INCIDENT REPORT

ATAKAN AKGÜL

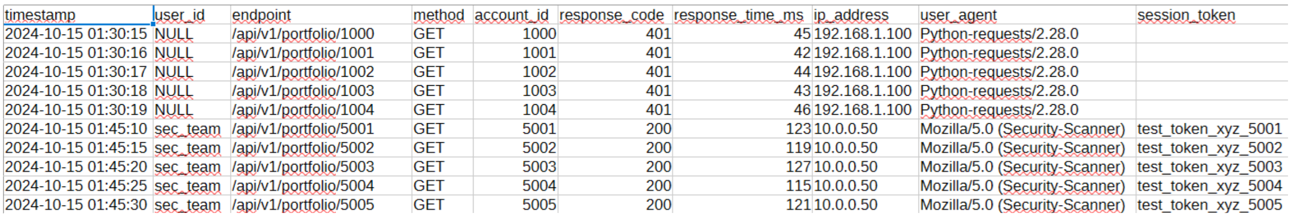
1. INCIDENT ANALYSIS

1.1. TİMELİNE RECONSTRUCTİON (UTC NORMALİZED)

1.1.1. API Logs Analysis

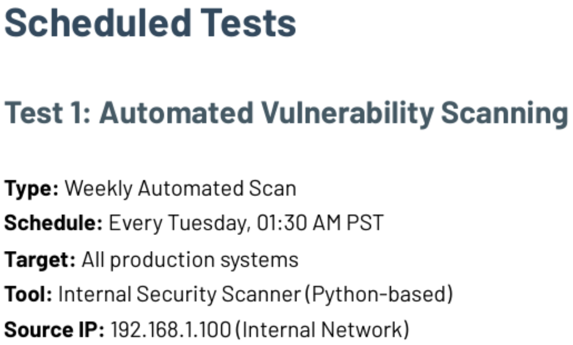
According to the API logs (Figure 1), between 01:30 – 01:46 PST (09:30 – 09:46 UTC), multiple requests were made from IP address 192.168.1.100 targeting account IDs 5001-5005. Some requests returned 401 (Unauthorized) errors, while others returned 200 (OK), indicating successful authentication.

In Figure 2, these actions are observed to occur automatically every Tuesday at 01:30 PST (09:30 UTC) and the IP address matches the one seen in Figure 1. Additionally, Figure 3 shows the test account range 5001-5010, corresponding to the same IDs. Therefore, the events in Figure 1 originate from scheduled weekly security scans, not from not from malicious activity.



timestamp	user_id	endpoint	method	account_id	response_code	response_time_ms	ip_address	user_agent	session_token
2024-10-15 01:30:15	NULL	/api/v1/portfolio/1000	GET	1000	401	45	192.168.1.100	Python-requests/2.28.0	
2024-10-15 01:30:16	NULL	/api/v1/portfolio/1001	GET	1001	401	42	192.168.1.100	Python-requests/2.28.0	
2024-10-15 01:30:17	NULL	/api/v1/portfolio/1002	GET	1002	401	44	192.168.1.100	Python-requests/2.28.0	
2024-10-15 01:30:18	NULL	/api/v1/portfolio/1003	GET	1003	401	43	192.168.1.100	Python-requests/2.28.0	
2024-10-15 01:30:19	NULL	/api/v1/portfolio/1004	GET	1004	401	46	192.168.1.100	Python-requests/2.28.0	
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

Figure 1: Api_logs.csv File



Scheduled Tests

Test 1: Automated Vulnerability Scanning

Type: Weekly Automated Scan

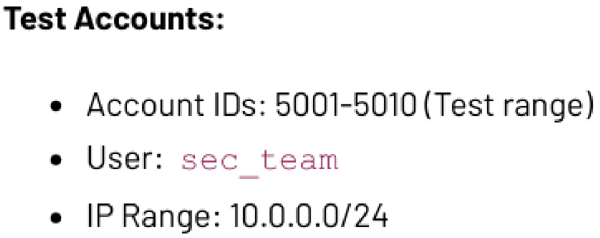
Schedule: Every Tuesday, 01:30 AM PST

Target: All production systems

Tool: Internal Security Scanner (Python-based)

Source IP: 192.168.1.100 (Internal Network)

