Impossible Travel / Anomalous Login

By Adam Mohmoud

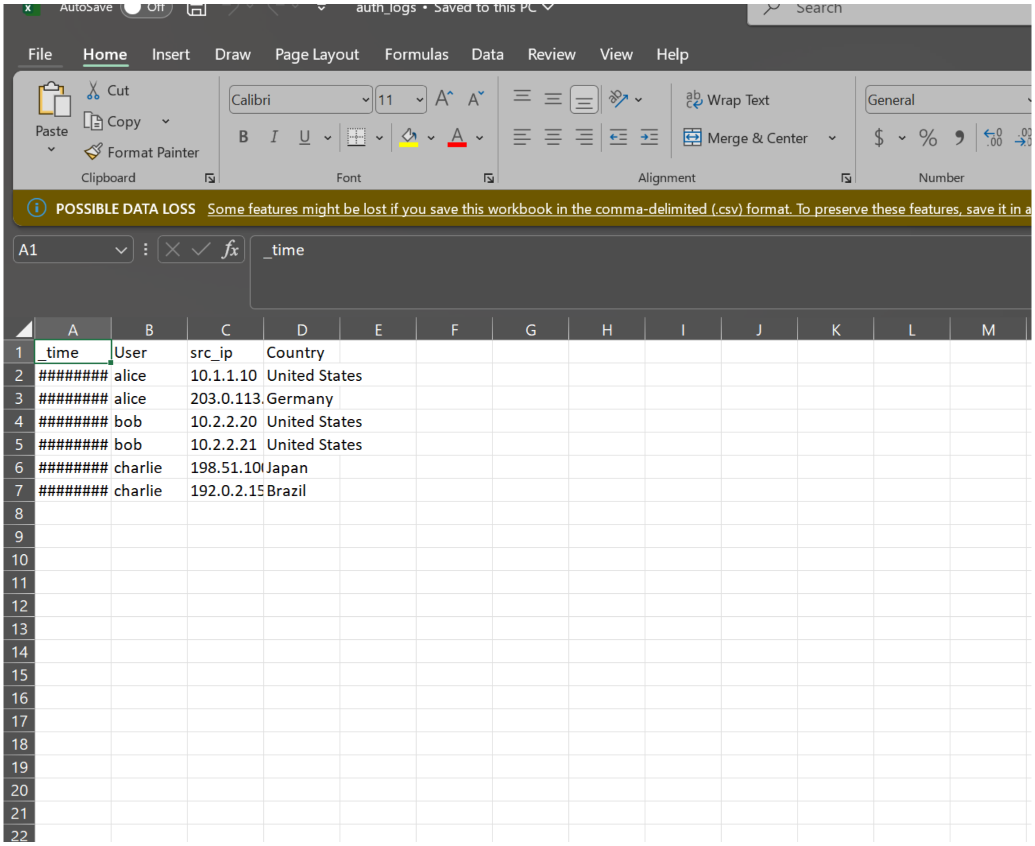
The objective of this lab is to detect impossible travel situations where a user logs in from two geographically distinct locations within a time window too short for legitimate travel. This can indicate credential compromise or unauthorized access. Using Splunk, authentication logs are ingested, and user login patterns are correlated across countries to highlight potential credential compromise or unauthorized access. This lab demonstrates stateful correlation, anomaly detection, and operational alerting.

Step 1: Dataset Preparation

This is the initial validation phase to ensure some users have consecutive logins from different countries to simulate impossible travel. The file is created as a '`.csv`' and uploaded into Splunk.

- `User`
- `src_ip`
- `Country`
- `time`

Confirming *index* = '`auth_logs`' *sourcetype* = '`csv`'



POSSIBLE DATA LOSS Some features might be lost if you save this workbook in the comma-delimited (.csv) format. To preserve these features, save it in a

_time

	A	B	C	D
1	_time	User	src_ip	Country
2	########	alice	10.1.1.10	United States
3	########	alice	203.0.113.	Germany
4	########	bob	10.2.2.20	United States
5	########	bob	10.2.2.21	United States
6	########	charlie	198.51.10(	Japan
7	########	charlie	192.0.2.15	Brazil

