Acme Security Incıdent Analysis Report

Preparer: Abdulbaki Taha Yeşilyurt

Date : 9.11.2025

Scope: Phishing campaign targeting employees, Web application SQL injection, Mobile API broken access control.

Executive Summary

The scope of the security review conducted is as follows:

- Phishing campaign targeting employees
- Web application SQL injection
- Mobile API broken access control

The necessary analyses have been performed, and recommendations have been provided.

A planned cyber attack was detected against the Acme Financial Services Platform on 15.10.2024. The attack successfully exposed three security vulnerabilities.

The identified vulnerabilities are stated below:

1. A social engineering email targeting users (Phishing) was identified.
2. Unauthorized access to the system was detected using an SQL Injection vulnerability that bypassed the Web Application Firewall (WAF).
3. User data exfiltration was detected utilizing a critical Broken Access Control vulnerability in the Mobile API.

API BROKEN ACCESS CONTROL

<Case 1>

As observed in the logs, requests were detected on 2024-10-15 at 01:45:10 to the `/api/v1/portfolio/` endpoint using different `user_id` values (5001-5005) via the user account with the `user_id` of “sec_team”, within the 01:45:10-01:45:30 timeframe. A detailed examination revealed that the corresponding IP address is **10.0.0.50**, confirming the request originated from an **internal address**. Furthermore, the User-Agent header contains the phrase "**Security-Scanner**". It is probable that a vulnerability scan was initiated from the local network using a security scanning tool.

2024-10-15 01:45:10	sec_team	/api/v1/portfolio/5001	GET	5001	200	123	10.0.0.50	Mozilla/5.0 (Security-Scanner)	test_token_xyz_5001
2024-10-15 01:45:15	sec_team	/api/v1/portfolio/5002	GET	5002	200	119	10.0.0.50	Mozilla/5.0 (Security-Scanner)	test_token_xyz_5002
2024-10-15 01:45:20	sec_team	/api/v1/portfolio/5003	GET	5003	200	127	10.0.0.50	Mozilla/5.0 (Security-Scanner)	test_token_xyz_5003
2024-10-15 01:45:25	sec_team	/api/v1/portfolio/5004	GET	5004	200	115	10.0.0.50	Mozilla/5.0 (Security-Scanner)	test_token_xyz_5004
2024-10-15 01:45:30	sec_team	/api/v1/portfolio/5005	GET	5005	200	121	10.0.0.50	Mozilla/5.0 (Security-Scanner)	test_token_xyz_5005

Upon reviewing the scheduled plan specified by the Security Team for vulnerability scanning, the information documented in the report is consistent with the details above.

Test Accounts:

- Account IDs: 5001-5010 (Test range)
- User: `sec_team`
- IP Range: 10.0.0.0/24

<Case 2>

As observed in the logs, on **2024-10-15 at 06:45:10**, requests were detected to the `/api/v1/portfolio/` endpoint using various `user_id` values (1523-1538) via the user account with `user_id` "**1523**", within the timeframe of 06:45:10 to 06:47:57. A detailed examination confirmed the corresponding user's IP address as **203.0.113.45**, indicating the request originated from an **external address**.

These actions expose a **Broken Access Control (IDOR)** vulnerability.

Direct access was obtained to portfolio data belonging to different users. The API's response to all these unauthorized requests with a **200 (OK)** success code proves the critical absence of an authorization control.