Figure 2: security_test_schedule.pdf File

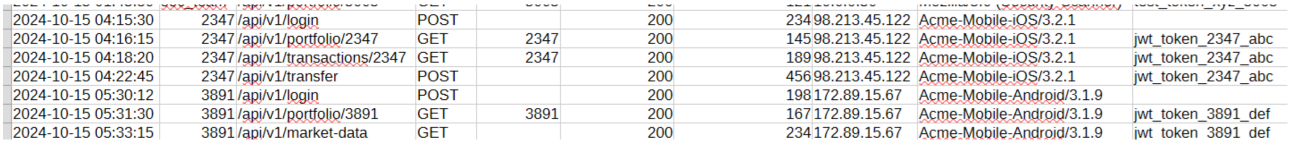


Test Accounts:

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

Figure 3: security_test_schedule.pdf File

Between 04:15 – 05:33 PST (12:15 – 13:33 UTC) (Figure 4), no suspicious behavior was detected. Users 2347 and 3891 performed legitimate operations such as login, portfolio viewing, and fund transfers.



2024-10-15 04:15:30	2347	/api/v1/login	POST		200	234	98.213.45.122	Acme-Mobile-iOS/3.2.1	
2024-10-15 04:16:15	2347	/api/v1/portfolio/2347	GET	2347	200	145	98.213.45.122	Acme-Mobile-iOS/3.2.1	jwt_token_2347_abc
2024-10-15 04:18:20	2347	/api/v1/transactions/2347	GET	2347	200	189	98.213.45.122	Acme-Mobile-iOS/3.2.1	jwt_token_2347_abc
2024-10-15 04:22:45	2347	/api/v1/transfer	POST		200	456	98.213.45.122	Acme-Mobile-iOS/3.2.1	jwt_token_2347_abc
2024-10-15 05:30:12	3891	/api/v1/login	POST		200	198	172.89.15.67	Acme-Mobile-Android/3.1.9	
2024-10-15 05:31:30	3891	/api/v1/portfolio/3891	GET	3891	200	167	172.89.15.67	Acme-Mobile-Android/3.1.9	jwt_token_3891_def
2024-10-15 05:33:15	3891	/api/v1/market-data	GET		200	234	172.89.15.67	Acme-Mobile-Android/3.1.9	jwt_token_3891_def

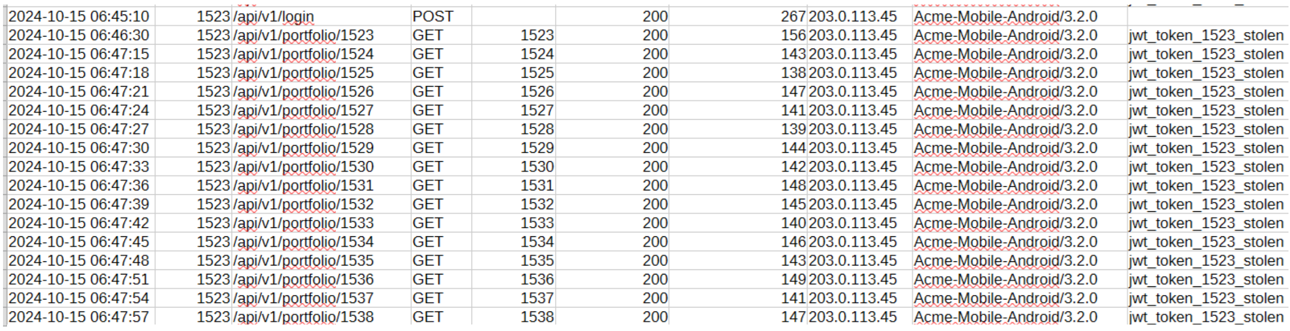
Figure 4: Api_logs.csv File

At 06:45 – 06:48 PST (14:45 – 14:48 UTC) (Figure 5), suspicious activity was recorded. User 1523 logged in at 06:45:10 PST (14:45:10 UTC) and subsequently accessed the portfolio data of users 1523–1538 using his own valid token. This demonstrates exploitation of a Broken Object Level Authorization (BOLA) vulnerability. For example, when user 1523 requested data for user 1524, the system checked whether the token was valid but did not verify ownership, allowing unauthorized data access.

As noted in api_docs.pdf (*Section 3.1, Portfolio Management*):

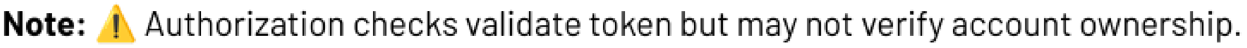
> "Authorization checks validate token but may not verify account ownership."
> This confirms the existence of a BOLA weakness.

