SECURITY INCIDENT REPORT

Coordinated Multi-Vector Attack on Trading Platform

Incident ID:	INC-2024-1015-001	**Classification:**	**CRITICAL**
Date of Incident:	October 15, 2024	**Report Date:**	November 7, 2025
Affected Systems:	Web, API, Email	**Impacted Users:**	16 customers

EXECUTIVE SUMMARY: On October 15, 2024, Acme Financial Services experienced a coordinated three-phase cyber attack: phishing campaign compromising employee credentials, SQL injection with WAF bypass exploiting web application vulnerabilities, and BOLA (Broken Object Level Authorization) attack accessing 16 customer portfolios via API. The attack, originating from IP 203.0.113.45, resulted in unauthorized disclosure of PII and financial data, constituting a material breach under GDPR, KVKK, and financial sector regulations. Total impact: 16 customers affected, mandatory regulatory notifications required within 72 hours.

Section 1: Incident Analysis

Timeline Reconstruction (UTC Normalized)

06:00:23-31 UTC | Phase 1: Phishing Campaign
Six employees received spoofed emails from "security@acme-finance.com". Three users (user1, user3, user5) clicked malicious links (50% success rate). Email logs show attacker IP 203.0.113.45.

06:18:30 UTC | Credential Compromise
Successful login for user ID 1523 from attacker IP. 18-minute delay suggests credential harvesting via fake portal. Web logs: 200 OK, desktop Chrome user agent.

06:20:30-22:00 UTC | Phase 2: SQL Injection Attempts
Three standard SQLi payloads blocked by WAF (rules 981173, 981318, 981257). All returned 403 Forbidden.

06:23:45 UTC | WAF Bypass Success
MySQL comment obfuscation payload "ticker=AAPL'/*!50000OR*/1=1--" bypassed WAF. Rule 981001 detected but did not block. Response: 200 OK, 156KB data exfiltrated.

06:24:10 UTC | CSV Export Dashboard export downloaded 892KB of user 1523's portfolio data.

06:45:10 UTC | Phase 3: API Access
Login via mobile API using stolen credentials. User agent switched to "Acme-Mobile-Android/3.2.0".

06:46:30-47:57 UTC | BOLA Exploitation
Systematic enumeration of accounts 1523-1538 (16 total) via /api/v1/portfolio/{id}. All returned 200 OK with full portfolio details. WAF detected "Rapid Sequential Access" but did not block. 3-second intervals suggest automation.

Attack Classification & Root Cause

The attack maps to multiple frameworks: **T1566.002 Spear-phishing** **T1078 Valid Accounts** **T1190 Exploit Public App** **T1213 Data Repositories** for MITRE ATT&CK, and **A01 Broken Access Control** **A03 Injection** **A07 Auth Failures** **A09 Logging Failures** for OWASP Top 10.
OWASP API Security (2023): API1 Broken Object Level Authorization (token 1523 accessed accounts 1524-1538), API4 Unrestricted Resource Consumption (no rate limiting despite documentation claims).

Root causes include: (1) API authorization logic validates JWT tokens but fails to verify account ownership, explicitly documented as "may not verify account ownership" in API docs—a known vulnerability left unaddressed, (2) SQL injection vulnerability from string concatenation in queries rather than parameterized statements, (3) WAF operating in detect-only mode for medium-severity rules, allowing bypass exploitation to succeed, (4) Absent rate limiting enforcement despite documentation claiming limits exist, enabling automated enumeration of 16 accounts without intervention, (5) Insufficient security awareness training evidenced by 50% phishing click rate versus industry target of <5%.

Impact Assessment

> **CONFIDENTIALITY BREACH:** 16 customers' PII (full names), complete portfolio compositions, individual holdings, purchase prices, current valuations, and total portfolio values exposed. Data enables identity theft, targeted social engineering, front-running strategies, and market manipulation.

Regulatory Compliance: GDPR Article 33/34 requires 72-hour supervisory authority notification and individual customer notification. Potential fines up to €20M or 4% global revenue. KVKK Article 12 mandates Turkish DPA notification. BDDK requires financial sector incident reporting. **Financial Impact:** Direct costs forensics, legal, notifications, fines. Customer attrition risk 5-10% of affected segment.
Reputational Damage: Public disclosure of security failures impacting trust in financial services platform, potential loss of institutional clients requiring SOC 2 compliance.