Step 2: Verify the Data Ingestion

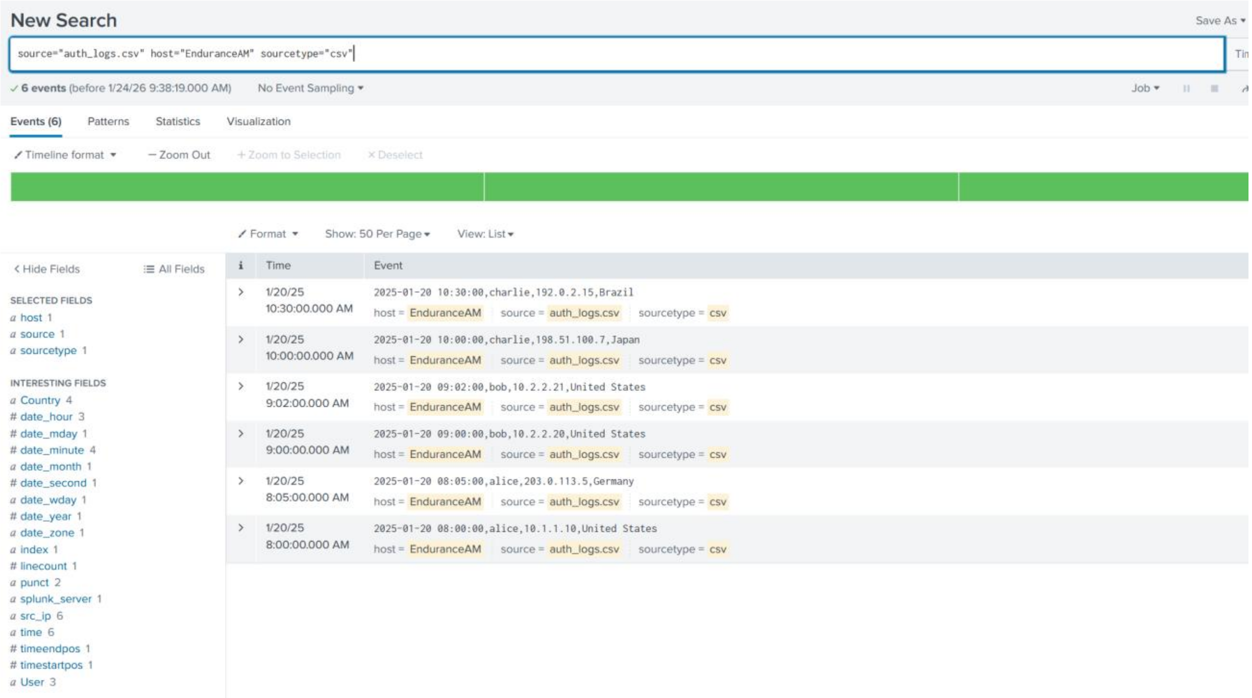
To confirm that the logs are ingested, I ran an SPL search:

```
SPL: source="auth_logs.csv" sourcetype="csv"

| table _time User src_ip Country
```

This ensures that all events appear and fields are correctly extracted.

- Data is searchable
- Fields are extracted (`User, src_ip, Country, _time`)
- Time parsing works correctly



```
source="auth_logs.csv" host="EnduranceAM" sourcetype="csv"
```

✓ 6 events (before 1/24/26 9:38:19.000 AM) No Event Sampling ▾

SELECTED FIELDS
a host 1
a source 1
a sourcetype 1

INTERESTING FIELDS
a Country 4
# date_hour 3
# date_mday 1
# date_minute 4
a date_month 1
# date_second 1
a date_wday 1
# date_year 1
a date_zone 1
a index 1
# linecount 1
a punct 2
a splunk_server 1
a src_ip 6
a time 6
# timeendpos 1
# timestartpos 1
a User 3

i	Time	Event
›	1/20/25 10:30:00.000 AM	2025-01-20 10:30:00,charlie,192.0.2.15,Brazil host = EnduranceAM source = auth_logs.csv sourcetype = csv
›	1/20/25 10:00:00.000 AM	2025-01-20 10:00:00,charlie,198.51.100.7,Japan host = EnduranceAM source = auth_logs.csv sourcetype = csv
›	1/20/25 9:02:00.000 AM	2025-01-20 09:02:00,bob,10.2.2.21,United States host = EnduranceAM source = auth_logs.csv sourcetype = csv
›	1/20/25 9:00:00.000 AM	2025-01-20 09:00:00,bob,10.2.2.20,United States host = EnduranceAM source = auth_logs.csv sourcetype = csv
›	1/20/25 8:05:00.000 AM	2025-01-20 08:05:00,alice,203.0.113.5,Germany host = EnduranceAM source = auth_logs.csv sourcetype = csv
›	1/20/25 8:00:00.000 AM	2025-01-20 08:00:00,alice,10.1.1.10,United States host = EnduranceAM source = auth_logs.csv sourcetype = csv