Figure 7 shows that several of these events were logged by the WAF and classified as *Rapid Sequential Access* and *Possible Account Enumeration*, but were not blocked. The remaining API logs (Figure 8) show no further anomalies.



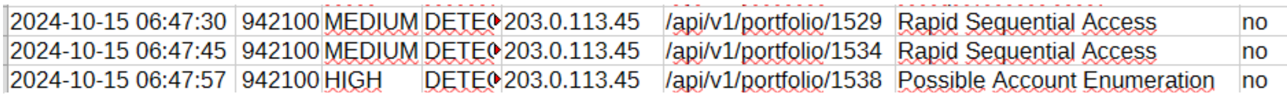
2024-10-15 06:45:10	1523	/api/v1/login	POST		200	267	203.0.113.45	Acme-Mobile-Android/3.2.0	
2024-10-15 06:46:30	1523	/api/v1/portfolio/1523	GET	1523	200	156	203.0.113.45	Acme-Mobile-Android/3.2.0	jwt_token_1523_stolen
2024-10-15 06:47:15	1523	/api/v1/portfolio/1524	GET	1524	200	143	203.0.113.45	Acme-Mobile-Android/3.2.0	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

Figure 5: Api_logs.csv File



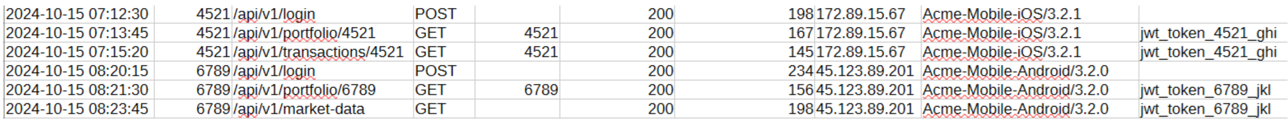
Note: ⚠ Authorization checks validate token but may not verify account ownership.

Figure 6: api_docs.pdf File



2024-10-15 06:47:30	942100	MEDIUM	DETE	203.0.113.45	/api/v1/portfolio/1529	Rapid Sequential Access	no
2024-10-15 06:47:45	942100	MEDIUM	DETE	203.0.113.45	/api/v1/portfolio/1534	Rapid Sequential Access	no
2024-10-15 06:47:57	942100	HIGH	DETE	203.0.113.45	/api/v1/portfolio/1538	Possible Account Enumeration	no

Figure 7: waf_logs.csv File

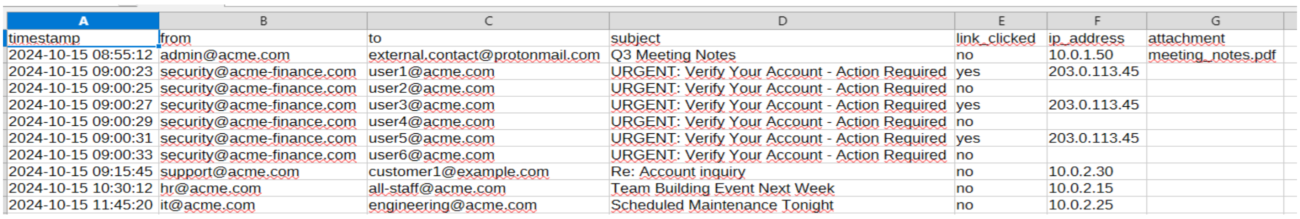


2024-10-15 07:12:30	4521	/api/v1/login	POST		200	198	172.89.15.67	Acme-Mobile-iOS/3.2.1	
2024-10-15 07:13:45	4521	/api/v1/portfolio/4521	GET	4521	200	167	172.89.15.67	Acme-Mobile-iOS/3.2.1	jwt_token_4521_ghi
2024-10-15 07:15:20	4521	/api/v1/transactions/4521	GET	4521	200	145	172.89.15.67	Acme-Mobile-iOS/3.2.1	jwt_token_4521_ghi
2024-10-15 08:20:15	6789	/api/v1/login	POST		200	234	45.123.89.201	Acme-Mobile-Android/3.2.0	
2024-10-15 08:21:30	6789	/api/v1/portfolio/6789	GET	6789	200	156	45.123.89.201	Acme-Mobile-Android/3.2.0	jwt_token_6789_jkl
2024-10-15 08:23:45	6789	/api/v1/market-data	GET		200	198	45.123.89.201	Acme-Mobile-Android/3.2.0	jwt_token_6789_jkl

