1.Executive Summary:

Unusual activities associated with the IP address 203.0.113.45 have been detected. On 15.10.2024 at 06:45:10, a successful login occurred with user ID 1523, followed by unauthorized GET requests using the stolen token jwt_token_1523_stolen to various account_ids from the same IP. The API's return of 200 OK responses reveals an access control vulnerability. At 09:00:23, phishing emails were sent (opened by user1, user3, and user5); at 09:18:30, a web session was initiated. Starting at 09:20:30, SQL injection attempts (OR 1=1--, DROP TABLE, UNION SELECT) were blocked by the WAF, but an obfuscated variant (/*!50000OR*/) bypassed the protection, resulting in the exfiltration of 892,341 bytes of data via /dashboard/export?format=csv. WAF logs confirm blocking events between 09:20:30–09:22:00, followed by the bypass. According to the security_test_schedule.pdf document, this IP is allocated for an authorized penetration test scheduled for 20–25 October. However, the activities began 5 days early, with initial access occurring via the mobile API application. This discrepancy raises the possibility of an insider threat or compromise of the pentest infrastructure. Consequently, the incident cannot be classified as part of the planned test; it constitutes a genuine security breach.

2.Timeline:

Time/Date (UTC)	Aktivite	Kaynak
15.10.2024 01:30:15	Starting scans with Python-requests/2.28.0 on the IP address 192.168.1.100	api_logs
15.10.2024 01:30:19	Completing scans with Python-requests/2.28.0 on the IP address 192.168.1.100	api_logs
15.10.2024 01:45:10	Making a GET request from IP address 10.0.0.50 to account_id 5001	api_logs
15.10.2024 01:45:15	Making a GET request from IP address 10.0.0.50 to account_id 5002	api_logs
15.10.2024 01:45:20	Making a GET request from IP address 10.0.0.50 to account_id 5003	api_logs
15.10.2024 01:45:25	Making a GET request from IP address 10.0.0.50 to account_id 5004	api_logs
15.10.2024 01:45:30	Making a GET request from IP address 10.0.0.50 to account_id 5005	api_logs
15.10.2024 06:45:10	Requesting /api/v1/login from IP 203.0.113.45	api_logs
15.10.2024 06:46:30	Viewing the portfolio of account_id 1523 with the request /api/v1/portfolio/1523 from the IP address 203.0.113.45	api_logs
15.10.2024 06:47:15	Viewing the portfolio of account_id 1524 with the request /api/v1/portfolio/1524 from the IP address 203.0.113.45	api_logs
15.10.2024 06:47:18	Viewing the portfolio of account_id 1525 with the request /api/v1/portfolio/1525 from the IP address 203.0.113.45	api_logs
15.10.2024 06:47:21	Viewing the portfolio of account_id 1526 with the request /api/v1/portfolio/1526 from the IP address 203.0.113.45	api_logs
15.10.2024 06:47:24	Viewing the portfolio of account_id 1527 with the request /api/v1/portfolio/1527 from the IP address 203.0.113.45	api_logs
15.10.2024 06:47:27	Viewing the portfolio of account_id 1528 with the request /api/v1/portfolio/1528 from the IP address 203.0.113.45	api_logs
15.10.2024 06:47:30	Viewing the portfolio of account_id 1529 with the request /api/v1/portfolio/1529 from the IP address 203.0.113.45	api_logs
15.10.2024 06:47:33	Viewing the portfolio of account_id 1530 with the request /api/v1/portfolio/1530 from the IP address 203.0.113.45	api_logs
15.10.2024 06:47:36	Viewing the portfolio of account_id 1531 with the request /api/v1/portfolio/1531 from the IP address 203.0.113.45	api_logs
15.10.2024 06:47:39	Viewing the portfolio of account_id 1532 with the request /api/v1/portfolio/1532 from the IP address 203.0.113.45	api_logs
15.10.2024 06:47:42	Viewing the portfolio of account_id 1533 with the request /api/v1/portfolio/1533 from the IP address 203.0.113.45	api_logs
15.10.2024 06:47:45	Viewing the portfolio of account_id 1534 with the request /api/v1/portfolio/1534 from the IP address 203.0.113.45	api_logs
15.10.2024 06:47:48	Viewing the portfolio of account_id 1535 with the request /api/v1/portfolio/1535 from the IP address 203.0.113.45	api_logs
15.10.2024 06:47:51	Viewing the portfolio of account_id 1536 with the request /api/v1/portfolio/1536 from the IP address 203.0.113.45	api_logs
15.10.2024 06:47:54	Viewing the portfolio of account_id 1537 with the request /api/v1/portfolio/1537 from the IP address 203.0.113.45	api_logs