Section 2: Architecture Review

Current Architecture Weaknesses

Analysis of the attack against the existing infrastructure reveals critical architectural vulnerabilities. The Email Gateway (Port 25/587) lacks advanced threat protection features such as link rewriting, sandbox detonation, or DMARC enforcement, allowing phishing emails from spoofed domain "acme-finance.com" to reach user inboxes undetected. The WAF layer, while positioned correctly between API Gateway and applications, operates in detect-only mode for medium-severity rules, creating an exploitable gap that allowed SQL injection bypass using MySQL comment obfuscation.

The Web Application (Python) and Trading API (Flask) both exhibit direct SQL database connectivity without proper input sanitization. String concatenation in query construction created the SQL injection vulnerability exploited in this incident. More critically, the Trading API implements authentication through the Auth Service but lacks authorization middleware to verify account ownership, enabling the BOLA attack that accessed 16 unauthorized portfolios. The Redis session store provides session management but lacks anomaly detection for suspicious authentication patterns (e.g., phishing click followed by login from same IP). The PostgreSQL database layer shows direct connections from multiple application tiers without database

activity monitoring or query pattern analysis. This architecture provides no visibility into anomalous queries such as those returning unusually large result sets (156KB SQL injection response). The absence of a centralized SIEM or log correlation system means that attack progression across Email Gateway → Web App → Trading API remained undetected despite comprehensive logging at each layer. The recommended defense-in-depth architecture addresses identified vulnerabilities while maintaining existing infrastructure investments:

🛡️ **Perimeter Layer Enhancements Email Gateway:** Add advanced threat protection with link rewriting (click-time URL inspection), sandbox
detonation for attachments, DMARC policy enforcement at "reject" level preventing domain spoofing. **NGWAF:** Upgrade from basic rules to next-generation WAF with ML-based anomaly detection, switch to blocking mode for HIGH/CRITICAL rules, add custom signatures for database-specific comment syntax (MySQL /*!*/, PostgreSQL /* */), implement virtual patching for zero-day vulnerabilities.

🔐 **Identity & Access Management**

Auth Service Enhancement: Implement mandatory MFA (TOTP) for all accounts, add adaptive authentication with risk scoring (device fingerprinting, geolocation analysis, impossible travel detection). **New: Authorization Service:** Deploy centralized authorization middleware between API Gateway and applications enforcing object-level permissions. Must validate user_id from JWT against resource ownership before forwarding requests to Trading API/Web App.

💻 **Application Layer Security**

Web App (Python) & Trading API (Flask): Refactor all database access to use SQLAlchemy ORM or parameterized queries eliminating string concatenation. Implement input validation middleware with whitelist approach. **API Gateway Enhancement:** Deploy rate limiting (60 req/min standard, 10 req/min auth), sequential access pattern detection (5 similar URIs with incrementing IDs triggers block), request/response validation against OpenAPI schemas. **Redis:** Add session anomaly detection module flagging suspicious patterns (rapid location changes, post-phishing authentications).

📊 **Data Layer Protection**

PostgreSQL: Deploy database activity monitoring (DAM) analyzing query patterns in real-time, detecting anomalies like unusually large result sets, suspicious UNION/OR patterns, or access to tables not typically queried by applications. Implement row-level security policies. Enable audit logging for all queries accessing PII tables. **Encryption:** TLS 1.3 for all connections, transparent data encryption (TDE) for sensitive columns.

👁️ **Detection & Response (New Layer)**

SIEM Platform: Deploy centralized log aggregation ingesting from Email Gateway, Load Balancer, API Gateway, WAF, Auth Service, Web App, Trading API, and PostgreSQL. Normalize all timestamps to UTC. Implement correlation rules: (1) Email link click + auth within 30min from same IP, (2) WAF SQLi detect + large response size >100KB, (3) >10 different portfolio IDs accessed within 5min, (4) Auth from new country + high-value transactions. **UEBA:** Establish behavioral baselines per user detecting anomalies invisible to rules. **SOAR:** Automated response for common scenarios (block IP after 3 WAF violations, force MFA step-up for suspicious auth).

Critical Security Controls (Justification)

1. Multi-Factor Authentication: Mandatory TOTP MFA prevents credential-based attacks even with stolen passwords. Addresses MITRE T1078.

2. API Authorization Middleware: Centralized service validating user_id from JWT against resource ownership before processing requests. Implementation: Authorization interceptor checking access control lists for /{resource}/{id} patterns. Remediates OWASP A01 BOLA that exposed 16 accounts. Example check:

```
if account.owner_id != jwt.user_id: return 403 Forbidden
```

3. Parameterized Queries: Eliminate string concatenation in SQL, use prepared statements. Makes WAF bypass irrelevant since query structure immutable regardless of input. Addresses OWASP A03 at root cause level.