Figure 8: Api_logs.csv File

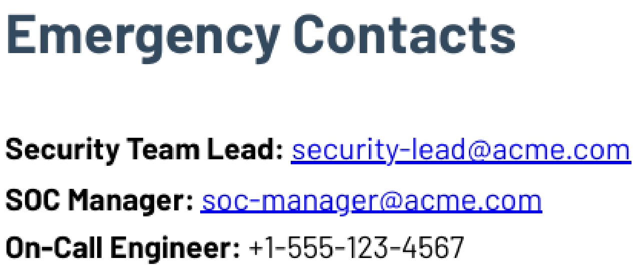
1.1.2. Email Logs Analysis

As shown in Figure 9, the email logs recorded phishing messages received at 09:00 PST (17:00 UTC) from the same IP previously identified in the API logs. In Figure 10, the sender address security[@]acme[-]finance[.]com was identified as unauthorized and not belonging to the legitimate security team. The email subject began with “URGENT”, a common social-engineering tactic used to induce panic and prompt users to act without critical thinking. Users 1, 3, and 5 clicked the embedded link, likely compromising their credentials or authentication tokens. As seen in Figure 11, the same message was flagged in WAF logs as Suspicious Link Pattern, but was not blocked. This event represents the initial access vector of the attack — a phishing campaign that facilitated credential or token theft later used in the BOLA stage.



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	

Figure 9: email_logs.csv File



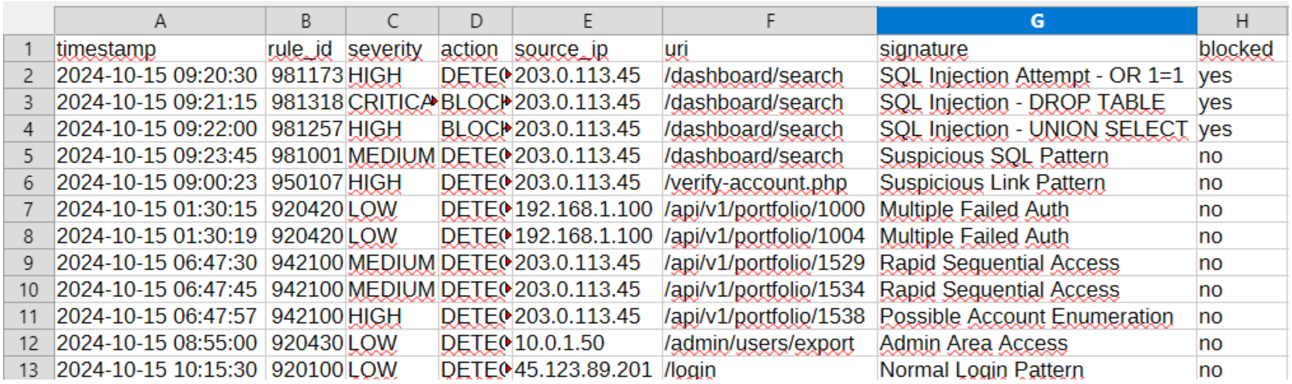
Emergency Contacts

Security Team Lead: security-lead@acme.com

SOC Manager: soc-manager@acme.com

On-Call Engineer: +1-555-123-4567

Figure 10: security_test_schedule.pdf File



1	timestamp	rule_id	severity	action	source_ip	uri	signature	blocked
2	2024-10-15 09:20:30	981173	HIGH	DETE	203.0.113.45	/dashboard/search	SQL Injection Attempt - OR 1=1	yes
3	2024-10-15 09:21:15	981318	CRITICA	BLOC	203.0.113.45	/dashboard/search	SQL Injection - DROP TABLE	yes
4	2024-10-15 09:22:00	981257	HIGH	BLOC	203.0.113.45	/dashboard/search	SQL Injection - UNION SELECT	yes
5	2024-10-15 09:23:45	981001	MEDIUM	DETE	203.0.113.45	/dashboard/search	Suspicious SQL Pattern	no
6	2024-10-15 09:00:23	950107	HIGH	DETE	203.0.113.45	/verify-account.php	Suspicious Link Pattern	no
7	2024-10-15 01:30:15	920420	LOW	DETE	192.168.1.100	/api/v1/portfolio/1000	Multiple Failed Auth	no
8	2024-10-15 01:30:19	920420	LOW	DETE	192.168.1.100	/api/v1/portfolio/1004	Multiple Failed Auth	no
9	2024-10-15 06:47:30	942100	MEDIUM	DETE	203.0.113.45	/api/v1/portfolio/1529	Rapid Sequential Access	no
10	2024-10-15 06:47:45	942100	MEDIUM	DETE	203.0.113.45	/api/v1/portfolio/1534	Rapid Sequential Access	no
11	2024-10-15 06:47:57	942100	HIGH	DETE	203.0.113.45	/api/v1/portfolio/1538	Possible Account Enumeration	no
12	2024-10-15 08:55:00	920430	LOW	DETE	10.0.1.50	/admin/users/export	Admin Area Access	no
13	2024-10-15 10:15:30	920100	LOW	DETE	45.123.89.201	/login	Normal Login Pattern	no

