INCIDENT REPORT

Alposman AVCI

09.11.2025

Section 1: Incident Analysis

1.1 Timeline reconstruction (UTC normalized)

UTC Timestamp	Layer	IP Adresi	Endpoint / E-mail	Event / Action	HTTP / WAF / E-mail Status
2024-10-15 01:30:15	API	192.168.1.100	/api/v1/portfolio/1000	Unauthorized Access	401
2024-10-15 01:30:16	API	192.168.1.100	/api/v1/portfolio/1000	Unauthorized Access	401
2024-10-15 01:30:17	API	192.168.1.100	/api/v1/portfolio/1000	Unauthorized Access	401
2024-10-15 01:30:18	API	192.168.1.100	/api/v1/portfolio/1000	Unauthorized Access	401
2024-10-15 01:30:19	API	192.168.1.100	/api/v1/portfolio/1000	Unauthorized Access	401
2024-10-15 06:47:15	API	203.0.113.45	/api/v1/portfolio/1523	Rapid Sequential Access / Token Abuse	200
2024-10-15 06:47:15	API	203.0.113.45	/api/v1/portfolio/1523	Rapid Sequential Access / Token Abuse	200
2024-10-15 08:55:00	Web	10.0.1.50	/admin/users/export	Admin Export İşlemi	200

2024-10-15 09:00:23	Email	203.0.113.45	URGENT: Verify Your Account	Phishing Link Tıklaması	Clicked / yes
2024-10-15 09:20:30	WAF	203.0.113.45	/dashboard/search	SQL Injection Attempt: OR 1=1	DETECT
2024-10-15 09:21:15	WAF	203.0.113.45	/dashboard/search	SQL Injection Attempt: DROP TABLE users	BLOCK
2024-10-15 09:22:00	WAF	203.0.113.45	/dashboard/search	SQL Injection Attempt: UNION SELECT	BLOCK
2024-10-15 09:23:45	Web	203.0.113.45	/dashboard/search	SQL Injection bypass	200
2024-10-15 09:24:10	Web	203.0.113.45	/dashboard/export	CSV Export (saldırgan)	200

1.2 Attack vector identification

The endpoints targeted in the attack are /dashboard/search, /dashboard/export, /api/v1/portfolio/<id>, /api/v1/login, and user emails. Upon observing the API log records, I believe a botnet attack was carried out by 192.168.1.100 against the /api/v1/portfolio/<id> endpoint. Additionally, a Mobile API broken access control attack was observed from the 203.0.113.45 IP address. I observed that other endpoints received high-risk attacks from the 203.0.113.45 IP address. From the email log records, I observed that a Phishing campaign targeting employees was executed and some users fell for the trap. From the WAF log records, it was observed that while threats classified as very high-risk originating from the 203.0.113.45 IP address were blocked, it did not block threats it deemed low-risk. This situation creates risk. It is visible in the web log records that the same IP address also conducted an SQL Injection attack.

1.3 Attack classification

I can say NIST. Because its fundamental functions are Identify, Protect, Detect, Respond, Recover. I believe that an attack detected through security monitoring can be blocked by writing a firewall rule, and permanent security can be established by taking measures to prevent its recurrence.

1.4 Root cause analysis

- **Lack of input validation** led to parameters being vulnerable to SQL Injection.
- **WAF policy deficiency** meant high-risk attacks were DETECTED, but some were allowed to pass.
- **Lack of token security** led to JWT token theft and exposed API access.
- **Users' lack of Phishing awareness** caused users to click on malicious links, leading to the risk of system infiltration.
- **Lack of rate limiting / anomaly detection** failed to prevent the rapid sequential access attack.

1.5 Impact assessment

Critical risks such as database manipulation (DROP TABLE attempt), data exfiltration (UNION SELECT, CSV Export), and account compromise (stolen JWT) have emerged. The system remains operational, but trust in the system is severely negatively impacted. Customer trust is lost. Highly probable outcomes include account compromise, data loss, unauthorized access, and security breaches in business and IT operations. This situation will cause a loss of confidence and larger problems for a financial institution.

