring(parse_json(LocationDetails).state),
Country = tostring(parse_json(LocationDetails).countryOrRegion)
| project UserPrincipalName, UserId, City, State, Country
| summarize PotentialImpossibleTravelInstances = count() by UserPrincipalName, UserId
```



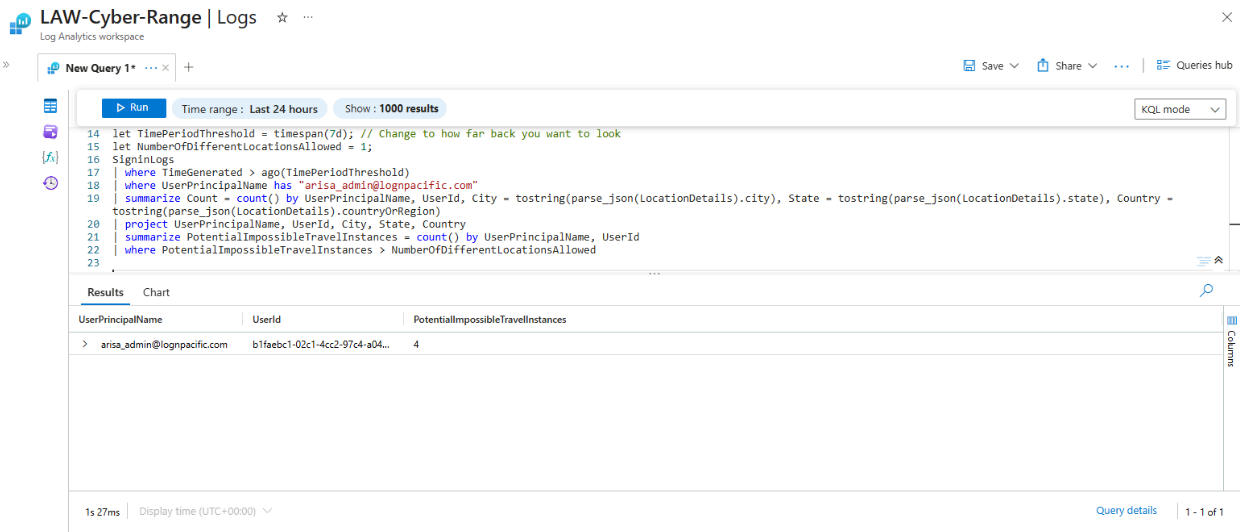
```
let TimePeriodThreshold = timespan(7d); // Change to how far back you want to look
let NumberOfDifferentLocationsAllowed = 1;
SigninLogs
| where TimeGenerated > ago(TimePeriodThreshold)
| where UserPrincipalName has "arisa_admin@lognpacific.com"
| summarize Count = count() by UserPrincipalName, UserId, City = tostring(parse_json(LocationDetails).city), State = tostring(parse_json(LocationDetails).state), Country =
tostring(parse_json(LocationDetails).countryOrRegion)
| project UserPrincipalName, UserId, City, State, Country
| summarize PotentialImpossibleTravelInstances = count() by UserPrincipalName, UserId


```

| UserPrincipalName | UserId | PotentialImpossibleTravelInstances |
|---|---|---|
| arisa_admin@lognpacific.com | b1faebc1-02c1-4cc2-97c4-a04... | 4 |

This screenshot shows the summarizing of the PotentialImpossibleTravelInstances amount, by the UserPrincipleName and UserID.

Adding the next line of our script from the Alert Rule narrows down this investigation even further.

let TimePeriodThreshold = timespan(7d); // Change to how far back you want to look
let NumberOfDifferentLocationsAllowed = 1;
SigninLogs
| where TimeGenerated > ago(TimePeriodThreshold)
| where UserPrincipalName has "arisa_admin@lognpacific.com"
| summarize Count = count() by UserPrincipalName, UserId, City = tostring(parse_json(LocationDetails).city), State = tostring(parse_json(LocationDetails).state), Country = tostring(parse_json(LocationDetails).countryOrRegion)
| project UserPrincipalName, UserId, City, State, Country
| summarize PotentialImpossibleTravelInstances = count() by UserPrincipalName, UserId
| where PotentialImpossibleTravelInstances > NumberOfDifferentLocationsAllowed



Time range : Last 24 hours

Show : 1000 results

KQL mode