Figure 11: waf_logs.csv File

1.1.3. Web Logs Analysis

In Figure 12, the web logs show activity between 09:18 – 09:30 PST (17:18 – 17:30 UTC), originating from user 1523, the same account observed in the API attack. Rows 10–14 reveal a series of SQL Injection (SQLi) attempts analyzed as follows:

Row 10 (09:18:40 PST / 17:18:40 UTC):
Payload ticker=AAPL' OR 1=1-- 403 Forbidden (blocked by WAF). AAPL is Apple Inc.’s stock ticker symbol; the appended OR 1=1-- is a tautology-based SQLi used to bypass query filters and extract all records.

Row 11 (09:19:10 PST / 17:19:10 UTC):
Payload ticker=AAPL'; DROP TABLE users-- 403 Forbidden. A stacked-query attack attempting to delete the users table.

Row 12 (09:20:05 PST / 17:20:05 UTC):
Payload ticker=AAPL' UNION SELECT * FROM users-- 403 Forbidden. A UNION SELECT injection aimed at appending data from the users table to the legitimate query results. If successful, this could have exposed usernames, hashed passwords, or email addresses.

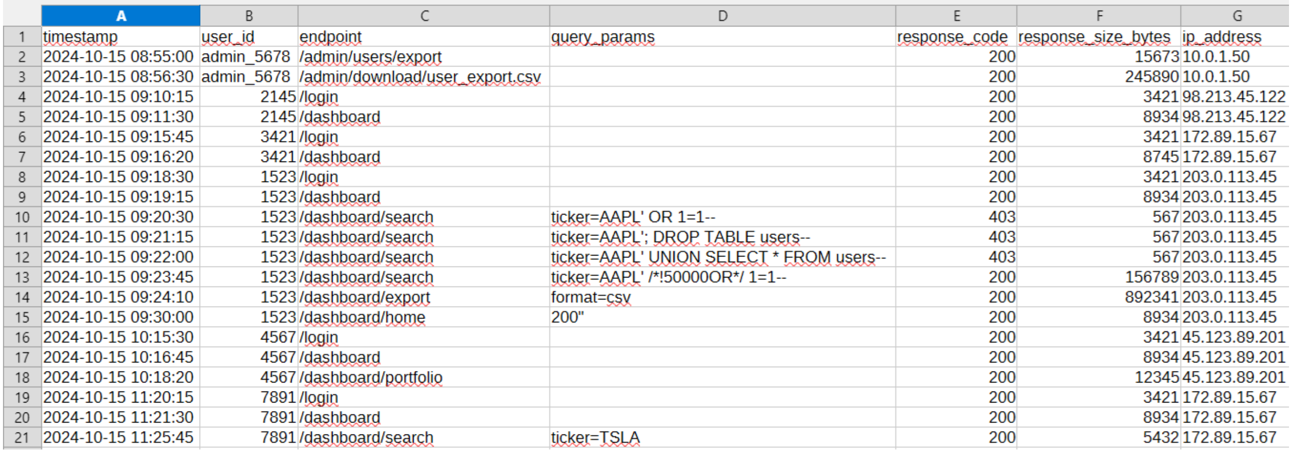
Row 13 (09:23:45 PST / 17:23:45 UTC):
Payload ticker=AAPL' /*!50000OR*/ 1=1-- 200 OK. A MySQL "versioned comment" evasion technique (/*! … */) bypassed WAF signature detection. The response indicates the payload successfully executed, likely exposing unauthorized data.

