Detecting Malicious Insider Data Theft in IaaS Cloud

Virtualization & Cloud Computing

Group 58

Nikita

Designed and developed system to publish unauthorised login attempts to GCP VM.

Aparna

Designed and developed system to push unauthorized login attempt messages from pub/sub to BigQuery.

Bhuvana J

Designed and developed K means ML model on Bigquery data.

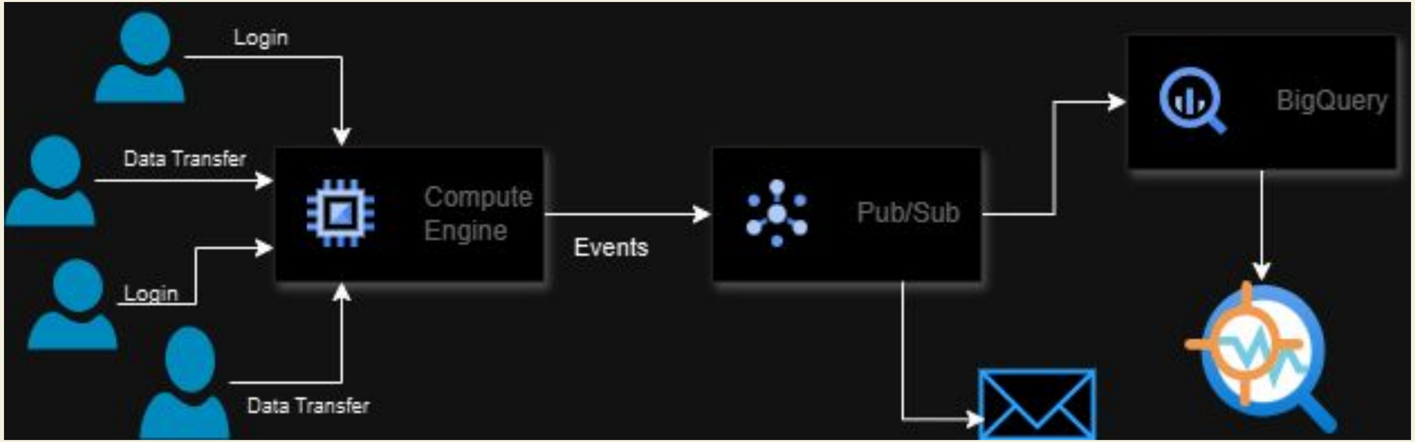
Agenda

1. **Problem statement**
2. **Solution**
3. **Architecture**
4. **Results**
5. **Conclusion and Future work**

Problem Statement

- In organizations, insider threats — malicious activities originating from within an organization by trusted users — pose significant security risks. Unlike external attacks, these threats are difficult to detect using traditional rule-based systems due to the legitimate access and behavior of insiders.
- The goal of this project is to detect potential insider threats by analyzing user login behaviors and access patterns using unsupervised learning techniques, specifically K-Means clustering, in Google BigQuery. By clustering user activity data, we aim to identify anomalous behavior that deviates significantly from typical usage patterns, such as unusual login times, access to sensitive resources, or repeated login failures.

Overall Architecture



System Overview

Detect insider threats using:

- Simulated login attempts/data transfer attack to a GCP VM
- Real-time monitoring via **Google Pub/Sub**
- Storage and analysis using **BigQuery**
- K-Means clustering for unsupervised learning.
- Google Colab for interactive development.

Why Monitor VM Logins?

- Insider attacks are often harder to detect than external threats
- Unauthorized access to VMs can lead to sensitive data theft
- Early detection relies on analyzing login behavior patterns
- Real-time visibility into login attempts is essential

Simulating Suspicious Logins

- VM simulates a **real cloud compute resource**
- Simulating real SSH access to VM generates meaningful logs
- Simulated logins help train and test the detection pipeline
- Recreating real-world scenarios (e.g., login by unknown user) helps identify pattern.
- Simulating both success and failure enables **behavioral anomaly detection.**
- Enables early-stage **model evaluation** without waiting for real breaches

Why Publish Login Events to Pub/Sub?

- Pub/Sub acts as a **real-time message pipeline**
- Decouples event producers (VM login simulation) from consumers (alert systems, dashboards, ML models)
- Enables **event-driven architecture** where login anomalies trigger alerts
- Prepares for **future scaling** across multiple VMs and users
- This integration allows centralized logging and alerting via GCP services
- Can be extended to monitor file transfers, downloads, or data exfiltration

Why Store in BigQuery?

- **BigQuery** is Google Cloud's **serverless, scalable data warehouse** that enables fast SQL queries using large datasets. It's perfect for storing structured data and running analytics at scale.
- Scalable and fast for querying large logs
- Easy integration with ML models
- SQL interface + export to visualization tools (Looker, Data Studio)
- Enables **data clustering**, **anomaly detection**, and **audit trail**

K Means

Detect Patterns in Login Events:

- Group similar login behaviors (e.g., successful logins from users, failed logins from unknown users, rapid login attempts).

Uncover Anomalies or Attacks:

- For example, if a group of login attempts is drastically different from others (e.g., lots of failures from guest roles), it may indicate a **potential intrusion or insider threat**.

