

# ACME Financial Services

## Coordinated Multi-Vector Attack Analysis Report

**Analyst:** Yusuf Yurt

**Date:** 09.11.2025

### Contents

Executive Summary.....	2
Incident Timeline and Analysis.....	2
Broken Access Control (IDOR), October 15, 2024, 6:45 AM .....	2
Phishing Campaign, October 15, 2024, 9:00 AM.....	3
SQL Injection, October 15, 2024, 9:20 AM .....	4
Impact And Improvements .....	5
Impact: .....	5
Improvements: .....	5

# Executive Summary

On October 15, 2024, Acme Financial Services experienced a coordinated cyberattack that targeted several components of its trading platform. The investigation revealed that the attacker leveraged a **phishing campaign**, a **web application SQL injection**, and a **broken access control vulnerability** to gain unauthorized access to Acme Financial Services' internal systems.

The purpose of this report is to analyze these attacks, reconstruct the timeline of events, and propose security improvements to strengthen the affected systems and prevent similar incidents in the future.

## Incident Timeline and Analysis

### Broken Access Control (IDOR), October 15, 2024, 6:45 AM

On October 15, 2024, between 6:45 AM and 6:47 AM, Acme Financial Services API handled a sequence of irregular requests from the IP address of 203.0.113.45.

The attacker started the malicious activity with a legitimate authentication request to the “api/v1/login” endpoint by using the account with 1523 ID number. After the successful login, attacker started a sequence of GET requests to the multiple different portfolio endpoints.

10/15/2024 6:45	1523 /api/v1/login	POST		200	267	203.0.113.45	Acme-Mobile-Android/3.2.0	
10/15/2024 6:46	1523 /api/v1/portfolio/1523	GET	1523	200	156	203.0.113.45	Acme-Mobile-Android/3.2.0	jwt_token_1523_stolen
10/15/2024 6:47	1523 /api/v1/portfolio/1524	GET	1524	200	143	203.0.113.45	Acme-Mobile-Android/3.2.0	jwt_token_1523_stolen
10/15/2024 6:47	1523 /api/v1/portfolio/1525	GET	1525	200	138	203.0.113.45	Acme-Mobile-Android/3.2.0	jwt_token_1523_stolen
10/15/2024 6:47	1523 /api/v1/portfolio/1526	GET	1526	200	147	203.0.113.45	Acme-Mobile-Android/3.2.0	jwt_token_1523_stolen
10/15/2024 6:47	1523 /api/v1/portfolio/1527	GET	1527	200	141	203.0.113.45	Acme-Mobile-Android/3.2.0	jwt_token_1523_stolen
10/15/2024 6:47	1523 /api/v1/portfolio/1528	GET	1528	200	139	203.0.113.45	Acme-Mobile-Android/3.2.0	jwt_token_1523_stolen
10/15/2024 6:47	1523 /api/v1/portfolio/1529	GET	1529	200	144	203.0.113.45	Acme-Mobile-Android/3.2.0	jwt_token_1523_stolen
10/15/2024 6:47	1523 /api/v1/portfolio/1530	GET	1530	200	142	203.0.113.45	Acme-Mobile-Android/3.2.0	jwt_token_1523_stolen
10/15/2024 6:47	1523 /api/v1/portfolio/1531	GET	1531	200	148	203.0.113.45	Acme-Mobile-Android/3.2.0	jwt_token_1523_stolen
10/15/2024 6:47	1523 /api/v1/portfolio/1532	GET	1532	200	145	203.0.113.45	Acme-Mobile-Android/3.2.0	jwt_token_1523_stolen
10/15/2024 6:47	1523 /api/v1/portfolio/1533	GET	1533	200	140	203.0.113.45	Acme-Mobile-Android/3.2.0	jwt_token_1523_stolen
10/15/2024 6:47	1523 /api/v1/portfolio/1534	GET	1534	200	146	203.0.113.45	Acme-Mobile-Android/3.2.0	jwt_token_1523_stolen
10/15/2024 6:47	1523 /api/v1/portfolio/1535	GET	1535	200	143	203.0.113.45	Acme-Mobile-Android/3.2.0	jwt_token_1523_stolen
10/15/2024 6:47	1523 /api/v1/portfolio/1536	GET	1536	200	149	203.0.113.45	Acme-Mobile-Android/3.2.0	jwt_token_1523_stolen
10/15/2024 6:47	1523 /api/v1/portfolio/1537	GET	1537	200	141	203.0.113.45	Acme-Mobile-Android/3.2.0	jwt_token_1523_stolen
10/15/2024 6:47	1523 /api/v1/portfolio/1538	GET	1538	200	147	203.0.113.45	Acme-Mobile-Android/3.2.0	jwt_token_1523_stolen

Picture 1.1 – API Logs That Indicates The Incident

These requests returned HTTP 200 (OK), showing that the system allowed user 1523 to access other people's portfolio without authorization.