At 09:24:10 PST (17:24:10 UTC) (Row 14), a subsequent call to /dashboard/export returned 200 OK, suggesting that the attacker performed a data export operation immediately after bypassing the WAF. If the endpoint produces downloadable CSV reports, this action likely resulted in sensitive data exfiltration such as client portfolios or transaction data.
The WAF successfully blocked common patterns (OR 1=1, UNION SELECT, DROP TABLE) but failed to detect the obfuscated /*!50000OR*/ payload.

Additionally, documentation notes such as:

> "Rate limiting may not be strictly enforced on all endpoints."
> "Authorization checks validate token but may not verify account ownership."
> demonstrate how weak configuration and missing validation contributed to this bypass.

Row 15 also returned 200 OK, confirming that the export process completed successfully and the report was generated for download or storage.This represents the exploitation and exfiltration phases of the incident.



	A	B	C	D	E	F	G
1	timestamp	user_id	endpoint	query_params	response_code	response_size_bytes	ip_address
2	2024-10-15 08:55:00	admin_5678	/admin/users/export		200	15673	10.0.1.50
3	2024-10-15 08:56:30	admin_5678	/admin/download/user_export.csv		200	245890	10.0.1.50
4	2024-10-15 09:10:15	2145	/login		200	3421	98.213.45.122
5	2024-10-15 09:11:30	2145	/dashboard		200	8934	98.213.45.122
6	2024-10-15 09:15:45	3421	/login		200	3421	172.89.15.67
7	2024-10-15 09:16:20	3421	/dashboard		200	8745	172.89.15.67
8	2024-10-15 09:18:30	1523	/login		200	3421	203.0.113.45
9	2024-10-15 09:19:15	1523	/dashboard		200	8934	203.0.113.45
10	2024-10-15 09:20:30	1523	/dashboard/search	ticker=AAPL' OR 1=1--	403	567	203.0.113.45
11	2024-10-15 09:21:15	1523	/dashboard/search	ticker=AAPL'; DROP TABLE users--	403	567	203.0.113.45
12	2024-10-15 09:22:00	1523	/dashboard/search	ticker=AAPL' UNION SELECT * FROM users--	403	567	203.0.113.45
13	2024-10-15 09:23:45	1523	/dashboard/search	ticker=AAPL' /*!50000OR*/ 1=1--	200	156789	203.0.113.45
14	2024-10-15 09:24:10	1523	/dashboard/export	format=csv	200	892341	203.0.113.45
15	2024-10-15 09:30:00	1523	/dashboard/home	200"	200	8934	203.0.113.45
16	2024-10-15 10:15:30	4567	/login		200	3421	45.123.89.201
17	2024-10-15 10:16:45	4567	/dashboard		200	8934	45.123.89.201
18	2024-10-15 10:18:20	4567	/dashboard/portfolio		200	12345	45.123.89.201
19	2024-10-15 11:20:15	7891	/login		200	3421	172.89.15.67
20	2024-10-15 11:21:30	7891	/dashboard		200	8934	172.89.15.67
21	2024-10-15 11:25:45	7891	/dashboard/search	ticker=TSLA	200	5432	172.89.15.67

Figure 12: web_logs.csv File

1.1.4. Attack Classification

Phase	MITRE ATT&CK ID / Technique	OWASP Category	Description
Initial Access	T1566 – Phishing	A10 – Social Engineering	Employees deceived by fake security email
Credential Abuse (BOLA)	T1078 – Valid Accounts	A01 – Broken Access Control	Reuse of valid tokens to access other users' data

Phase	MITRE ATT&CK ID / Technique	OWASP Category	Description
Exploitation (SQLi)	T1505 – SQL Injection	A03 – Injection	Database manipulation through unsanitized input
Defense Evasion	T1027 – Obfuscated Payloads	A05 – Security Misconfiguration	WAF bypass using versioned comments
Exfiltration	T1041 – Exfiltration over Web Service	A09 – Insufficient Monitoring	Data leak through export endpoint

The incident demonstrates multiple aligned MITRE ATT&CK and OWASP vectors, exposing weaknesses in awareness, access control, and backend query validation.