Simulating Suspicious Logins

```
Listening for messages on projects/insider-detector-data/subscriptions/login-events-sub...


Received message:
Data: {"event_type": "data_transfer", "local_address": "192.168.1.10", "remote_address": "unknown.ip.address", "timestamp": 1744760568.324291}
Attributes: {}

Publishing suspicious data transfer event: {'event_type': 'data_transfer', 'local_address': '192.168.1.10', 'remote_address': 'unknown.ip.address', 'timestamp': :
Successfully published suspicious data transfer event.

Received message:
Data: {"event_type": "data_transfer", "local_address": "192.168.1.10", "remote_address": "unknown.ip.address", "timestamp": 1744760630.1215324}
Attributes: {}
```

Simulating data transfer

```
Listing for messages on projects/insider-detector-data/subscriptions/login-events-sub...


Received message:
Data: {"event_type": "data_transfer", "local_address": "192.168.1.10", "remote_address": "unknown.ip.address", "timestamp": 1744760568.324291}
Attributes: {}

Publishing suspicious data transfer event: {'event_type': 'data_transfer', 'local_address': '192.168.1.10', 'remote_address': 'unknown.ip.address', 'timestamp': 
Successfully published suspicious data transfer event.

Received message:
Data: {"event_type": "data_transfer", "local_address": "192.168.1.10", "remote_address": "unknown.ip.address", "timestamp": 1744760630.1215324}
Attributes: {}
```

BigQuery Insertion

BigQuery Table data


```
Query Results:
{'event_type': 'login', 'user': 'malicious', 'timestamp': datetime.datetime(2025, 4, 15, 23, 36, 20, 596825, tzinfo=datetime.timezone.utc), 'status': 'failure',
{'event_type': 'login', 'user': 'unknown', 'timestamp': datetime.datetime(2025, 4, 15, 23, 36, 20, 257129, tzinfo=datetime.timezone.utc), 'status': 'failure', 'r
{'event_type': 'login', 'user': 'invalid', 'timestamp': datetime.datetime(2025, 4, 15, 23, 36, 19, 915883, tzinfo=datetime.timezone.utc), 'status': 'failure', 'r
{'event_type': 'login', 'user': 'unknown_user', 'timestamp': datetime.datetime(2025, 4, 15, 23, 36, 19, 572799, tzinfo=datetime.timezone.utc), 'status': 'failure
{'event_type': 'login', 'user': 'admin', 'timestamp': datetime.datetime(2025, 4, 15, 23, 36, 19, 195427, tzinfo=datetime.timezone.utc), 'status': 'failure', 'rol
{'event_type': 'login', 'user': 'bhuvna', 'timestamp': datetime.datetime(2025, 4, 15, 23, 36, 18, 817345, tzinfo=datetime.timezone.utc), 'status': 'failure', 'ro
{'event_type': 'login', 'user': 'aparna', 'timestamp': datetime.datetime(2025, 4, 15, 23, 36, 18, 436523, tzinfo=datetime.timezone.utc), 'status': 'failure', 'ro
{'event_type': 'login', 'user': 'nikita', 'timestamp': datetime.datetime(2025, 4, 15, 23, 36, 18, 52188, tzinfo=datetime.timezone.utc), 'status': 'failure', 'rol
{'event_type': 'login', 'user': 'malicious', 'timestamp': datetime.datetime(2025, 4, 15, 23, 33, 50, 571571, tzinfo=datetime.timezone.utc), 'status': 'failure',
{'event_type': 'login', 'user': 'unknown', 'timestamp': datetime.datetime(2025, 4, 15, 23, 33, 50, 235552, tzinfo=datetime.timezone.utc), 'status': 'failure', 'r
```

K means result

```
✅ Created features table
📌 Sample rows from features table:
   role_num  status_num  timestamp_unix                        timestamp
0         0           1      1744759839 2025-04-15 23:30:39.058075+00:00
1         0           1      1744759839 2025-04-15 23:30:39.788179+00:00
2         0           1      1744759840 2025-04-15 23:30:40.128342+00:00
3         0           1      1744760029 2025-04-15 23:33:49.565607+00:00
4         0           1      1744760030 2025-04-15 23:33:50.235552+00:00
✅ Trained KMeans model
✅ Created clustered login event table
📌 Sample rows from clustered table:
  event_type       user                        timestamp   status   role  \
0      login     aparna 2025-04-15 23:36:18.436523+00:00  failure   user
1      login     bhuvna 2025-04-15 23:36:18.817345+00:00  failure   user
2      login    invalid 2025-04-15 23:36:19.915883+00:00  failure   user
3      login  malicious 2025-04-15 23:36:20.596825+00:00  failure  guest
4      login  malicious 2025-04-15 23:33:50.571571+00:00  failure  guest

   predicted_cluster
0                  1
1                  1
2                  1
3                  1
4                  1
```

Conclusion

- In this project, we successfully demonstrated an approach to detecting insider threats using K-Means clustering on user activity data stored and analyzed in **Google BigQuery**.
- **The clustering approach helped group similar behavioral patterns and distinguish outliers, such as unauthorized access attempts or unusual access times, which are critical indicators of insider threats. Our implementation showed how leveraging cloud-native tools can offer both scalability and flexibility in cybersecurity analytics.**

FutureWork

- Implement more sophisticated clustering algorithms like **DBSCAN** or **Spectral Clustering** to detect irregular patterns that K-Means might miss due to its assumptions on data distribution.
- Extend the Pub/Sub-based pipeline to trigger real-time alerts (emails, Slack notifications, etc.) when an anomaly is detected.
- With labeled data (malicious vs. legitimate activity), supervised models such as Random Forest or SVM can be trained for better precision in detecting threats.