15.10.2024 06:47:57	Viewing the portfolio of account_id 1538 with the request /api/v1/portfolio/1538 from the IP address 203.0.113.45	api_logs
15.10.2024 09:00:23	User user1@acme.com clicks on the phishing email	email_logs
15.10.2024 09:00:27	User3@acme.com clicks on the phishing email	email_logs
15.10.2024 09:00:31	User5@acme.com clicks on the phishing email	email_logs
15.10.2024 09:18:30	/login request from IP 203.0.113.45	web_logs
15.10.2024 09:19:15	Switching to /dashboard with IP 203.0.113.45	web_logs
15.10.2024 09:20:30	Running the command ticker=AAPL' OR 1=1– in /dashboard/search with the IP address 203.0.113.45	web_logs
15.10.2024 09:21:15	Running the command ticker=AAPL'; DROP TABLE users-- in /dashboard/search with the IP address 203.0.113.45	web_logs
15.10.2024 09:22:00	Running the command ticker=AAPL' UNION SELECT * FROM users– in /dashboard/search with the IP address 203.0.113.45	web_logs
15.10.2024 09:23:45	Running the command ticker=AAPL' /*!50000OR*/ 1=1-- in /dashboard/search with the IP address 203.0.113.45	web_logs
15.10.2024 09:24:10	Performing format=csv operation with /dashboard/export with IP address 203.0.113.45	web_logs
15.10.2024 09:30:00	Opening the /dashboard/home panel with the IP address 203.0.113.45 and the account_id 1523	web_logs
15.10.2024 09:20:30	Attempting SQL injection into /dashboard/search with IP address 203.0.113.45	waf_logs
15.10.2024 09:21:15	An attempt was made to delete a table using SQL injection in the /dashboard/search section using the IP address 203.0.113.45.	waf_logs
15.10.2024 09:22:00	Data extraction was performed via SQL injection from /dashboard/search using the IP address 203.0.113.45.	waf_logs
15.10.2024 09:23:45	The /dashboard/search area was accessed at the IP address 203.0.113.45. A Suspicious SQL pattern was detected in the WAF logs.	waf_logs
15.10.2024 09:00:23	The /dashboard/search domain was accessed at the IP address 203.0.113.45. It was detected as a Suspicious Link Pattern in the WAF logs.	waf_logs

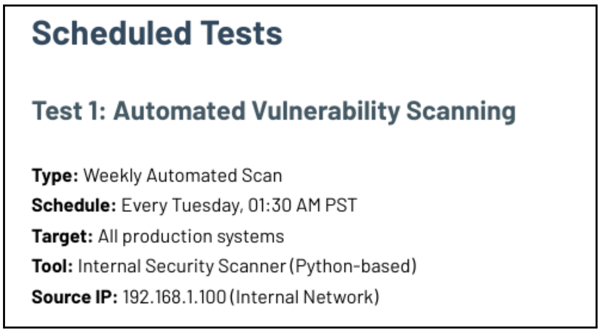
3. Technical Analysis:

3.1. API Log Analysis:

On 15.10.2024 at 01:30:15, an unauthenticated source from IP address 192.168.1.100 sent consecutive GET requests to the /api/v1/portfolio/1000–1004 range using Python-based automation. All requests received a 401 Unauthorized response. Although this behavior is suspicious, it does not constitute conclusive evidence on its own. According to the security_test_schedule.pdf document, this IP is assigned for the Internal Security Scanner (Python-based) under “Test 1: Automated Vulnerability Scanning.” Therefore, the activity was conducted as part of a planned penetration test. At 01:45:10, the IP address 10.0.0.50, using the user sec_team, sent GET requests to the /api/v1/portfolio/5001–5005 range. The requests used User-Agent: acme-mobile-ios/3.2.1 and session token: test_token_xyz_5001–5005, and all responses were 200 OK. According to the document, account IDs 5001–5010 and the 10.0.0.0/24 range are reserved for test accounts. This activity was verified as a planned scan by the Security Team. However, an anomaly was detected for user ID 1523. At 06:45:10, a user successfully logged in from IP address 203.0.113.45 and, shortly afterward, sent GET requests to different account_ids using jwt_token_1523_stolen. The API responded with 200 OK for these requests. This indicates JWT token theft and an access control vulnerability.

In conclusion:

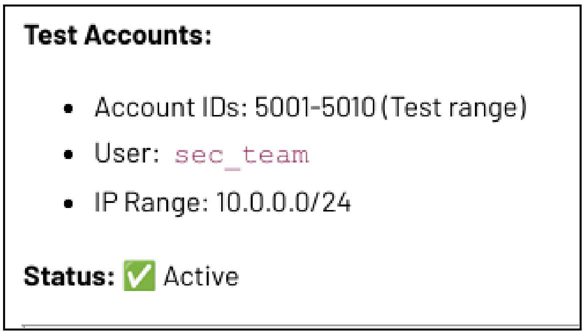
- 192.168.1.100 and 10.0.0.50 → Planned tests (documented)
- 203.0.113.45 → Real security breach (unauthorized access, token theft)



Scheduled Tests

Test 1: Automated Vulnerability Scanning

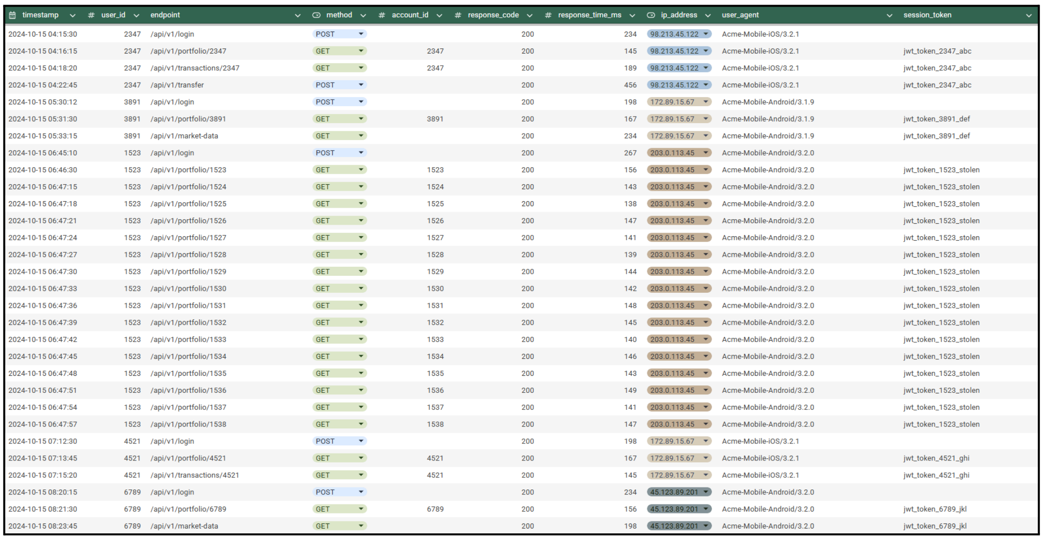
Type: Weekly Automated Scan
Schedule: Every Tuesday, 01:30 AM PST
Target: All production systems
Tool: Internal Security Scanner (Python-based)
Source IP: 192.168.1.100 (Internal Network)



