| | |
|---|---|
| Severity: | **High** |
| Impact: | **Unauthorized Access and Data Exposure** |
| Access Point: | **Web API (Port 443)** |
| CVSS Skoru: | |
| MITRE ATT&CK ID: | **T1078 – Valid Accounts** |
| Tools Used: | **Log Correlation, Threat Intelligence Analysis** |
| User Profile: | **Unauthorized** |
| Affected Components | • `/api/v1/portföy` endpoint on multiple user IDs<br>• IP: `203.0.113.45`<br>• Related user accounts: `1523`, `1524` … `1538` |
| Vulnerability Name | **Stolen Token and Mass Enumeration via Insecure API Authorization** |

Description:

Through analysis of API access logs, multiple unauthorized requests were detected originating from the IP address **203.0.113.45**.
The attacker used a **valid JWT access token** (`jwt_token_1523_stolen`) belonging to user **ID 1523** to sequentially query other users' portfolio endpoints (`/api/v1/portföy/1524` to `/1538`).

All requests received **HTTP 200 OK**, confirming successful access to resources that should have been restricted. This indicates a **Broken Object Level Authorization (BOLA / IDOR)** vulnerability, where the API does not validate whether the authenticated token owner has permission to access a specific resource.

This behavior demonstrates that once a single token is compromised, an attacker can freely enumerate other users' data without server-side ownership validation. The incident also shows possible credential leakage or phishing compromise earlier in the attack chain.

- **Data Exposure:** Attackers can retrieve sensitive financial portfolio data of multiple users.
- **Session Hijacking:** Reuse of stolen JWTs allows persistent access without re-authentication.
- **Regulatory Risk:** Violates data privacy principles (GDPR / KVKK).
- **Reputation Damage:** Compromise of user trust and potential financial liability.

## Root Cause:

- Missing ownership validation on backend endpoints.
- Long-lived JWT tokens without device/IP binding.
- Lack of token revocation upon compromise.
- No anomaly detection for rapid sequential requests.

## Technical Evidence (Log Summary):

| Timestamp | User ID | Endpoint | Response | IP Address | Token |
|---|---|---|---|---|---|
| 2024-10-15 | 1523 | `/api/v1/portföy/1523` | 200 | 203.0.113.45 | jwt_token_1523_stolen |

| | | | | |
|---|---|---|---|---|
| 06:46:30 | | | | |
| 2024-10-15<br>06:47:15<br>06:47:57 | 1523 | /api/v1/portföy/1524-1538 | 200 | 203.0.113.45 jwt_token_1523_stolen |

Recommendations:

- **Implement Object-Level Authorization Controls**

Validate that the `account_id` or `user_id` in the request matches the authenticated user associated with the JWT.

```
if request.user.id != resource.owner_id:
    return 403
```

- **Shorten Token Lifespan and Use Refresh Tokens**

Enforce token expiry (e.g., 10–15 minutes) and implement refresh token rotation.

- **Enable Token Revocation and Device Binding**

Revoke compromised tokens immediately; associate tokens with device fingerprint/IP to prevent reuse elsewhere.

- **Introduce Rate Limiting and Behavior Analytics**

Detect and block rapid sequential requests (e.g., >10 requests within 1 second) using the same token.

- **Enhance Monitoring and SIEM Rules**

Create alerts for:

  - Same token accessing multiple user IDs.
  - High-frequency API requests from single IPs.
  - Tokens reused from new geolocations.

- **Security Awareness & MFA Enforcement**

Educate users on phishing attacks, enforce Multi-Factor Authentication for login and sensitive operations.