4. WAF Hardening + Rate Limiting: Blocking mode for HIGH/CRITICAL rules, custom MySQL comment detection. API gateway enforcing strict limits with sequential access pattern detection (e.g., 5 sequential /{resource}/incrementing_id triggers block). Would have limited BOLA to <5 accounts. **5. SIEM with Correlation:** Detects attack progressions invisible to individual systems. Critical rules: (a) phishing click + auth within 30min, (b) SQLi alert + large export, (c) >10 different account IDs within 5min. Provides visibility current architecture lacks.

Section 3: Response & Remediation

Immediate Actions (0-24 Hours)

Containment priorities: (1) Block 203.0.113.0/24 subnet on all perimeter devices, (2) Terminate and suspend accounts for user 1523 and three phishing victims (user1, user3, user5) with out-of-band password reset requiring identity verification, (3) Notify 16 affected customers (accounts 1523-1538) of unauthorized portfolio access, offer credit monitoring, (4) Preserve forensic evidence by exporting all logs for October 14-16 to WORM storage for chain of custody, (5) Initiate regulatory notification process for GDPR, KVKK, and BDDK submissions, (6) Emergency WAF configuration: switch rule 981001 and all SQL injection rules to blocking mode despite false positive risk.

Short-Term Fixes (1-2 Weeks)

Week 1-2: Implement MFA. Deploy TOTP authentication for all accounts. Week 1: optional rollout for testing, Week 2: enforce requirement. **API Authorization Fix:** Deploy authorization middleware on /api/v1/portfolio/{account_id} validating account ownership against JWT user_id. Extend to /transactions and all resource endpoints. **SQL Injection Remediation:** Refactor /dashboard/search to use parameterized queries, conduct static analysis audit to identify all string-concatenated queries, prioritize by user-input exposure. **Rate Limiting:** Configure API gateway: 60/min standard, 10/min auth, sequential access detection (5 similar URIs with incrementing IDs = temp block). **Security Training:** Deploy phishing simulation using attack examples, establish reporting mechanism, target <5% click rate within 3 months.

Long-Term Improvements (1-3 Months)

Month 1: SIEM Deployment. Ingest logs from email gateway, web servers, WAF, API gateway, databases, auth services. Normalize all timestamps to UTC. Implement five critical correlation rules including phishing-to-auth, SQLi-to-export, and rapid account enumeration patterns.

Month 2: Email Security Upgrade. Deploy URL rewriting (click-time inspection), sandbox detonation for attachments, DMARC reject enforcement preventing domain spoofing. Database activity monitoring for anomalous query detection and compliance audit trails.

Month 3: UEBA Platform. Establish behavioral baselines per user, detect anomalies like unusual geolocations, atypical data access patterns, API volume spikes. API security testing integration in CI/CD pipelines testing OWASP API Top 10, quarterly manual penetration testing, mandatory security architecture review before production deployment.

Compliance Considerations

GDPR Articles 33/34 require supervisory authority notification within 72 hours and individual notifications when high risk to rights/freedoms exists (met: financial data + PII enabling identity theft). Turkish KVKK Article 12 requires KVDP notification for affected Turkish customers. BDDK expects detailed incident reports including root cause and remediation plans; may mandate security audits as operational conditions. Civil liability exposure exists despite terms of service limitations given inadequate security practices; proactive legal consultation recommended for settlement strategies limiting litigation risk. PCI-DSS Requirement 12.10 mandates incident response procedures for cardholder data exposure. Immediate actions: notify acquiring bank within 24 hours, engage PCI Forensic Investigator if cardholder data compromised, preserve audit logs per Requirement 10.5.1. Remediation timeline: address failed controls (6.5.1, 6.5.8, 8.2.1) within 90 days to avoid merchant account penalties. Card brand fines range $5,000-$100,000 monthly during non-compliance period.

Timeframe	Action	Owner	Success Criteria
0-24h	Block attacker IP, suspend accounts, notify customers	Security Ops	203.0.113.0/24 blocked, 16 customers notified
0-72h	Regulatory notifications (GDPR, KVKK, BDDK)	Legal/Compliance	Notifications submitted within deadline
1-2 weeks	Deploy MFA, fix API authorization, remediate SQLi	Dev Teams	MFA enforced, BOLA fixed, parameterized queries
1 month	SIEM with correlation rules	Security Ops	All logs integrated, 5 rules active
2 months	Email security upgrade, DB monitoring	Infrastructure	URL rewriting, DMARC reject, DB auditing
3 months	UEBA platform, API security testing	Security Ops	Behavioral baselines, CI/CD integration