Test Accounts:

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

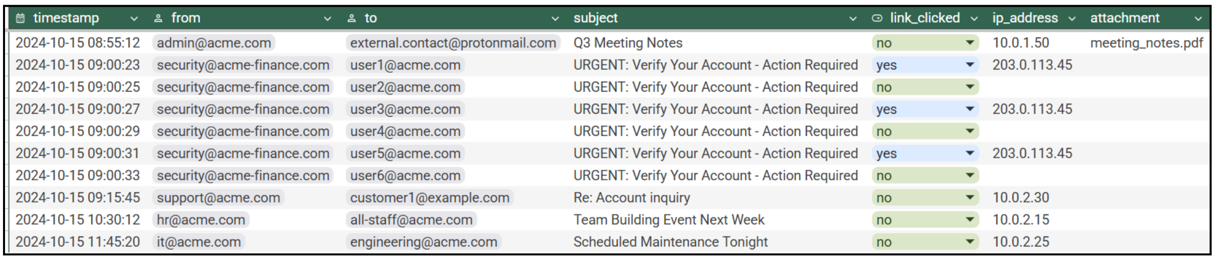
Status: ✅ Active



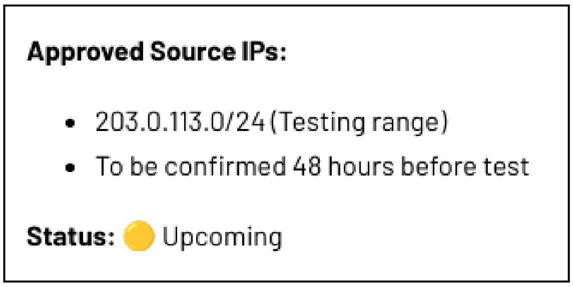
timestamp	user_id	endpoint	method	account_id	response_code	response_time_ms	ip_address	user_agent	session_token
2024-10-15 04:15:30	2347	/api/v1/login	POST		200	234	98.213.45.122	Acme-Mobile-iOS/3.2.1	
2024-10-15 04:16:15	2347	/api/v1/portfolio/2347	GET	2347	200	145	98.213.45.122	Acme-Mobile-iOS/3.2.1	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
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
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

3.2. Email Log Analysis:

Analysis of email log files revealed that on 15.10.2024, between 09:00:23 and 09:00:33, emails with the subject “URGENT: Verify Your Account - Action Required” were sent from security@acme-finance.com to internal users of the organization. The email contained a clickable link, had no attachments, and it was observed that the users user1@acme.com, user3@acme.com, and user5@acme.com clicked on the link. The email was sent from IP address 203.0.113.45, which is listed in the security_test_schedule.pdf document as an approved source IP within the 203.0.113.0/24 testing range for “Test 2: Quarterly Penetration Test” under scheduled penetration testing and security assessments. This indicates that the emails sent to users were part of a phishing attack designed to create fear and panic to prompt users to click the link. To prevent phishing attacks, users should be regularly trained on security awareness. Multi-factor authentication (MFA) and up-to-date security software should be implemented. Email filtering, URL verification, and protocols such as SPF, DKIM, and DMARC help block fraudulent emails. Suspicious links should be reported and addressed promptly, and regular backups should be maintained to minimize potential damage.



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	



Approved Source IPs:

- 203.0.113.0/24 (Testing range)
- To be confirmed 48 hours before test

Status: 🟡 Upcoming

3.3. Web Log Analysis:

Analysis of web log files revealed that on 15.10.2024, between 09:18:30 and 09:30:00, an account with user ID 1523 interacting from IP address 203.0.113.45 was observed. At 09:18:30, a successful login (HTTP 200) occurred, and at 09:19:15 the user accessed the dashboard. From the search field on the dashboard, the following payloads were attempted in sequence:

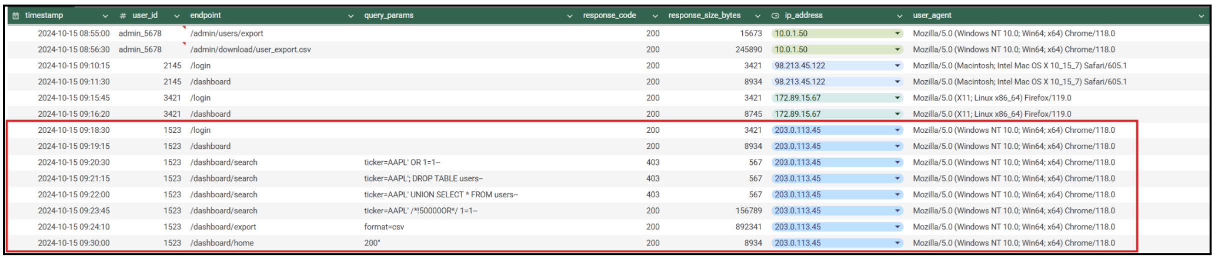
- ticker=AAPL' OR 1=1-- (intended to return all records/bypass session),
- ticker=AAPL'. DROP TABLE users-- (intended to delete a database table),
- ticker=AAPL' UNION SELECT * FROM users-- (intended to extract user data).

These attempts were blocked by the WAF with 403 Forbidden responses. Subsequently, a variant with obfuscation, ticker=AAPL' /*!50000OR*/ 1=1--, bypassed the WAF, and the request to /dashboard/export?format=csv returned a 200 OK response with 892,341 bytes of data. Finally, the /dashboard/home request returned 200 OK with a logged response size of 8,934 bytes.

To prevent SQL injection attacks:

- Use parameterized queries / prepared statements or an ORM for database queries.

- Sanitize all user inputs using whitelist-based validation and length/character checks.
- Restrict database access according to the principle of least privilege and remove critical permissions (DROP/ALTER) from application users.
- Block known attack patterns with a WAF.
- Keep application and database software up to date; conduct regular DAST/SAST scans and penetration tests.
- Maintain logging for suspicious activities, correlate with SIEM, implement rapid response procedures, and enforce regular backup policies.



timestamp	user_id	endpoint	query_params	response_code	response_size_bytes	ip_address	user_agent
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
2024-10-15 09:10:15	2145	/login		200	3421	98.213.45.122	Mozilla/5.0 (Macintosh; Intel Mac OS X 10_15_7) Safari/605.1
2024-10-15 09:11:30	2145	/dashboard		200	8934	98.213.45.122	Mozilla/5.0 (Macintosh; Intel Mac OS X 10_15_7) Safari/605.1
2024-10-15 09:15:45	3421	/login		200	3421	172.89.15.67	Mozilla/5.0 (X11; Linux x86_64) Firefox/119.0
2024-10-15 09:16:20	3421	/dashboard		200	8745	172.89.15.67	Mozilla/5.0 (X11; Linux x86_64) Firefox/119.0
2024-10-15 09:18:30	1523	/login		200	3421	203.0.113.45	Mozilla/5.0 (Windows NT 10.0; Win64; x64) Chrome/118.0
2024-10-15 09:19:15	1523	/dashboard		200	8934	203.0.113.45	Mozilla/5.0 (Windows NT 10.0; Win64; x64) Chrome/118.0
2024-10-15 09:20:30	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
2024-10-15 09:21:15	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
2024-10-15 09:22:00	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
2024-10-15 09:23:45	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
2024-10-15 09:24:10	1523	/dashboard/export	format=csv	200	892341	203.0.113.45	Mozilla/5.0 (Windows NT 10.0; Win64; x64) Chrome/118.0
2024-10-15 09:30:00	1523	/dashboard/home	200"	200	8934	203.0.113.45	Mozilla/5.0 (Windows NT 10.0; Win64; x64) Chrome/118.0