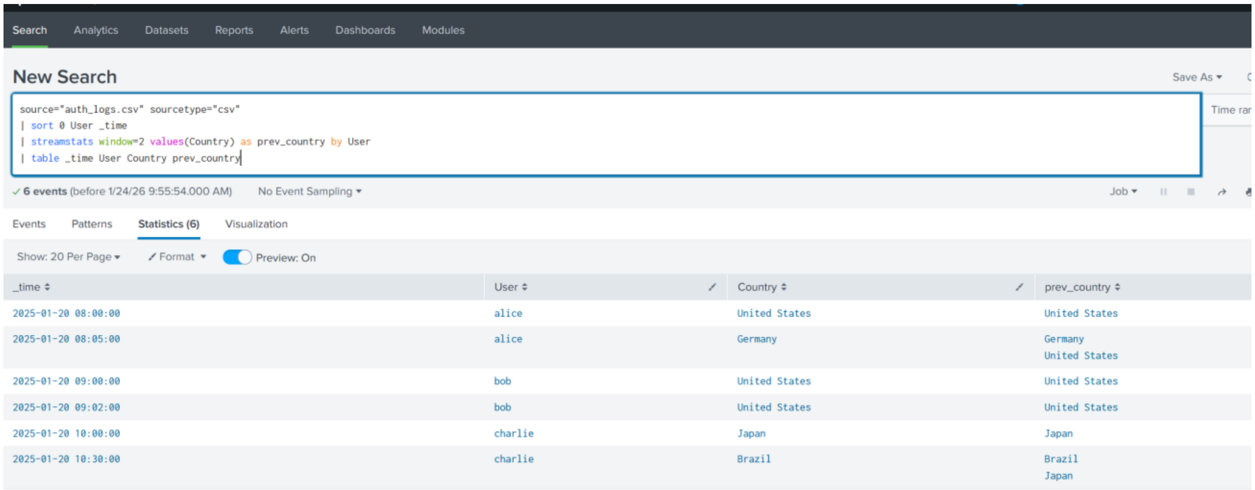
Step 3: Sort Events Per User

This action is to order the authentication events by user and timestamp. This prepares data for sequential analysis, essential for detecting impossible travel.

```
SPL: source="auth_logs.csv" sourcetype="csv"

| sort 0 User _time

| table _time User src_ip Country
```



New Search

```
source="auth_logs.csv" sourcetype="csv"
| sort 0 User _time
| streamstats window=2 values(Country) as prev_country by User
| table _time User Country prev_country
```

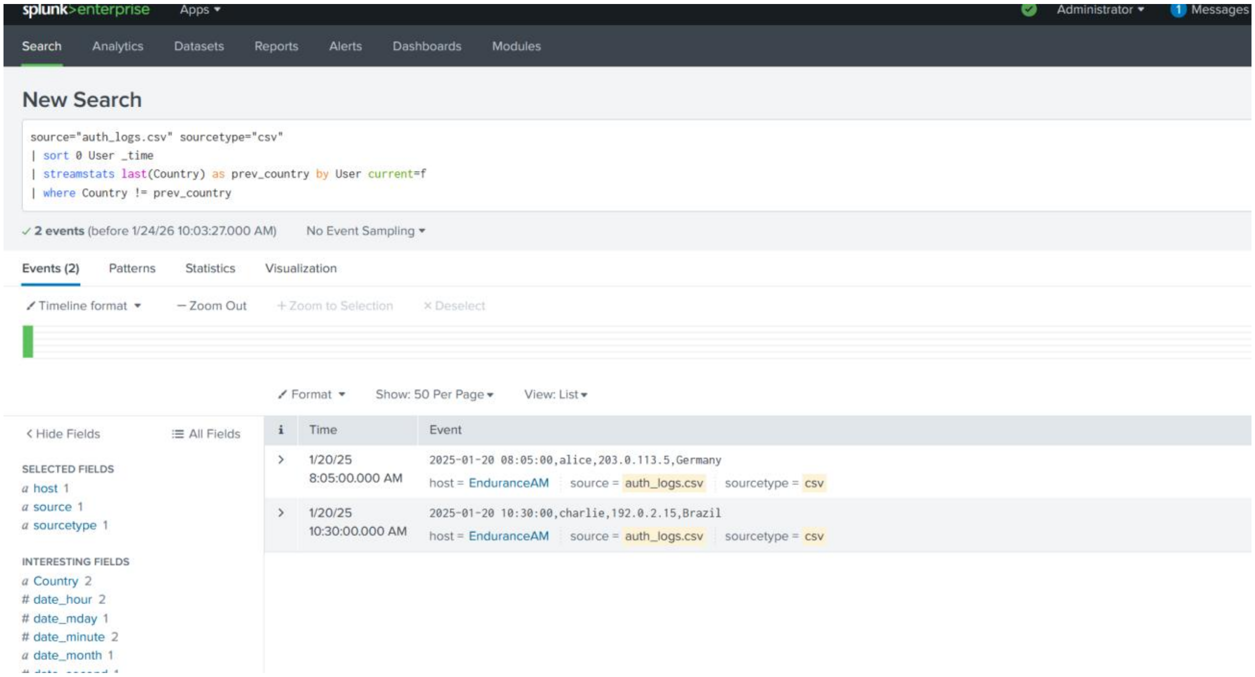
✓ 6 events (before 1/24/26 9:55:54.000 AM) No Event Sampling ▾