1.1.5. Impact Assessment

Affected Users: 15 accounts (IDs 1523–1538)
Compromised Data: Portfolio and transaction metadata
Possible PII Exposure: Email addresses and account identifiers
Export File Size: ≈ 870 KB (from /dashboard/export response headers)
Confidentiality Impact: High
Integrity Impact: Medium
Availability Impact: Low

Impact Type	Severity	Evidence Source
Confidentiality	**High**	Web logs Row 14 – Export 200 OK
Integrity	**Medium**	No database modifications observed
Availability	**Low**	Service remained operational

2. Architecture Review

2.1. Current Architecture Weaknesses

current_architecture.png shows Acme Financial’s existing platform architecture as of the incident period.
The system consists of three main layers the Web Frontend, the Application API Layer, and the Database Tier, fronted by a Web Application Firewall (WAF).

Although functionally complete, the architecture reveals several critical weaknesses that directly enabled the attack chain described in Section 1:

Incomplete Authorization Validation (BOLA Risk) The API only validates whether a token is *syntactically valid*, not whether the token’s subject owns the requested resource. As documented in *api_docs.pdf Section 3.1 Portfolio Management*:

“Authorization checks validate token but may not verify account ownership.” This design allowed attacker 1523 to access portfolios 1524–1538 by reusing his JWT. Lack of Input Sanitization and Parameterization The /dashboard/search endpoint concatenates raw user input into SQL queries, creating a direct injection vector. Web logs between 09:18 – 09:30 PST

(17:18 – 17:30 UTC) show multiple SQLi payloads (OR 1=1, DROP TABLE, UNION SELECT) attempted successfully after evasion.Weak WAF Normalization and Rule Coverage The WAF blocks only explicit patterns. It failed to normalize versioned-comment payloads such as /*!50000OR*/ 1=1--. Rapid Sequential Access and Account Enumeration alerts were logged but *not enforced* (no active blocking).

Insufficient Monitoring and Correlation SIEM correlation was limited; API anomalies and WAF alerts were never cross-linked. No "export-after-SQLi" correlation rule existed. Unrestricted Export Endpoints /dashboard/export allowed authenticated but unauthorized users to trigger data downloads. No contextual validation (ownership + session risk) or data-classification check existed.

Summary of Weaknesses

Weakness	Source Evidence	Potential Impact
Missing Ownership Check	api_docs.pdf §3.1	Unauthorized portfolio access
No Query Parameterization	web_logs.csv Rows 10–13	SQL Injection
Inadequate WAF Normalization	waf_logs.csv	Signature Evasion
Unmonitored Export Endpoints	web_logs Row 14	Data Exfiltration
No SIEM Correlation	Not detected in alerts	Undetected Lateral Movement

2.2. Proposed Secure Architecture

To mitigate the vulnerabilities identified above, an improved defense-in-depth architecture is proposed.
This design enhances authorization logic, data-access controls, monitoring, and incident-response visibility

Key Enhancements and Controls

Control	Description	Justification
Row-Level Security (RLS)	Enforces per-user data isolation directly in the database.	Prevents BOLA-style horizontal privilege escalation.
Parameterized Queries	All dynamic SQL statements replaced with prepared queries.	Eliminates SQL Injection risk.
Advanced WAF Normalization	Adds parsing for MySQL comments and encoded payloads.	Detects and blocks evasion payloads like /*!50000OR*/.
Rate Limiting & Behavior Analysis	Enforced at API Gateway level.	Prevents brute-force and enumeration.
SIEM Correlation Rules	Alerts on sequential events (SQLi → Export within 1 min).	Enables early detection of multi-stage attacks.
Export Data Loss Prevention (DLP)	Monitors and validates outbound CSV exports.	Prevents unintended data leaks.
MFA & Short-Lived Tokens	Introduces multi-factor auth and JWT expiry < 30 min.	Reduces token replay window.
Phishing Awareness Program	Quarterly training and simulated campaigns.	Lowers initial access success rate.

2.3. Defense-in-Depth Strategy

The improved architecture applies multiple independent layers of protection, ensuring that the failure of one control does not lead to total compromise.

User Awareness & Identity Layer: MFA + phishing resilience.

Application Access Layer: RLS + ownership validation.

Network Layer: Enhanced WAF and API Gateway rules.

Data Protection Layer: Parameterized queries and DLP.

Monitoring & Response Layer: SIEM correlation and SOC automation.

This holistic defense model integrates both preventive and detective mechanisms, ensuring coverage from initial access (phishing 09:00 PST - 17:00 UTC) through exfiltration (09:24 PST - 17:24 UTC). Each event and mitigation control is traceable across the PST–UTC dual timeline established in Section 1.