3.4. Waf Log Analysis:

On 15.10.2024, a series of suspicious activities were detected from IP address 203.0.113.45. At 09:20:30, 09:21:15, and 09:22:00, attempts were made via the /dashboard/search endpoint to query data or compromise the database using SQL commands. For example:

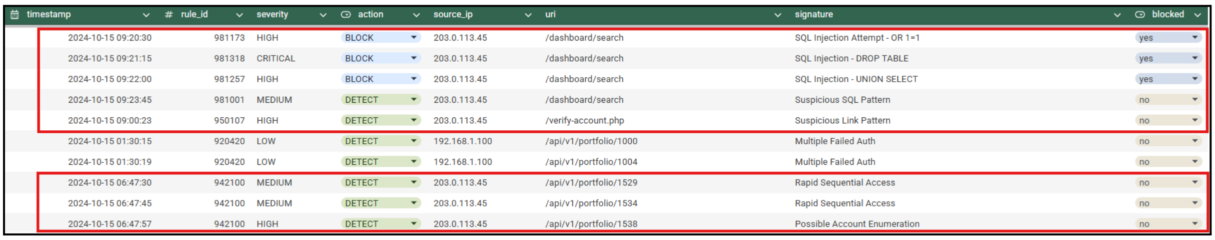
- ticker=AAPL' OR 1=1-- aimed to return all records,
- ticker=AAPL'. DROP TABLE users-- aimed to delete the users table,

- ticker=AAPL' UNION SELECT * FROM users-- aimed to extract user information.

These three attempts were blocked by the WAF and did not succeed. However, the attacker later used a variant, ticker=AAPL' /*!50000OR*/ 1=1--, to bypass the WAF. The /*!50000OR*/ expression exploits MySQL's comment parsing to evade standard WAF rules. Following this bypass, requests to /dashboard/export?format=csv and /dashboard/home were successful and data was retrieved.

Earlier in the day, around 06:47, the same IP was observed performing rapid consecutive accesses to portfolio endpoints and other activities resembling account scanning.

In summary, this IP exhibited both SQL injection attempts and information-gathering behaviors, successfully bypassed the WAF in one instance, and obtained some data. Therefore, monitoring this IP, strengthening WAF rules, and conducting a detailed investigation of suspicious activities are recommended.



timestamp	rule_id	severity	action	source_ip	uri	signature	blocked
2024-10-15 09:20:30	981173	HIGH	BLOCK	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
2024-10-15 09:23:45	981001	MEDIUM	DETECT	203.0.113.45	/dashboard/search	Suspicious SQL Pattern	no
2024-10-15 09:00:23	950107	HIGH	DETECT	203.0.113.45	/verify-account.php	Suspicious Link Pattern	no
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
2024-10-15 06:47:30	942100	MEDIUM	DETECT	203.0.113.45	/api/v1/portfolio/1529	Rapid Sequential Access	no
2024-10-15 06:47:45	942100	MEDIUM	DETECT	203.0.113.45	/api/v1/portfolio/1534	Rapid Sequential Access	no
2024-10-15 06:47:57	942100	HIGH	DETECT	203.0.113.45	/api/v1/portfolio/1538	Possible Account Enumeration	no