_time	User	Country	prev_country
2025-01-20 08:00:00	alice	United States	United States
2025-01-20 08:05:00	alice	Germany	Germany United States
2025-01-20 09:00:00	bob	United States	United States
2025-01-20 09:02:00	bob	United States	United States
2025-01-20 10:00:00	charlie	Japan	Japan
2025-01-20 10:30:00	charlie	Brazil	Brazil Japan

Step 4: Add Previous Login Context

Used ‘`streamstats`’ to track the previous login country. Each login event knows the country of the previous login for that user. This allows an analyst to detect if a user logged in from a different country immediately after the previous login resulting in the “impossible travel” scenario.

- `streamstats` tracks previous values
- `current=f` ensures only prior events are used
- `prev_country` now holds the country of the previous login



New Search

```
source="auth_logs.csv" sourcetype="csv"
| sort 0 User _time
| streamstats last(Country) as prev_country by User current=f
| where Country != prev_country
```

✓ 2 events (before 1/24/26 10:03:27.000 AM) No Event Sampling ▾

i	Time	Event
>	1/20/25 8:05:00.000 AM	2025-01-20 08:05:00,alice,203.0.113.5,Germany host = EnduranceAM source = auth_logs.csv sourcetype = csv
>	1/20/25 10:30:00.000 AM	2025-01-20 10:30:00,charlie,192.0.2.15,Brazil host = EnduranceAM source = auth_logs.csv sourcetype = csv

SELECTED FIELDS
a host 1
a source 1
a sourcetype 1

INTERESTING FIELDS
a Country 2
# date_hour 2
# date_mday 1
# date_minute 2
a date_month 1

```
source="auth_logs.csv" sourcetype="csv"

| sort 0 User _time

| streamstats last(Country) as prev_country by User current=f

| where Country != prev_country
```

Step 5: Identify Impossible Travel Events

Filter only logins (events) where the country differs from the previous country.

```
source="auth_logs.csv" sourcetype="csv"

| sort 0 User _time

| streamstats last(Country) as prev_country by User current=f

| where Country != prev_country

| table _time User prev_country Country src_ip
```