```
let TimePeriodThreshold = timespan(7d); // Change to how far back you want to look
let NumberOfDifferentLocationsAllowed = 1;
SigninLogs
| where TimeGenerated > ago(TimePeriodThreshold)
| where UserPrincipalName has "arisa_admin@lognpacific.com"
| summarize Count = count() by UserPrincipalName, UserId, City = tostring(parse_json(LocationDetails).city), State = tostring(parse_json(LocationDetails).state), Country =
tostring(parse_json(LocationDetails).countryOrRegion)
| project UserPrincipalName, UserId, City, State, Country
| summarize PotentialImpossibleTravelInstances = count() by UserPrincipalName, UserId
| where PotentialImpossibleTravelInstances > NumberOfDifferentLocationsAllowed
```

| UserPrincipalName | UserId | PotentialImpossibleTravelInstances |
|---|---|---|
| arisa_admin@lognpacific.com | b1faebc1-02c1-4cc2-97c4-a04... | 4 |

1s 27ms   Display time (UTC+00:00)   1 - 1 of 1

This screenshot shows the final line to the Alert Rule script that show the PotentialImpossibleTravelInstances Greater than the NumberOfDifferentLocationsAllowed with the UserPrincipleName, UserID, and PotentialImpossibleTravelInstances.

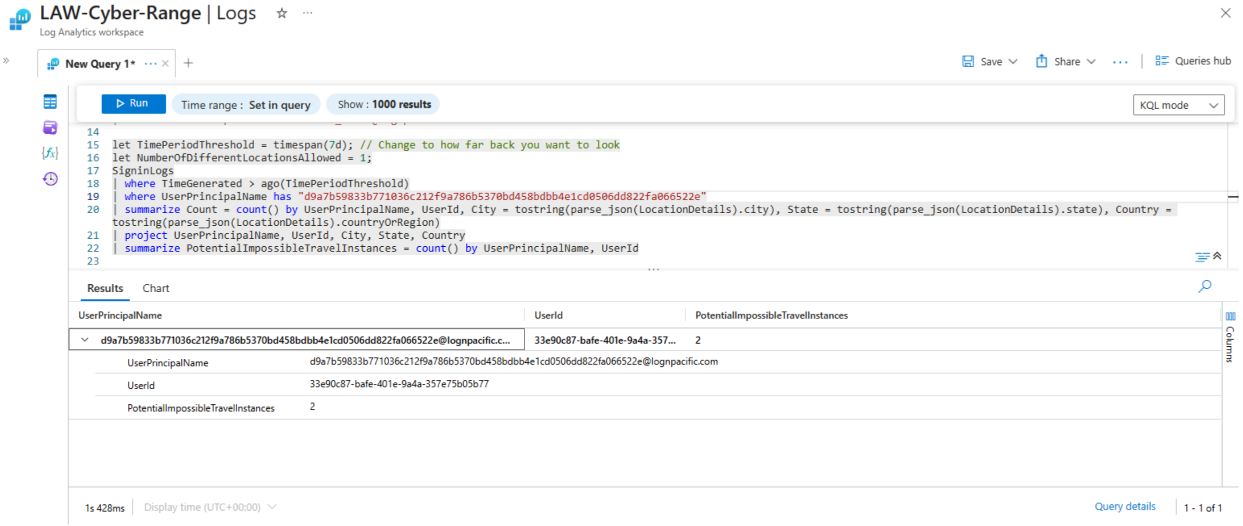
---

Observed the first account: arisa_admin@lognpacific.com  that I have selected and this has some activity that could be viewed as suspicious. It is actually quite normal.

The next instance that will be selected is the Virtual Machine brucesept20vm8.
This is the device that the Alert Rule was associated with, and this is the device that triggered this rule.

Here the line in the script has been modified to include the UserPrincipleName in the same way that we studied the account from earlier. It is a long string, however it is very helpful in narrowing down this information:

This screenshot shows 2 instances that occurred from my account, where I initiated triggering the Alert Rule.



