ACME Financial Services Security Incident Report

Date of Incident: October 15, 2024

Severity: CRITICAL

Breach Types: Phishing, IDOR, SQL Injection

Section 1: Incident Analysis

1.1 Attack Summary

On October 15, 2024, an attacker **(IP: 203.0.113.45)** used the stolen login details of **user_id 1523** to attack the system. The attacker used IDOR, Email Phishing, and SQL Injection to steal sensitive customer data. This happened five days before the planned external penetration test (October 20–25), which shows that it was a real malicious attack.

Very İmportant: The attack came from IP 203.0.113.45. This IP was allowed for the planned penetration test. Because of this, the security system did not stop the attack.

1.2 Critical Event Timeline

All attack actions were done using the stolen access token from user_id 1523:

06:45:10 - Unauthorized Login: The attacker successfully logged in as user_id 1523 using previously obtained credentials.

06:46:30 - 06:47:57 - IDOR Vulnerability: The attacker changed the account ID in the API link **/api/v1/portfolio/{account_id}**. They accessed portfolio data for **15 different customers** in just 42 seconds.

09:00:23 - Phishing Attack: The attacker sent a phishing email to users (user1, user3, user5) to get backup access.

09:23:45 - SQL Injection: The attacker used **(/*!50000OR*/ 1=1--)** to bypass the Web Application Firewall and damage the system.

09:24:10 - Data Leak: All collected data, totaling 892 KB, was successfully downloaded from the system via the /dashboard/export link.

Section 2: MITRE ATT&CK Mapping

This section maps the identified malicious activities to the corresponding tactics and techniques in the MITRE ATT&CK framework, providing a common language for threat intelligence and defense planning.

Tactic	Technique ID	Technique Name	Incident Activity
Initial Access	T1078	Valid Accounts	**Unauthorized Login** using stolen credentials (user_id 1523).
Defense Evasion	T1190	Exploit Public-Facing Application	Using an obfuscated **SQL Injection** payload (/*!50000OR*/ 1=1--) to bypass the WAF.
Discovery / Privilege Escalation	T1505.004	Server Software Component: Application Abuse	Exploitation of the **IDOR vulnerability** in the API endpoint (/api/v1/portfolio/{account_id}) to access non-authorized customer data.
Initial Access / Persistence	T1566	Phishing	Attempting to establish a **backup access channel** by sending a **Phishing email** to users (user1, user3, user5).
Exfiltration	T1041	Exfiltration Over C2 Channel	Successful **Data Leak** of 892 KB via the /dashboard/export function.

3. Architecture Review

The whitelisting of the **IP:203.0.113.0** range for a future penetration test, combined with the standing policy to ”Do NOT block test traffic from scheduled IPs”, resulted in the attack bypassing the automated WAF and traffic blocking mechanisms.

3.1 Application Security Vulnerabilities

1. **IDOR:** The API did not check if the user who sent the request actually owned the account ID requested in the link. This allowed the attacker to access other users' data.
2. **SQL Injection:** The system did not use safe database commands. This allowed the attacker to use special characters to trick the database into running malicious code.
3. **WAF Vulnerabilities:** It did not have enough rules to find and stop the advanced SQL trick used by the attacker.

3.2 Attacker's Strategic Reason for Phishing

Attacker started a phishing attack hours after getting access because they needed a backup plan. 1523 account was high-risk. Getting new accounts phishing ensures the attacker can keep accessing the system even if account 1523 is blocked.

Section 4: Response & Remediation

4.1 Emergency Response Support

Account Block: user_id 1523 has been permanently blocked. The passwords for the clicking users (user1, user3, user5) have been temporarily suspended.

IP Block: The attacker's IP address 203.0.113.45 is now blocked on all network security devices.

Management Notice: All departments were officially notified of the breach at 09:35.

4.2 Long-Term Technical Fixes

Fixing SQL: All development teams must be forced to use safe database commands to prevent all SQL injection attacks.

Improving System Security: All users must immediately enable Multi-Factor Authentication.

Fixing IDOR: We need to add a security check to all API links. This check makes sure the user's token ID matches the resource ID they want to access.

4.3 Process Improvements

Monitoring Update: New security rules must be created to trigger a CRITICAL alarm if traffic comes from external testing IP (203.0.113.0) before the official test start date.

Training Update: All staff must do phishing and social engineering training more often.