What do you think could've happened to these people & their funds?

Wells Fargo 3.5 MILLION FAKE accounts - Insider Threat !!!

WireCard faced Accounting Fraud €1.9 BILLION was missing from its balance sheet !!!

JPMorgan encountered Security Breach - compromised 76 Million Accounts !!!
Subsequent investigations that revealed the root cause for these threats:

1. **Failed** to adequately **monitor**.

2. **Unable to Report** suspicious transactions.

3. **Allowed illicit funds** to pass through its systems!!!

4. **Failed to detect fraud** and non-transparent transactions.

5. **Failed to trace the origin** and purpose of the funds.
Financial Crime

is a broad term that encompasses a variety of illegal activities that are often associated insider threat, external threat & security breach.
Role of an SRE in Fighting Financial Crime
How do you implement these tools in Cloud to build a Secure Infrastructure?
Architecture utilises the multi-tier approach to help us identify the patterns of potential Vulnerabilities & threats by analysing the logs from these AWS equivalent services.
CloudTrailLogs - SIEM & IAM

Scenario - Insider Threat

Unauthorised employee attempted to login to Server:

The event was triggered by an "AssumedRole" identity type, which would have had a set of permissions that allowed it to run the EC2 instances, which is the "RunInstances" event mentioned in the logs.

What is an "AssumedRole" identity type?

It's a security mechanism that allows a user to assume a specific set of permissions to access AWS resources that they wouldn't normally have access to.

This is done by Temporarily granting users to access application by providing Access Key & Secret Key that they use to make API request to AWS services.
CloudWatch Logs: IDS/IPS


1. CloudWatch records various types of errors and warnings related to login attempts from a specific IP address.

2. During a DDoS attack, CloudFront leverages its distributed edge location network to distribute and handle incoming traffic across multiple edge locations. This distribution helps absorb and mitigate the impact of the attack by spreading the load and preventing a single point of failure.
Amazon Macie Logs: DLP

Scenario - Accounting Fraud
An attempts to Modify sensitive data.

1. First log entry:
   - "eventAction": WRITE - Indicates an attempt to modify the file.
   - "dataLocation" field provides details about the specific version of the file accessed.

2. Second log entry:
   - "eventAction": DELETE - Indicates an attempt to delete the file.
   - "dataLocation" field provides details about the specific version of the deleted file.

3. Classification and severity:
   - "classificationResults" field: "Highly Sensitive Data"
   - "severity": CRITICAL

4. User information:
   - "userIdentity" field: User assumed AdminRole.
   - "errorCode": "AccessDenied"
   - "errorMessage": User was not authorised to delete the resource.
Architecture utilises the multi-tier approach to help us identify the patterns of potential Vulnerabilities & Threats by analysing the logs from these AWS equivalent Tools.
What is Cap's Shield & Thor's Mjolnir for an SRE?

Aren't We the Worthy Ones?
Data Analytics and Automation
Unleashing the AWS capabilities to make our job easier

**Data storage**
Can be used to store large amounts of dataset, including VPC logs, Malice events & CloudTrail logs.

**Data analysis**
Use cluster algorithms to group similar log entries together and identify outliers that could be threat or vulnerabilities.

**Automation**
Simple schedulers and cron jobs will do the magic.
from pyspark.sql import SparkSession

# Initialize Spark Session & Read the log files
spark = SparkSession.builder.getOrCreate()
logs_df = spark.read.json("s3://my-logs-bucket/logs/ibankingapp/*")

# Apply transformations
# Example: Select specific columns and filter records
transformed_df = logs_df.select("timestamp", "user_identity_type", "event_name").filter(col("highly_sensitive_data") == "true")

# Write or Store the transformed data to a destination
transformed_df.write.csv("s3://my-output-bucket/transformed_logs/user_access.csv")

def createSparkSession()
    spark = SparkSession.builder.appName("UserAccessAnalysis").getOrCreate()

# Load the filtered log data
data = spark.read.csv("s3://my-output-bucket/transformed_logs/user_access.csv")

# Perform analysis on the filtered data
# Example: Detect suspicious login patterns and potential account compromise
result = data.groupBy("user_identity_type")
    .agg(
        count(col("event_name") == "failed_login", True).alias("failed_login_count"),
        count(col("event_name") == "access_s3_sensitive_info", True).alias("s3_sensitive_info_count"),
        count(col("event_name") == "attempt_modify_sensitive_data", True).alias("modify_sensitive_data_count"),
        collect_set(col("event_name") == "ipaddress_login", True).alias("ipaddress_login_count")
    )

# Identify potential vulnerabilities and threats
result = result.withColumn("potential_compromise",
    when(col("login_count") > col("logout_count") |
        col("failed_login_count") > 0 |
        col("s3_sensitive_info_count") > 0 |
        col("modify_sensitive_data_count") > 0), True).otherwise(False)

# Save the result to another location, such as an Amazon S3 bucket
result.write.csv("s3://my-logs-bucket/logs/ibankingapp ANALYSIS_RESULT")
THANK YOU!

@linkedin.com/in/anisha-manoharan/