4. Acme Financial Services Current Architecture review and improvement

4.1. Current Architecture – Critical Vulnerabilities Identified

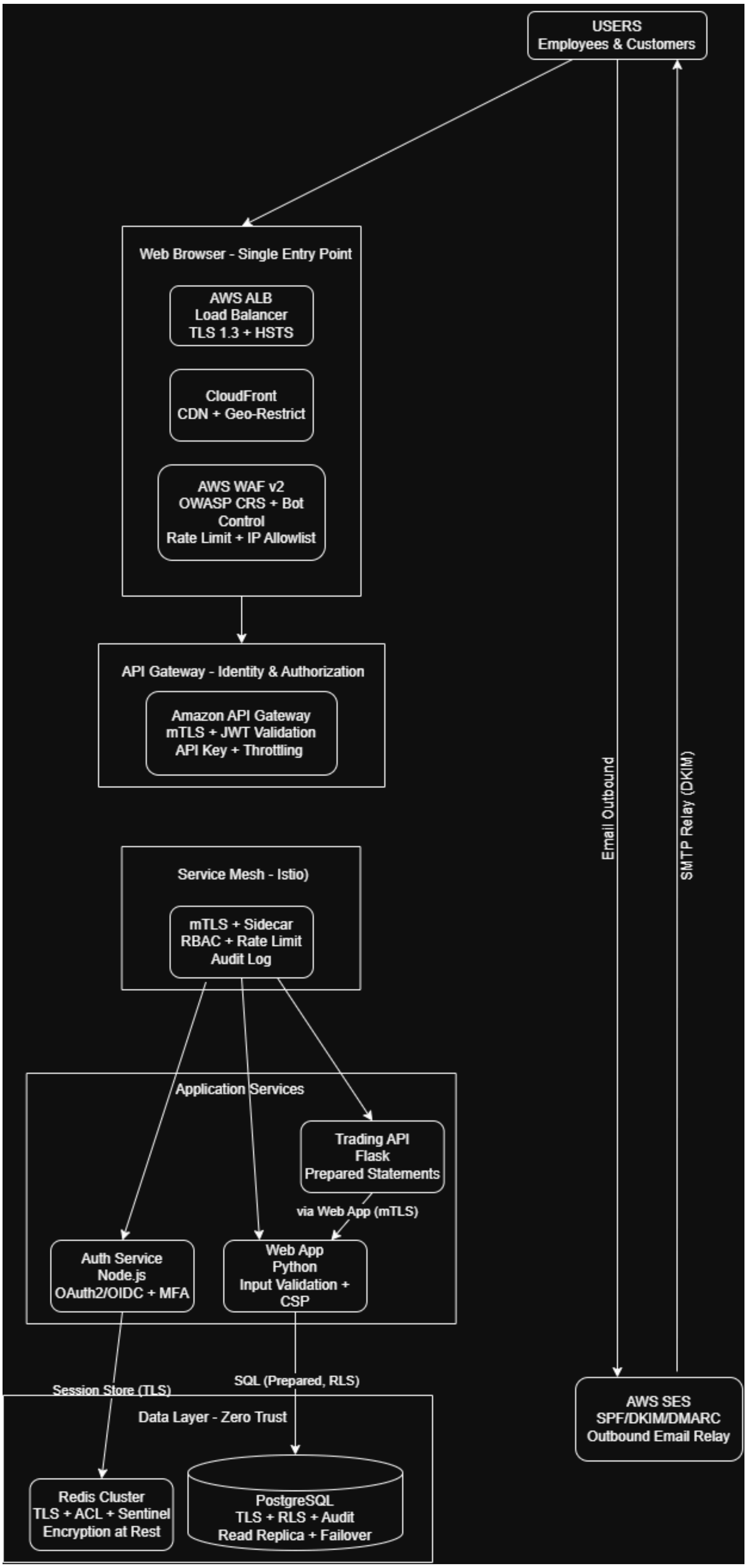
Component	**Risk**	**Effect**
Email Gateway (Port 25/587)	Open SMTP, spam/phishing, open relay	Critical
TLS only on API Gateway	Internal traffic is unencrypted (WAF → App → DB)	Critical
WAF – Basic Rules	OWASP Top 10, bot, no rate limit	High
Trading API → PostgreSQL (SQL)	Direct DB access, SQLi risk	Critical
Load Balancer only web	Bypasses email traffic	High
Redis (Sessions)	Password-free connection, no ACL	Medium
Auth Service → Web App	Authentication can be bypassed	High

4.2. New Architecture – Zero Trust & Security Focused

Improvement	**Description**	**ISO 27001**
All traffic → AWS ALB	Single entry point	A.13.1.1 (Network controls)
Email Gateway → AWS SES	Open SMTP removed	A.12.4.1 (Event logging) A.13.1.3 (Segregation in networks)
mTLS between all services	Encryption + authentication	A.8.24 (Use of cryptography) A.12.2.1 (Controls against malware)
WAF → OWASP CRS + Rate Limit + Bot Control	Attack blocking	A.12.4.1 (Logging) A.14.2.7(Secure development)