The pattern of this attack indicates automated exploitation of an **Insecure Direct Object Reference (IDOR)** vulnerability. The system failed to implement a server-side access

validation. Lastly, page 3 of the API documents mentions that the system fails to verify account ownership.

## Phishing Campaign, October 15, 2024, 9:00 AM

On October 15, 2024, at 9:00 AM, Acme Financial services faced a coordinated phishing campaign. The attacker leveraged a possible data breach about a planned pentesting that supposed to happen on October 20<sup>th</sup>. The campaign used the subject line "URGENT: Verify Your Account - Action Required" and messages that appeared to originate from security@acme-finance.com, targeting multiple employees (user1@acme.com–user6@acme.com).

10/15/2024 9:00 security@acme-finance.com	user1@acme.com	URGENT: Verify Your Account - Action Required	yes	203.0.113.45
10/15/2024 9:00 security@acme-finance.com	user2@acme.com	URGENT: Verify Your Account - Action Required	no	
10/15/2024 9:00 security@acme-finance.com	user3@acme.com	URGENT: Verify Your Account - Action Required	yes	203.0.113.45
10/15/2024 9:00 security@acme-finance.com	user4@acme.com	URGENT: Verify Your Account - Action Required	no	
10/15/2024 9:00 security@acme-finance.com	user5@acme.com	URGENT: Verify Your Account - Action Required	yes	203.0.113.45
10/15/2024 9:00 security@acme-finance.com	user6@acme.com	URGENT: Verify Your Account - Action Required	no	

Picture 2.1 – Email Log Records About the Phishing Campaign

Mail logs show delivery activity from the IP address 203.0.113.45 within the organization's approved testing range (203.0.113.0/24). This information can be found in the Security\_Test\_Schedule file. This file also contains other information regarding the planned "Manual Pentesting". The planned Manual Pentesting is supposed to start on October 20<sup>th</sup>. Also, the contractor company, CyberSec Partners, should be contacted through the email mentioned in the file, which is "pentesting@cybersecpartners.com". As we can observe in the log records, the attacker used the "security@acme-finance.com" mail address. This shows that the attacker created a fake email address using a fake domain and used this mail address to send phishing emails. Ultimately 3 of the 6 employees that targeted with this attack clicked the suspicious mail and were a victim of this attack.

10/15/2024 9:00	950107	HIGH	DETECT	203.0.113.45	/verify-account.php	Suspicious Link Pattern	no
-----------------	--------	------	--------	--------------	---------------------	-------------------------	----

Picture 2.2 – WAF Log Records Of Phishing Mails

Although the phishing emails triggered entries in the Web Application Firewall (WAF) logs, no active blocking or mitigation actions were performed. The logs simply recorded the delivery attempts and source IP activity, but the WAF did not prevent the emails from reaching the recipients' inboxes. This highlights a gap in email security controls, where detection exists but automatic response or alerting mechanisms are limited, emphasizing the need for integrated phishing protection and proactive mitigation strategies.

## SQL Injection, October 15, 2024, 9:20 AM

On October 15, 2024, between 9:18 AM and 9:30 AM, Acme Financial Services observed a series of suspicious activities originating from IP address **203.0.113.45**, targeting the **/dashboard/search** endpoint. The attacker authenticated using account **1523** and attempted multiple SQL Injection payloads, including '`OR 1=1--`', '`DROP TABLE users--`', and '`UNION SELECT * FROM users--`'. Web server logs show that some of these requests were blocked or detected by the system's Web Application Firewall (WAF), while one request containing a complex SQL pattern bypassed initial filters and returned HTTP 200.

10/15/2024 9:18	1523 /login					200	3421 203.0.113.45 Mozilla/5.0 (Windows NT 10.0; Win64; x64) Chrome/118.0
10/15/2024 9:19	1523 /dashboard					200	8934 203.0.113.45 Mozilla/5.0 (Windows NT 10.0; Win64; x64) Chrome/118.0
10/15/2024 9:20	1523 /dashboard/search	ticker=AAPL' OR 1=1--				403	567 203.0.113.45 Mozilla/5.0 (Windows NT 10.0; Win64; x64) Chrome/118.0
10/15/2024 9:21	1523 /dashboard/search	ticker=AAPL'; DROP TABLE users--				403	567 203.0.113.45 Mozilla/5.0 (Windows NT 10.0; Win64; x64) Chrome/118.0
10/15/2024 9:22	1523 /dashboard/search	ticker=AAPL' UNION SELECT * FROM users--				403	567 203.0.113.45 Mozilla/5.0 (Windows NT 10.0; Win64; x64) Chrome/118.0
10/15/2024 9:23	1523 /dashboard/search	ticker=AAPL'/*!50000OR*/1=1--				200	156789 203.0.113.45 Mozilla/5.0 (Windows NT 10.0; Win64; x64) Chrome/118.0
10/15/2024 9:24	1523 /dashboard/export	format=csv				200	892341 203.0.113.45 Mozilla/5.0 (Windows NT 10.0; Win64; x64) Chrome/118.0
10/15/2024 9:30	1523 /dashboard/home	200"				200	8934 203.0.113.45 Mozilla/5.0 (Windows NT 10.0; Win64; x64) Chrome/118.0