| | |
|---|---|
| | **EMAİL01** |
| Severity: | **Critical** |
| Impact: | **Credential Theft, Account Compromise & Potential Data Exfiltration** |
| Access Point: | **Corporate Email Gateway (SMTP / Port 25, 465, 587)** |
| CVSS Skoru: | |
| MITRE ATT&CK ID: | **T1566 – Phishing** |
| Tools Used: | **Email Log Analysis, Threat Intelligence, Header Inspection** |
| User Profile: | **Non-privileged internal users** |
| Affected Components | • **Email system (acme.com domain)**<br>• **Multiple internal mailboxes (user1@acme.com – user6@acme.com)** |

| | |
|---|---|
| | **• IP address 203.0.113.45 (external malicious sender)**<br>**• Outbound email from admin@acme.com → external.contact@protonmail.com** |
| Vulnerability Name | **Phishing Campaign and Uncontrolled Data Exfiltration via Corporate Email** |
| Description: | **Analysis of the corporate email gateway logs revealed a coordinated phishing campaign and a potential data exfiltration event.**<br>**At 09:00 on 2024-10-15, multiple internal users received an email from the spoofed address security@acme-finance.com, with the subject “URGENT: Verify Your Account – Action Required.”**<br>**These messages originated from the IP 203.0.113.45, a known malicious source also associated with web attacks in other findings.**<br>**Among the targeted recipients, three users clicked the embedded phishing link, as indicated by the "yes" values in the link_clicked field. These interactions likely exposed their login credentials, enabling subsequent unauthorized access and token theft observed in the API activity logs.**<br>**Additionally, at 08:55, a separate outbound message from admin@acme.com to an external ProtonMail address (external.contact@protonmail.com) contained an attachment named meeting_notes.pdf, raising the possibility of sensitive data leakage if this transfer was not business-approved.**<br>**Technical Evidence (Email Log Summary):**<br>**Timestamp From To Subject Clicked IP Attachment**<br>**2024-10-15 09:00:23 security@acme-finance.com user1@acme.com URGENT: Verify Your Account Yes 203.0.113.45 –**<br>**2024-10-15 09:00:27 security@acme-finance.com user3@acme.com URGENT: Verify Your Account Yes 203.0.113.45 –**<br>**2024-10-15 09:00:31 security@acme-finance.com user5@acme.com URGENT: Verify Your Account Yes 203.0.113.45 –**<br>**2024-10-15 08:55:12 admin@acme.com external.contact@protonmail.com Q3 Meeting Notes No 10.0.1.50 meeting_notes.pdf**<br>**• Credential Compromise: Users who clicked the phishing link likely submitted login credentials to a fake portal.**<br>**• Lateral Movement: Compromised accounts can be used for internal reconnaissance or further phishing.**<br>**• Sensitive Data Leakage: Uncontrolled outbound transfer of meeting documents may violate data handling policies.**<br>**• Reputation & Compliance Risk: Potential breach of privacy laws** |

| | |
|---|---|
| | **(GDPR/KVKK) and corporate information-security policy.**<br>**Root Cause:**<br>**• Lack of anti-phishing protection (SPF/DKIM/DMARC misconfigured).**<br>**• Insufficient email content filtering and link sandboxing.**<br>**• Lack of user awareness training on phishing identification.**<br>**• No DLP (Data Loss Prevention) control for outbound attachments.** |
| Recommendations: | **• Strengthen Email Authentication:**<br>**• Implement and enforce SPF, DKIM, and DMARC with p=reject policy.**<br>**• Monitor DMARC reports for spoofing attempts.**<br>**• Deploy Advanced Threat Protection (ATP):**<br>**• Use sandboxing for attachments and embedded links.**<br>**• Quarantine emails from unverified senders or with “urgent verification” phrases.**<br>**• Enable Data Loss Prevention (DLP):**<br>**• Detect and block sensitive attachments sent to external domains such as protonmail.com.**<br>**• Require managerial approval or encryption for outbound sensitive files.**<br>**• Conduct Security Awareness Training:**<br>**• Simulate phishing campaigns quarterly.**<br>**• Train users to identify suspicious sender domains and report via “Report Phish” button.**<br>**• Implement MFA and Session Review:**<br>**• Enforce MFA for all user accounts to mitigate credential reuse.**<br>**• Review sessions from IP 203.0.113.45 and revoke compromised tokens immediately.**<br>**• Integrate SIEM Correlation:**<br>**• Link phishing click events to subsequent API or login activity for rapid incident response.** |