```
let TimePeriodThreshold = timespan(7d); // Change to how far back you want to look
let NumberOfDifferentLocationsAllowed = 1;
SigninLogs
| where TimeGenerated > ago(TimePeriodThreshold)
| where UserPrincipalName has "d9a7b59833b771036c212f9a786b5370bd458bdbb4e1cd0506dd822fa066522e"
| summarize Count = count() by UserPrincipalName, UserId, City = tostring(parse_json(LocationDetails).city), State = tostring(parse_json(LocationDetails).state), Country =
tostring(parse_json(LocationDetails).countryOrRegion)
| project UserPrincipalName, UserId, City, State, Country
| summarize PotentialImpossibleTravelInstances = count() by UserPrincipalName, UserId
```

| UserPrincipalName | UserId | PotentialImpossibleTravelInstances |
|---|---|---|
| d9a7b59833b771036c212f9a786b5370bd458bdbb4e1cd0506dd822fa066522e@lognpacific.c... | 33e90c87-bafe-401e-9a4a-357... | 2 |

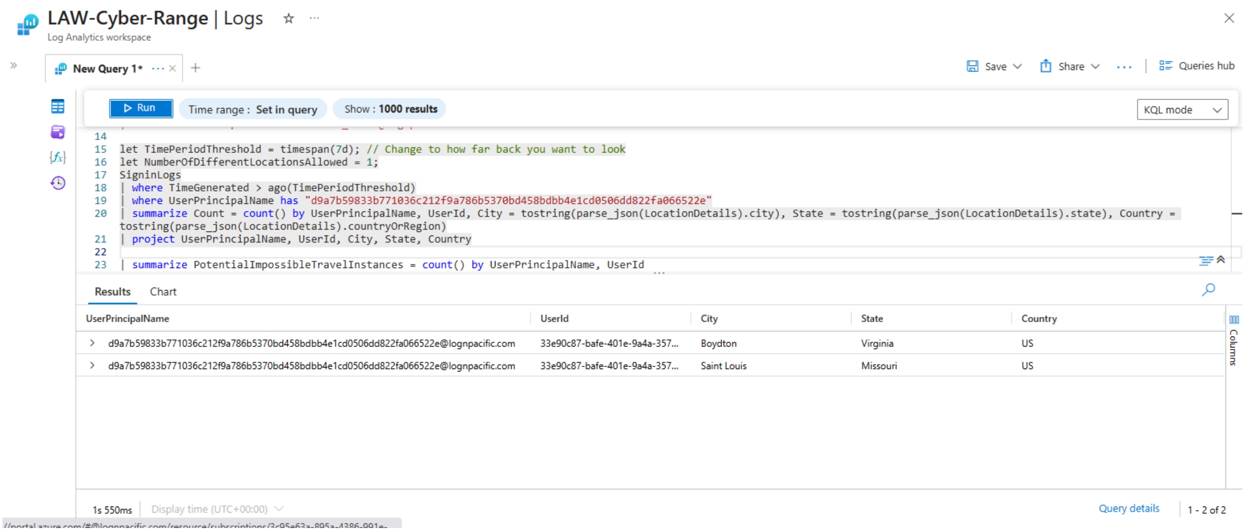
| | |
|---|---|
| UserPrincipalName | d9a7b59833b771036c212f9a786b5370bd458bdbb4e1cd0506dd822fa066522e@lognpacific.com |
| UserId | 33e90c87-bafe-401e-9a4a-357e75b05b77 |
| PotentialImpossibleTravelInstances | 2 |

1s 428ms | Display time (UTC+00:00) | Query details | 1 - 1 of 1

The script was changed to demonstrate that when I logged into the Virtual Machine inside of the Virtual Machine, it spawned an instance that appears to sign me in from Boydton, Virginia US. Shown here in this screenshot, the first legitimate sign in occurred from St. Louis Missouri. Then the following sign in that is suspicious occurred in Virginia.