Picture 3.1 – SQL Injection Web Log Records

The WAF logs confirm that high-severity SQL Injection attempts were detected and blocked at 9:20, 9:21, and 9:22 AM, corresponding to attempts to manipulate query logic and exfiltrate database contents. The final request at 9:23 AM, although detected as medium-severity suspicious SQL activity, successfully returned data, indicating that the attack partially bypassed the protective controls.

timestamp	rule_id	severity	action	source_ip	uri	signature	blocked
10/15/2024 9:20	981173	HIGH	DETECT	203.0.113.45	/dashboard/search	SQL Injection Attempt - OR 1=1	yes
10/15/2024 9:21	981318	CRITICAL	BLOCK	203.0.113.45	/dashboard/search	SQL Injection - DROP TABLE	yes
10/15/2024 9:22	981257	HIGH	BLOCK	203.0.113.45	/dashboard/search	SQL Injection - UNION SELECT	yes
10/15/2024 9:23	981001	MEDIUM	DETECT	203.0.113.45	/dashboard/search	Suspicious SQL Pattern	no

Picture 3.2 – SQL Injection WAF Log Records

Analysis indicates that the attacker systematically tested input fields for common SQL Injection patterns, exploiting insufficient input validation and lack of parameterized queries in the application. The sequence of events demonstrates both reconnaissance and exploitation phases: after several blocked injection attempts against `/dashboard/search`, a crafted payload at 09:23 returned HTTP 200 with a large response (156,789 bytes), and at 09:24 the attacker successfully invoked `/dashboard/export?format=csv`, which returned HTTP 200 with 892,341 bytes — strongly suggesting the attacker exfiltrated CSV-formatted data from the export endpoint using the compromised session (account 1523).

# Impact And Improvements

## Impact:

The combined attacks observed on October 15, 2024, could have had severe consequences for Acme Financial Services:

1. **Unauthorized Access and Data Exposure:** The IDOR vulnerability allowed the attacker to access multiple users' portfolios without authorization, exposing sensitive financial data.
2. **Credential Compromise:** The phishing campaign successfully tricked three employees, potentially exposing internal credentials and enabling further unauthorized system access.
3. **Data Exfiltration:** The SQL Injection attempts and the successful CSV export demonstrate that the attacker could extract large volumes of structured data from the system. While some SQL Injection attempts were blocked by the WAF, the partial bypass indicates that unprotected endpoints remain at risk.
4. **Operational and Reputational Risk:** SQL payloads such as DROP TABLE users-- could have caused database corruption or service disruption, while successful data exfiltration could lead to regulatory compliance issues and reputational damage.

## Improvements:

### 1. Access Control and Authorization:

- Implement strict server-side validation for all endpoints, ensuring users can only access resources they own.
- Apply Role-Based Access Control (RBAC) for sensitive operations, including data exports.

### 2. Phishing and Social Engineering Mitigation:

- Enforce strong email authentication (SPF, DKIM, DMARC).
- Limit exposure of sensitive internal information, such as penetration testing schedules and vendor contacts.
- Conduct regular security awareness training for employees focused on identifying and reporting phishing attempts.

### 3. SQL Injection and Input Validation:

- Use parameterized queries or prepared statements throughout the application.
- Apply server-side input validation and escaping for all user-supplied data.
- Harden the WAF to detect blind or obfuscated SQL payloads.

#### **4. Data Export Security:**

- Restrict access to bulk export endpoints with authentication, authorization, and possibly multi-factor authentication for high-risk operations.
- Implement anomaly detection and alerting for unusually large downloads.
- Apply rate limiting to prevent automated exfiltration attempts.

#### **5. Monitoring and Incident Response:**

- Improve logging and monitoring for suspicious sequences of requests and failed attempts.
- Establish alerts for high-risk actions such as bulk exports or repeated failed queries.
- Conduct regular penetration tests and red-team exercises to verify controls.