2024-10-15 06:45:10	1523	/api/v1/login	POST		200	267	203.0.113.45	Acme-Mobile-Android/3.2.0	
2024-10-15 06:46:30	1523	/api/v1/portfolio/1523	GET	1523	200	156	203.0.113.45	Acme-Mobile-Android/3.2.0	jwt_token_1523_stolen
2024-10-15 06:47:15	1523	/api/v1/portfolio/1524	GET	1524	200	143	203.0.113.45	Acme-Mobile-Android/3.2.0	jwt_token_1523_stolen
2024-10-15 06:47:18	1523	/api/v1/portfolio/1525	GET	1525	200	138	203.0.113.45	Acme-Mobile-Android/3.2.0	jwt_token_1523_stolen
2024-10-15 06:47:21	1523	/api/v1/portfolio/1526	GET	1526	200	147	203.0.113.45	Acme-Mobile-Android/3.2.0	jwt_token_1523_stolen
2024-10-15 06:47:24	1523	/api/v1/portfolio/1527	GET	1527	200	141	203.0.113.45	Acme-Mobile-Android/3.2.0	jwt_token_1523_stolen
2024-10-15 06:47:27	1523	/api/v1/portfolio/1528	GET	1528	200	139	203.0.113.45	Acme-Mobile-Android/3.2.0	jwt_token_1523_stolen
2024-10-15 06:47:30	1523	/api/v1/portfolio/1529	GET	1529	200	144	203.0.113.45	Acme-Mobile-Android/3.2.0	jwt_token_1523_stolen
2024-10-15 06:47:33	1523	/api/v1/portfolio/1530	GET	1530	200	142	203.0.113.45	Acme-Mobile-Android/3.2.0	jwt_token_1523_stolen
2024-10-15 06:47:36	1523	/api/v1/portfolio/1531	GET	1531	200	148	203.0.113.45	Acme-Mobile-Android/3.2.0	jwt_token_1523_stolen
2024-10-15 06:47:39	1523	/api/v1/portfolio/1532	GET	1532	200	145	203.0.113.45	Acme-Mobile-Android/3.2.0	jwt_token_1523_stolen
2024-10-15 06:47:42	1523	/api/v1/portfolio/1533	GET	1533	200	140	203.0.113.45	Acme-Mobile-Android/3.2.0	jwt_token_1523_stolen
2024-10-15 06:47:45	1523	/api/v1/portfolio/1534	GET	1534	200	146	203.0.113.45	Acme-Mobile-Android/3.2.0	jwt_token_1523_stolen
2024-10-15 06:47:48	1523	/api/v1/portfolio/1535	GET	1535	200	143	203.0.113.45	Acme-Mobile-Android/3.2.0	jwt_token_1523_stolen
2024-10-15 06:47:51	1523	/api/v1/portfolio/1536	GET	1536	200	149	203.0.113.45	Acme-Mobile-Android/3.2.0	jwt_token_1523_stolen
2024-10-15 06:47:54	1523	/api/v1/portfolio/1537	GET	1537	200	141	203.0.113.45	Acme-Mobile-Android/3.2.0	jwt_token_1523_stolen
2024-10-15 06:47:57	1523	/api/v1/portfolio/1538	GET	1538	200	147	203.0.113.45	Acme-Mobile-Android/3.2.0	jwt_token_1523_stolen

<Case 3>

As observed in the logs on **2024-10-15 at 01:30:15**, Rule **920420**, named "Multiple Failed Auth," detected malicious attempts by an attacker seeking to access different user portfolios. The attacker's request was not blocked.

2024-10-15 01:30:15	920420	LOW	DETECT	192.168.1.100	/api/v1/portfolio/1000	Multiple Failed Auth	no
2024-10-15 01:30:19	920420	LOW	DETECT	192.168.1.100	/api/v1/portfolio/1004	Multiple Failed Auth	no

<Case 4>

As observed in the logs on **2024-10-15 at 06:47:30**, Rule **942100**, named "Rapid Sequential Access," detected various attempts made by modifying the `user_id` parameter to access different user portfolios.

942100	MEDIUM	DETECT	203.0.113.45	/api/v1/portfolio/1529	Rapid Sequential Access	no
942100	MEDIUM	DETECT	203.0.113.45	/api/v1/portfolio/1534	Rapid Sequential Access	no

<Case 5>

As observed in the logs on **2024-10-15 at 06:47:57**, the rule numbered **942100**, named "Possible Account Enumeration," indicates that the attacker attempted to perform user discovery.

942100	HIGH	DETECT	203.0.113.45	/api/v1/portfolio/1538	Possible Account Enumeration	no

<Case 6>

As observed in the logs on **2024-10-15 at 08:55:00**, a request was detected to the `/admin/users/export/` endpoint via the user account with the `user_id` value of "**admin_5678**" at 08:55:00. A detailed examination confirmed the corresponding user's IP address as **10.0.1.50**, and the request was observed to originate from an **internal address**.

920430	LOW	DETECT	10.0.1.50	/admin/users/export	Admin Area Access	no

Access to the Admin endpoint was successful, and the server returned a **200 (OK)** response. The possibility of an insider threat vector should be evaluated.