| | |
|---|---|
| | **WAF-01** |

| | |
|---|---|
| Severity: | **High (with Critical sub-findings)** |
| Impact: | **SQL Injection Attempts, Account Enumeration, and Credential/Account Abuse** |
| Access Point: | **Web Application Firewall / IDS (HTTP endpoints: /dashboard/search, /verify-account.php, /api/v1/portfolio/*, /admin/users/export)** |
| CVSS Skoru: | |
| MITRE ATT&CK ID: | **T1190 (Exploit Public-Facing Application), T1078 (Valid Accounts) — plus T1588/T1589 for credential phishing** |
| Tools Used: | **WAF/IDS (rule signatures), log correlation, SIEM** |
| User Profile: | **External attacker (203.0.113.45) and internal suspicious client (192.168.1.100)** |
| Affected Components | **• Web application search endpoint: /dashboard/search**<br>**• Account verification endpoint: /verify-account.php**<br>**• REST API resource endpoints: /api/v1/portfolio/{id}**<br>**• Admin export endpoint: /admin/users/export**<br>**• Source IPs involved: 203.0.113.45, 192.168.1.100, and normal login IP 45.123.89.201** |
| Vulnerability Name | **Multiple Web Attacks Detected: SQL Injection Attempts, Account Enumeration & Rapid Sequential Access** |
| Description: | **The WAF/IDS logs indicate a concentrated attack campaign from 203.0.113.45 combining multiple techniques:**<br>**• SQL Injection attempts targeting /dashboard/search with payloads such as OR 1=1, DROP TABLE, and UNION SELECT. Several attempts were blocked (critical/high rules fired).**<br>**• Suspicious link pattern seen against /verify-account.php, consistent with phishing payloads or malicious redirect probes.**<br>**• Rapid sequential access and possible account enumeration on /api/v1/portfolio/{id} using sequential resource IDs—this behavior indicates token misuse or IDOR/authorization weaknesses.**<br>**• Multiple failed authentication attempts seen from 192.168.1.100 (credential stuffing/brute force).**<br>**• An admin export access was detected from internal IP 10.0.1.50 (requires validation whether authorized).**<br>**• Normal login patterns were also observed (45.123.89.201) and can be used as a baseline for anomaly detection.**<br>**Collectively these signals indicate an attacker performing multi-vector exploitation: reconnaissance → injection probes → credential harvesting / token reuse → enumeration.**<br>**Technical Evidence (selected log rows)**<br>**Timestamp Rule ID Severity Action Source IP URI Signature** |

| | |
|---|---|
| | **Blocked**<br><br>**2024-10-15 09:20:30 981173 HIGH DETECT 203.0.113.45 /dashboard/search SQL Injection Attempt - OR 1=1 yes**<br><br>**2024-10-15 09:21:15 981318 CRITICAL BLOCK 203.0.113.45 /dashboard/search SQL Injection - DROP TABLE yes**<br><br>**2024-10-15 09:22:00 981257 HIGH BLOCK 203.0.113.45 /dashboard/search SQL Injection - UNION SELECT yes**<br><br><br><br><br>**• Database compromise or data leak if any successful SQL injection exploited data retrieval or destructive queries.**<br><br>**• Mass data disclosure via account enumeration or BOLA/IDOR.**<br><br>**• Credentials compromise from brute-force or phishing correlation.**<br><br>**• Potential business disruption (if injection caused DB corruption) and regulatory exposures from leaked PII/financial data.**<br><br>**Root Causes (likely)**<br><br>**• Insufficient input validation and improper parameterization in /dashboard/search and other endpoints.**<br><br>**• Lack of strict object-level authorization on resource endpoints (/api/v1/portfolio/{id}).**<br><br>**• Weak or absent rate-limiting and bot detection allowing rapid sequential access.**<br><br>**• Incomplete WAF tuning (some suspicious events detected but not blocked).**<br><br>**• Possible lack of DLP or monitoring for admin export events.** |
| Recommendations: | **Immediate Remediation (0–4 hours)**<br><br>**1. Block IP 203.0.113.45 at perimeter (WAF/Firewall) and add to threat-intel blocklist.**<br><br>**2. Ensure WAF blocks are enabled for the critical SQLi signatures (IDs 981318, 981257) and tune to block rather than detect for high-confidence signatures.**<br><br>**3. Quarantine suspicious inbound traffic and capture full request bodies for forensic analysis.**<br><br>**4. Rotate DB backups and verify integrity; take DB snapshot for forensics.**<br><br>**5. Rate-limit requests to /api/v1/portfolio/* and /dashboard/search (e.g., 10 requests/min** |