```
let TimePeriodThreshold = timespan(7d); // Change to how far back you want to look
let NumberOfDifferentLocationsAllowed = 1;
SigninLogs
| where TimeGenerated > ago(TimePeriodThreshold)
| where UserPrincipalName has
"d9a7b59833b771036c212f9a786b5370bd458bdbb4e1cd0506dd822fa066522e"
| summarize Count = count() by UserPrincipalName, UserId, City =
tostring(parse_json(LocationDetails).city), State = tostring(parse_json(LocationDetails).state),
Country = tostring(parse_json(LocationDetails).countryOrRegion)
| project UserPrincipalName, UserId, City, State, Country
```

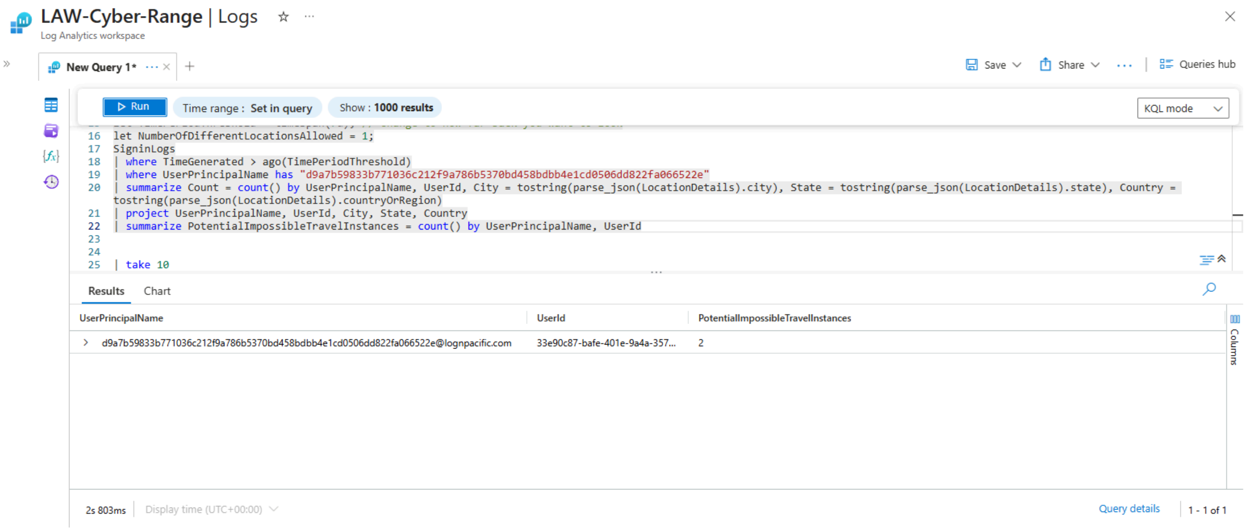
Utilizing the option of separating out the UserPrincipalName, UserId, City, State, and Country shows the Impossible Travel that would have had to occur for this output and this screenshot:



```
let TimePeriodThreshold = timespan(7d); // Change to how far back you want to look
let NumberOfDifferentLocationsAllowed = 1;
SigninLogs
| where TimeGenerated > ago(TimePeriodThreshold)
| where UserPrincipalName has "d9a7b59833b771036c212f9a786b5370bd458bdbb4e1cd0506dd822fa066522e"
| summarize Count = count() by UserPrincipalName, UserId, City = tostring(parse_json(LocationDetails).city), State = tostring(parse_json(LocationDetails).state), Country =
tostring(parse_json(LocationDetails).countryOrRegion)
| project UserPrincipalName, UserId, City, State, Country

| summarize PotentialImpossibleTravelInstances = count() by UserPrincipalName, UserId
```

| UserPrincipalName | UserId | City | State | Country |
|---|---|---|---|---|
| d9a7b59833b771036c212f9a786b5370bd458bdbb4e1cd0506dd822fa066522e@lognpacific.com | 33e90c87-bafe-401e-9a4a-357... | Boydton | Virginia | US |
| d9a7b59833b771036c212f9a786b5370bd458bdbb4e1cd0506dd822fa066522e@lognpacific.com | 33e90c87-bafe-401e-9a4a-357... | Saint Louis | Missouri | US |

And then the final script that was included in our Alert Rule.



```
let NumberOfDifferentLocationsAllowed = 1;
SigninLogs
| where TimeGenerated > ago(TimePeriodThreshold)
| where UserPrincipalName has "d9a7b59833b771036c212f9a786b5370bd458bdbb4e1cd0506dd822fa066522e"
| summarize Count = count() by UserPrincipalName, UserId, City = tostring(parse_json(LocationDetails).city), State = tostring(parse_json(LocationDetails).state), Country =
tostring(parse_json(LocationDetails).countryOrRegion)
| project UserPrincipalName, UserId, City, State, Country
| summarize PotentialImpossibleTravelInstances = count() by UserPrincipalName, UserId


| take 10
```

| UserPrincipalName | UserId | PotentialImpossibleTravelInstances |
|---|---|---|
| d9a7b59833b771036c212f9a786b5370bd458bdbb4e1cd0506dd822fa066522e@lognpacific.com | 33e90c87-bafe-401e-9a4a-357... | 2 |

## Containment, Eradication, and Recovery

This event is suspicious and warrants an update of policies and tools to prevent recurrence. In a real case scenario this account would be suspended through Azure.

This incident is classified as a “True Positive.”

This incident has had its Status set to Closed.

## Post-Incident Activities

What can be implemented is a geo-fencing policy within Azure that prevents logins outside of certain regions. Depending on corporate policy and evidence, we might immediately disable the account in Entra ID (Azure Active Directory) and contact the user or the user’s manager to investigate.