2024-10-15 08:55:00	admin_5678	/admin/users/export		200	15673	10.0.1.50	Mozilla/5.0 (Windows NT 10.0; Win64; x64) Chrome/118.0
2024-10-15 08:56:30	admin_5678	/admin/download/user_export.csv		200	245890	10.0.1.50	Mozilla/5.0 (Windows NT 10.0; Win64; x64) Chrome/118.0

Solution Proposal

To prevent broken access control in APIs, enforce **strict authentication and authorization checks** for every endpoint and ensure users can only access resources they are permitted to. Regularly **review and test access policies** to identify and fix potential vulnerabilities.

PHISHING CAMPAIGN TARGETİNG EMPLOYEES

<Case 1>

As observed in the logs on **2024-10-15 at 09:00:23**, the attacker sent fraudulent emails to six users from the address **security@acme-finance.com**, impersonating the company's security team. Users **user1, user3, and user5** were detected clicking the link. All clicks were observed to redirect to the external IP address **203.0.113.45**. This event constitutes a **phishing vulnerability**. By believing the fraudulent email and clicking the link, some employees provided the attacker with the opportunity to steal their usernames and passwords.

timestamp	from	to	subject	link_clicked	ip_address	attachment
2024-10-15 08:55:12	admin@acme.com	external.contact@protonmail.com	Q3 Meeting Notes	no	10.0.1.50	meeting_notes.pdf
2024-10-15 09:00:23	security@acme-finance.com	user1@acme.com	URGENT: Verify Your Account - Action Required	yes	203.0.113.45	
2024-10-15 09:00:25	security@acme-finance.com	user2@acme.com	URGENT: Verify Your Account - Action Required	no		
2024-10-15 09:00:27	security@acme-finance.com	user3@acme.com	URGENT: Verify Your Account - Action Required	yes	203.0.113.45	
2024-10-15 09:00:29	security@acme-finance.com	user4@acme.com	URGENT: Verify Your Account - Action Required	no		
2024-10-15 09:00:31	security@acme-finance.com	user5@acme.com	URGENT: Verify Your Account - Action Required	yes	203.0.113.45	
2024-10-15 09:00:33	security@acme-finance.com	user6@acme.com	URGENT: Verify Your Account - Action Required	no		
2024-10-15 09:15:45	support@acme.com	customer1@example.com	Re: Account inquiry	no	10.0.2.30	
2024-10-15 10:30:12	hr@acme.com	all-staff@acme.com	Team Building Event Next Week	no	10.0.2.15	
2024-10-15 11:45:20	it@acme.com	engineering@acme.com	Scheduled Maintenance Tonight	no	10.0.2.25	

Solution Proposal

Provide employees with regular awareness training and realistic phishing simulations to help them recognize and quickly report suspicious emails. Technically, implement strong email

filtering, SPF/DKIM/DMARC, multi-factor authentication (MFA), and the principle of least privilege to reduce the impact of failed logins and malicious links.

WEB APPLİCATİON SQL İNJECTİON

<Case 1>

As observed in the logs on **2024-10-15 at 09:20:30**, an **SQLi (SQL Injection)** attack was detected, originating from the IP address **203.0.113.45**. The attack was blocked by the WAF

2024-10-15 09:20:30	981173	HIGH	DETECT	203.0.113.45	/dashboard/search	SQL Injection Attempt - OR 1=1	yes
2024-10-15 09:21:15	981318	CRITICAL	BLOCK	203.0.113.45	/dashboard/search	SQL Injection - DROP TABLE	yes
2024-10-15 09:22:00	981257	HIGH	BLOCK	203.0.113.45	/dashboard/search	SQL Injection - UNION SELECT	yes

<Case 2>

As observed in the logs on **2024-10-15 at 09:23:45**, a suspicious SQLi (SQL Injection) attack was attempted.

2024-10-15 09:23:45	981001	MEDIUM	DETECT	203.0.113.45	/dashboard/search	Suspicious SQL Pattern	no

<Case 3>

As observed in the logs on **2024-10-15 at 09:00:23**, a suspicious link was detected on the account verification endpoint.

2024-10-15 09:00:23	950107	HIGH	DETECT	203.0.113.45	/verify-account.php	Suspicious Link Pattern	no

Solution Proposal

To reduce the risk of SQL Injection in web applications, use prepared statements and stored procedures, and always validate and sanitize user-supplied data. Additionally, restrict unnecessary database permissions to minimize the attack surface.