Credentials Leakage Incident Report

Presented by:

Ahmed Elhalwagy
Ahmed Khaled
Ali Ibrahim
Anas Khalil
Prisklla Nabil
Yosef Khaled

Presented for: Eng. Ahmed Abulhamd

Table of

CONTENTS

Situation

A security researcher told Hamdon w Awlado's SOC Team that a file containing confidential login credentials had been posted to a public GitHub repository. An inquiry revealed that this upload was not an accident, but rather an intentional act by an employee with hostile intent, indicating an insider attack. The employee intended to divulge important company credentials, which might have resulted in unauthorized access and serious data breaches.

Report Date : 09/05/2024

SIEM Rule to Identify multiple login failure

Objective: To detect and respond to potential brute-force attacks or compromised accounts by identifying multiple login failures in a short period.
Configure SIEM to trigger an alert when the number of failed login attempts from a single IP or account exceeds a predefined threshold within a set timeframe like 5 failed attempts within 10 minutes.
Actions:

- Alert: The alert should include details such as the IP address or user account, timestamp of the attempts, and the total number of failed attempts.
- Notification: Send an immediate notification to the SOC team for manual review and further action.
- Response: Tier 1 SOC will implement basic investigation to see if it is true positive or not.

SIEM Rule To identify suspicious login times

Objective**:** To detect and investigate unauthorized access attempts by identifying logins occurring at unusual times, outside of normal working hours or established patterns.
SIEM rule that trigger alerts when logins occur outside of standard business hours, such as late at night or on weekends, particularly for high-risk accounts or administrative users.
Actions:

- Alert: For logins occurring outside of the defined business hours. The alert should include the account, timestamp of the login, and the login source.
- Notification: Send an alert to the SOC team, highlighting the unusual login times for immediate review.
- Response: Tier 1 SOC will implement basic investigation to see if it is true positive or not.

Identification

- **Unusual Geographic Logins:** Logins originating from unfamiliar or suspicious geographic locations where legitimate employees are not expected to be.
- **Failed Login Attempts:** Anomalies in the form of multiple failed login attempts, which might indicate a brute-force attack or unauthorized access attempts.
- **Login Attempts at Unusual Times:** Logins occurring during off-hours or at times when legitimate users are not typically active, raising suspicion of unauthorized access.

Scoping

- **Identify Exposed Credentials:** Identify all usernames, passwords, API keys, SSH keys, and any other sensitive information that was exposed.
- **all login credentials exposed in the public GitHub repository all are under the assumption that they are possibly hacked (in the inflicted scope)**

Containment

- **Disable Exposed Credentials:**

The team disabled all login credentials exposed in the public GitHub repository, particularly those tied to admin or privileged accounts.

- **Remove Public Repository:**

Immediate action was taken to request the removal of the public GitHub repository or file containing the exposed data.

- **Network Segmentation:**

The team isolated all systems and networks accessed by the malicious employee and enhanced monitoring to detect any unusual activity.

Eradication

Root Cause Analysis:

- **Bad implementation of user privileges:** The core issue was the poor implementation of user privileges. The employee had been granted excessive access rights, which allowed them to access, extract, and upload confidential credentials without adequate oversight. This over-privileging was a critical factor that facilitated the insider attack.
- **Weak Internal Controls and Procedures:** The organization lacked robust internal controls and procedures to prevent or detect unauthorized access and data exfiltration by employees. This weakness in internal security processes contributed to the successful execution of the insider attack.

Eradication

- **Enhance Access Controls**

Recovery

- **Reset Passwords for compromised Accounts:** reset the passwords for these accounts. Ensure that new passwords are strong, using a combination of uppercase and lowercase letters, numbers, and special characters. Avoid using easily guessable information or previously used passwords.
- **Ensure strong monitoring for these accounts**
- **Unlock the locked Accounts**
- **Configure Security Settings:** Ensure that restored accounts are configured with appropriate security settings, including multi-factor authentication (MFA) where applicable, and that they follow the principle of least privilege.

Lessons learned

- **Implement Data Loss Prevention (DLP) Solutions:**

Use DLP tools to monitor and control the movement of sensitive data within and outside the organization.
Configure DLP policies to detect and block unauthorized sharing of confidential information.

- **Conduct Regular Penetration Testing:**

Perform regular vulnerability scans to identify and remediate security weaknesses.
Conduct penetration tests to simulate insider attacks and evaluate the effectiveness of security controls.

- **Implement Multi-Factor Authentication (MFA):**

Require MFA for accessing sensitive systems and data.
Use MFA to add an extra layer of security, making it harder for insiders to misuse credentials.

- **Enable Detailed Logging and Auditing:**

Ensure detailed logging of all access to sensitive data and systems.
Regularly audit logs to detect and investigate suspicious activities.

- **Use Behavioral Analytics:**

Implement user and entity behavior analytics (UEBA) to identify deviations from normal behavior.
Use UEBA to detect potential insider threats based on unusual patterns of activity.

- **Encrypt Sensitive Data:**

Encrypt sensitive data both at rest and in transit.
Use strong encryption algorithms and manage encryption keys securely.

- **Establish a Secure Code Repository:**

Use secure code repositories with access controls to prevent unauthorized changes.
Monitor repository activities for unusual commits or access patterns.

- **Implement least privilege architecture**

Threat Hunting

- **The threat hunters began by focusing on the user who caused the data leakage:** They conducted extensive searches on GitHub to identify any repositories, commits, or activities associated with this user. This search aimed to uncover any files or data related to the compromised information. Additionally, they reviewed the user's interactions on GitHub for unusual patterns, such as frequent updates to repositories or engagement with suspicious accounts.
- **Threat hunters have traced other users activities for similar behaviors:** The threat hunters also examined the behavior of other users to identify any similar patterns. They reviewed user actions such as data access and repository modifications, looking for anomalies that might suggest insider threats. This included behaviors like accessing data outside of normal working hours or attempting to share sensitive information.
- **Hunters have looked for other data leakages on the internet for the company:** A thorough search was conducted to find additional data leaks related to the company. This involved checking various platforms, including public forums, paste sites, and data-sharing websites, for any new data dumps. Data breach monitoring tools and databases, such as Have I Been Pwned, were used to see if other parts of the company's data had been compromised.