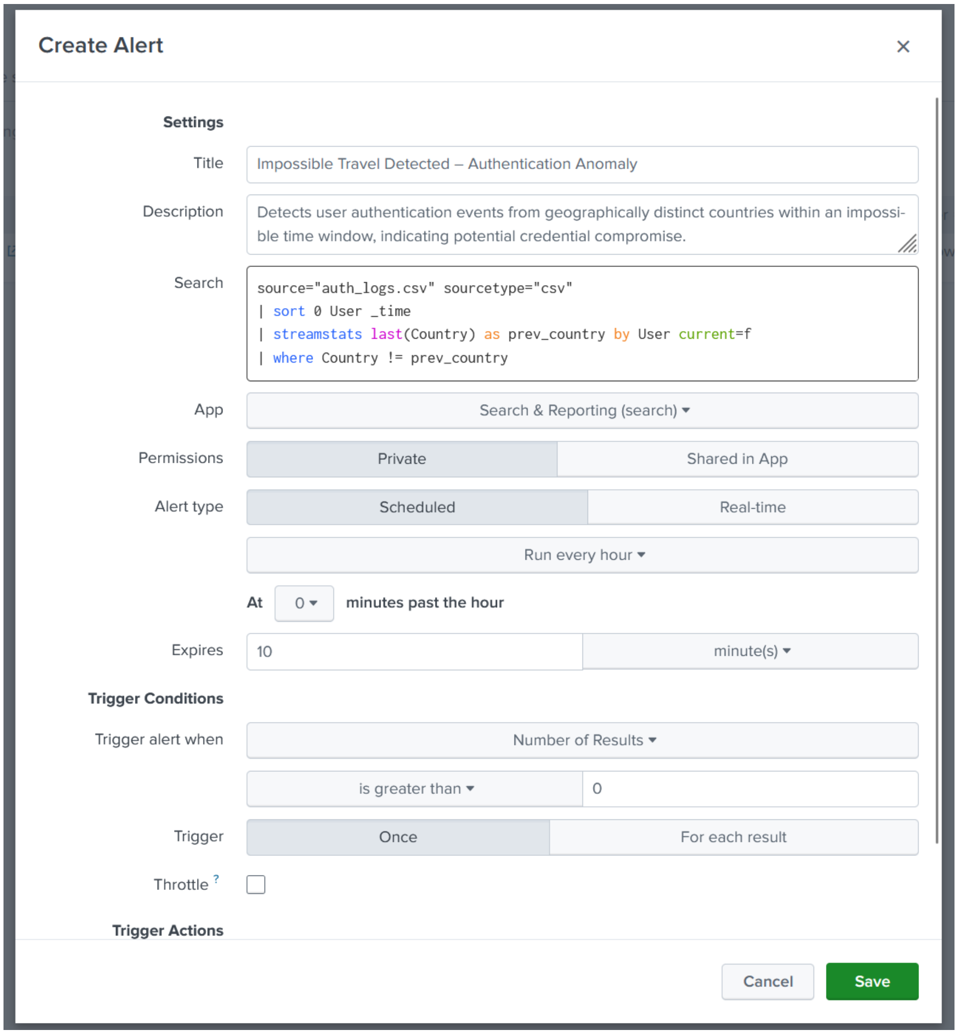
The expected result:

- Only users with “impossible travel” events appear
- Bob, who logs in from the same country, is excluded

Shows stateful correlation and anomaly detection logic.

Operationalize as Alert

The impossible travel detection was operationalized by saving the search as a scheduled Splunk alert.



Settings

Title: Impossible Travel Detected – Authentication Anomaly

Description: Detects user authentication events from geographically distinct countries within an impossible time window, indicating potential credential compromise.

Search:

```
source="auth_logs.csv" sourcetype="csv"
| sort 0 User _time
| streamstats last(Country) as prev_country by User current=f
| where Country != prev_country
```

App: Search & Reporting (search)

Permissions: [✓ Private] Shared in App

Alert type: [✓ Scheduled] Real-time

Run every hour

At 0 minutes past the hour

Expires: 10 minute(s)

Trigger Conditions

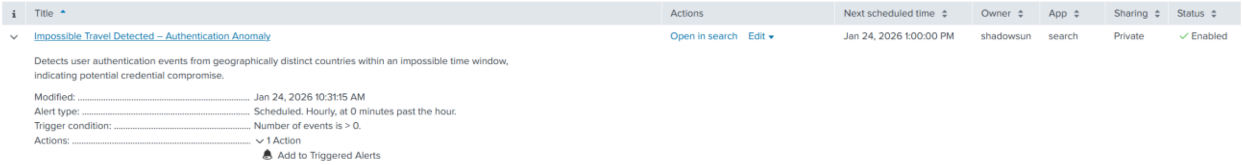
Trigger alert when: Number of Results

is greater than 0

Trigger: [✓ Once] For each result

Throttle ? □

Trigger Actions



i	Title	Actions	Next scheduled time	Owner	App	Sharing	Status
⌄	Impossible Travel Detected – Authentication Anomaly	Open in search Edit	Jan 24, 2026 1:00:00 PM	shadowsun	search	Private	✓ Enabled

Detects user authentication events from geographically distinct countries within an impossible time window, indicating potential credential compromise.

Modified: .... Jan 24, 2026 10:31:15 AM
Alert type: .... Scheduled. Hourly, at 0 minutes past the hour.
Trigger condition: .... Number of events is > 0.
Actions: .... ⌄ 1 Action
Add to Triggered Alerts

Alert Overview

Alert Name: Impossible Travel Detected/Authentication Anomaly

MITRE ATT&CK: T1078 – Valid Accounts

Severity: High

Detection Type: Behavioral / Anomaly