Section 2: Architecture Review

2.1 Current architecture weaknesses

In the log analysis conducted, multiple weaknesses were identified in the current system architecture. In the API layer, it was observed that authentication processes are not sufficiently secure, especially that JWT tokens are not checked against IP address or session duration. This leaves the system vulnerable to token theft or replay attacks. Additionally, the lack of rate limiting or throttling controls on API endpoints enables brute-force and authorization attacks. On the web application side, the success of some SQL Injection attempt variations indicates a lack of parameterized queries or input validation mechanisms in the application. In the email infrastructure, users clicking on phishing emails from spoofed addresses shows a lack of personnel awareness and internal training. In the network layer, the WAF operating only in "detect" mode also weakens active defense. This situation allowed the attacker to conduct tests on both the API and web layers from the same IP (203.0.113.45).

2.2 Improved security architecture diagram

I believe a layered system, inspired by the OSI model, with separate security at each layer, should be designed. All requests from the external network should first be filtered at the WAF (Web Application Firewall) and Reverse Proxy layer. Here, anomalous requests, known attack signatures, and malicious IP addresses should be blocked. Then, requests are directed to the API Gateway layer. In this layer, authentication, rate limiting, and IP-based session control are provided. The application layer is designed according to secure coding principles and includes parameterized queries, ORM usage, and input validation filters. In the database layer, a system is designed where only specific roles have access.

2.3 Recommended security controls

- **User awareness training:** Reduces the success rate of social engineering attacks. All these controls directly address the root causes of the identified vulnerabilities and enhance security at both the application and user levels.
- **Parameterized queries / ORM:** Eliminates the risk of SQL Injection.
- **Short-lived, IP-bound JWT tokens:** Prevents a stolen token from being used from a different IP.
- **Running WAF policies in “block” mode:** Ensures detected attacks are automatically blocked.
- **Additionally,** firewall rules can be written to take necessary precautions, thus blocking the attacking IP address from launching repeated attacks.

2.4 Defense-in-depth strategy

While traffic is inspected at the network layer with WAF, IDS/IPS, and firewall systems, input validation, parameterized queries, session management, and MFA (multi-factor authentication) should be implemented at the application layer. At the data layer, sensitive data must be protected using encryption algorithms and role-based access control. Event monitoring and alerting processes should be continuously tracked via a SIEM system, and automated alarm systems should be established for potential threats. From a user security perspective, regular phishing simulations and security training should be conducted. Thus, a multi-layered defense chain complementing each other at the network, application, data, and human layers is established. This will not only block existing threats but also future attack vectors and will increase the organization's overall cyber resilience.

Section 3: Response & Remediation

3.1 Immediate actions (0-24 hours)

The 203.0.113.45 IP address, identified as the attack source, must be blocked at both the WAF and firewall levels. Sessions of user accounts exposed to the attack must be terminated, and their JWT tokens must be invalidated (deleted). The phishing messages from the fraudulent "security@acme-finance.com" address detected in the email logs must be blocked at the corporate email gateway, and spam filters must be updated to prevent the same messages from reaching other employees.

3.2 Short-term fixes (1-2 weeks)

First, API authorization controls should be reviewed, and matching the token holder's account ID for every portfolio request must be enforced. New rule sets should be defined on the WAF, and alert levels for attack patterns like SQL Injection, Rapid Enumeration, and Credential Stuffing should be raised. On the email side, SPF, DKIM, and DMARC policies should be reconfigured; security layers that detect phishing should be updated. During this process, user training sessions should be organized to increase phishing awareness.

3.3 Long-term improvements (1-3 months)

A behavioral detection module should be activated on the API Gateway, making unusual request patterns automatically monitorable. Additionally, the SOC (Security Operations Center) unit's threat intelligence feeds should be updated, and alert scenarios compliant with NIST techniques should be developed for the early detection of similar attacks. User email awareness will continue to be increased, aiming for all personnel to participate in phishing simulation tests within three months. DevSecOps practices should be integrated into the software development lifecycle, and secure coding standards (OWASP ASVS) should be included in internal audit procedures.

3.4 Compliance considerations

The incident has been examined within the framework of legal regulations as it involves suspicion of personal data leakage or unauthorized access. Under KVKK (Law No. 6698) valid in Turkey, measures must be taken to protect the affected user data, and the notification procedure to the Personal Data Protection Authority must be carried out if necessary. Additionally, considering the potential impact on financial systems, documentation must be updated in line with ISO/IEC 27001 Information Security Management System requirements. The incident management process must be reorganized in accordance with the NIST 800-61 Incident Response Guide and OWASP Incident Response Framework standards. The organization's compliance policy must be strengthened in terms of both regulation and customer trust, and a detailed incident report must be created for use in future audits.