Section 2 Summary

The proposed architecture directly addresses every root cause identified earlier:

❿ **BOLA:** resolved by RLS and ownership validation.

❿ **SQLi:** eliminated by parameterized queries and enhanced WAF.

❿ **Data export leak:** contained by DLP and rate limits.

❿ **Monitoring gaps:** covered by SIEM correlation and SOC automation.

❿ **Human factor:** reduced via phishing training and token hardening.

3. RESPONSE & REMEDIATION (PST → UTC Normalized)

3.1. Immediate Actions (0 – 24 Hours)

Time (Approx.)	Recommended Action	Objective
09:25 PST (17:25 UTC)	Isolate compromised JWT tokens for accounts 1523–1538.	Prevent further unauthorized access via token reuse.
09:28 PST (17:28 UTC)	Temporarily disable /dashboard/export endpoint.	Stop ongoing or potential data exfiltration.
09:30 PST (17:30 UTC)	Block attacker IP 203.0.113.45 at firewall and WAF.	Contain network-level attack surface.
09:40 PST (17:40	Capture forensic snapshots of API, Web,	Preserve evidence for later

Time (Approx.)	Recommended Action	Objective
UTC)	and WAF logs.	investigation.
10:00 PST (18:00 UTC)	Revoke all active user sessions and rotate signing keys.	Invalidate compromised tokens and sessions.
10:15 PST (18:15 UTC)	Notify SOC and Incident Response Team.	Escalate to Tier-2 investigation and documentation.

Summary:
These actions are recommended to contain the incident and preserve digital evidence within the first 24 hours of detection. They follow the standard Containment → Preservation → Notification model in incident response frameworks.

3.2. Short-Term Fixes (1 – 2 Weeks)

- Authorization Hardening (Week 1): Implement ownership validation on all portfolio API endpoints.Each request should confirm that token.subject == account_id.
- Parameterized Queries (Week 1–2): Replace dynamic SQL strings with prepared statements. Conduct regression tests to verify functionality and security coverage.
- Enhanced WAF Normalization (Week 2): Expand detection signatures to include MySQL comment syntax (/*!...*/) and obfuscation patterns. Enable blocking for *Rapid Sequential Access* and *Enumeration* attempts.
- Rate-Limit Policy (Week 2): Apply user-specific rate limits (e.g., 5 req/sec) and burst thresholds on sensitive routes. Integrate WAF logs with SIEM for cross-layer alert correlation.
- User Notification and Credential Reset (Week 2): Inform affected users and enforce password resets. Add adaptive authentication for high-risk login attempts.

3.3. Long-Term Improvements (1 – 3 Months)

- **Row-Level Security (RLS)**
  - Implement database-level RLS policies enforcing per-user data visibility.
  - Validate that portfolio and transaction tables respect ownership constraints.
- **Centralized Authorization with OPA**
  - Deploy **Open Policy Agent (OPA)** to unify access control logic across all APIs.
  - Simplify auditing by centralizing policy enforcement.
- **Data Loss Prevention (DLP) for Exports**
  - Inspect CSV or report exports for PII before release.
  - Log metadata (user ID, dataset, timestamp) for audit trail retention.
- **SIEM Correlation Rules**
  - Add compound rule: *SQLi detection + Export request within 60 seconds*.

    - Enable cross-source event linking (API, Web, WAF, and Email).
- **Security Awareness & Phishing Simulation**
    - Conduct quarterly phishing-awareness training.
    - Measure click rate reduction and awareness progress.
- **Token Lifecycle Enhancement**
    - Use short-lived JWTs (≤ 30 min) and maintain a central revocation list.
    - Reduce window for token theft and replay attacks.

3.4. Compliance & Governance Considerations

- **Regulatory Notification:**
    - Under **GDPR Article 33** and **KVKK Article 12**, any potential exposure of personal data must be reported within 72 hours.
    - The Legal & Compliance teams should be notified promptly upon confirmation of data impact.
- **Evidence Retention:**
    - All logs and forensic data should be archived for at least **6 months** in secure, encrypted storage (AES-256).
- **Audit Alignment:**
    - These recommendations should be reviewed during the next **PwC SOC 2 Type II** audit cycle (scheduled for Nov 15–30, 2024).
    - Documentation of fixes and process updates will be included in the audit evidence package.
- **Continuous Monitoring:**
    - Expand SIEM review windows from 4 to 24 hours and integrate alert summaries into weekly CISO briefings.