Trading API → Redirect to Web App, not DB	DB access restricted	A.9.1.2(Access to systems)
		A.9.4.2(Secure log-on)

API Gateway → JWT + mTLS + Throttling	ID + speed limit	A.9.2.4 (Management of secret authentication)
Istio Service Mesh	Policy, audit, mTLS	A.12.4.1 (Event logging) A.12.4.3 (Administrator logs)
PostgreSQL → TLS + RLS + Audit	secure access	A.8.24 (Cryptography) A.12.4.2 (Protection of log info)
Redis → TLS + ACL + Sentinel	Secure cache	A.8.24 (Cryptography)


USERS
Employees & Customers
Web Browser - Single Entry Point
AWS ALB
Load Balancer
TLS 1.3 + HSTS
CloudFront
CDN + Geo-Restrict
AWS WAF v2
OWASP CRS + Bot Control
Rate Limit + IP Allowlist
API Gateway - Identity & Authorization
Amazon API Gateway
mTLS + JWT Validation
API Key + Throttling
Service Mesh - Istio)
mTLS + Sidecar
RBAC + Rate Limit
Audit Log
Application Services
Trading API
Flask
Prepared Statements
via Web App (mTLS)
Auth Service
Node.js
OAuth2/OIDC + MFA
Web App
Python
Input Validation + CSP
Session Store (TLS)
SQL (Prepared, RLS)
Data Layer - Zero Trust
Redis Cluster
TLS + ACL + Sentinel
Encryption at Rest
PostgreSQL
TLS + RLS + Audit
Read Replica + Failover
Email Outbound
SMTP Relay (DKIM)
AWS SES
SPF/DKIM/DMARC
Outbound Email Relay

5. Intervention and Solution Processes

5.1.Emergency Response (0–24 hours)

- Immediately isolate the IP address 203.0.113.45 and associated accounts (especially user ID 1523).
- Cancel and regenerate all active JWT tokens.
- Determine the extent of the data leak (what CSV data was leaked).
- Implement temporary access restriction for API and web application (e.g. maintenance mode).
- Stop phishing email traffic; block domains of fake emails.
- Start IP and domain based IOC monitoring via SOC/SIEM.

5.2. Short-Term Solutions (1–2 weeks)

- Strengthen access control mechanism: Add JWT authentication and scope-based authorization.
- Update WAF configuration: add OWASP CRS + SQLi bypass detection + rate limit.
- Enable SPF, DKIM, DMARC records in email infrastructure.
- Require MFA for all users.
- Conduct phishing awareness training (including users such as user1, user3, user5).
- Update SIEM correlation rules based on attack patterns.

5.3. Long-Term Improvements (1–3 months)

- Implement the “Zero Trust Architecture” transition plan (as outlined in section 4.2 of the report):
  - Route all traffic through AWS ALB.
  - Enforce inter-service encryption with mTLS.
  - Enable Service Mesh (Istio) policies and audit logging.
  - Remove direct DB access via Trading API → Web App redirect.
  - Secure PostgreSQL, Redis and Auth services with TLS + ACL + audit.
- Schedule regular DAST/SAST + pentesting sessions.
- Make backup and incident response plans ISO 27001 compliant.

5.4. Compatibility Issues

- ISO 27001 control mappings:
  - A.13.1.1 / A.13.1.3: Network traffic segregation (Zero Trust – ALB).
  - A.8.24 / A.12.2.1: mTLS and cryptographic protection.
  - A.12.4.1–3: Logging, event monitoring, protection of administrator logs.
  - A.9.1.2 / A.9.4.2 / A.9.2.4: Authorization, login security, secret credential management
- Data leak notifications must be made under the KVKK/GDPR.
- A digital forensic investigation should be initiated for potential internal threats or pentest breaches.