**per IP) and apply CAPTCHA/Challenge on excessive requests.**

**6. Force password rotation / MFA for accounts exhibiting failed auth or related suspicious events (and for accounts that clicked phishing links in correlated logs).**

**7. Review admin export audit — confirm if 10.0.1.50 export is authorized; if not, revoke and investigate.**

Short-to-Mid Term Remediations (1–14 days)

- **Code fixes: Convert all dynamic SQL to parameterized queries / prepared statements; validate/whitelist search inputs; limit characters and length for query parameters.**

- **Enforce object-level authorization: Verify resource ownership server-side before returning data.**

- **WAF tuning & testing: Create test cases to validate WAF blocks legitimate attack payloads, reduce false positives, and add custom rules for rapid sequential access detection.**

- **Implement strict rate-limiting & bot mitigation: Per-IP and per-token limits, progressive throttling, and challenge-response for suspicious clients.**

- **SIEM correlation rules: Alert on patterns: same token accessing multiple resource IDs; same IP performing SQLi and phishing; rapid sequential IDs.**

- **Harden auth flows: Shorten token lifetimes, enable refresh rotation, token binding (device/IP fingerprinting) and introduce anomaly-based session invalidation.**

- **DLP for admin exports: Prevent or flag exports that contain sensitive fields or are sent to external domains.**

Long Term / Strategic Controls (2+ weeks)

- **Automated attack simulation / pentesting: Include SQLi, IDOR, and enumeration detection tests in CI/CD.**

- **AppSec practices: SAST/DAST integrated into build pipeline; dependency scanning; secure coding training.**

- **UEBA (User/Entity Behavior Analytics): Detect lateral movements and anomalous token reuse across systems.**

- **Threat intel & feed integration: Block known-malicious IP ranges globally and feed back IOC’s to mail/web gateways.**

Suggested Detection Rules (SIEM) — quick examples

- **Token Reuse Across Resources: Alert if same authorization token accesses >3 distinct account_id values within 2 minutes.**

- **Sequential Enumeration Pattern: Alert if one IP requests 8+ portfolio IDs with ascending IDs within 1 minute.**

- **SQLi Signature Correlation: If WAF detects SQLi signature AND same source IP has**

| | |
|---|---|
| | **phishing mail events, escalate to high priority.** |

<table>
<tr><td></td><td><b>Web</b></td></tr>
<tr><td>Severity:</td><td><b>Critical</b></td></tr>
<tr><td>Impact:</td><td><b>Successful SQL Injection &amp; Data Exfiltration through Export Functionality</b></td></tr>
<tr><td>Access Point:</td><td><b>/dashboard/search and /dashboard/export endpoints</b></td></tr>
<tr><td>CVSS Skoru:</td><td><b>9.1</b></td></tr>
<tr><td>MITRE ATT&amp;CK ID:</td><td><b>T1190 – Exploit Public-Facing Application</b></td></tr>
<tr><td>Tools Used:</td><td><b>Web Application Logs, Request Parameter Inspection, User-Agent Correlation</b></td></tr>
<tr><td>User Profile:</td><td><b>External Attacker (203.0.113.45, masquerading as normal Chrome client)</b></td></tr>
<tr><td>Affected Components</td><td>• Web Application (Dashboard module)<br>◦ /dashboard/search<br>◦ /dashboard/export<br>• Database connected to search functionality (likely users table)<br>• Admin user export endpoints (/admin/users/export, /admin/download/user_export.csv)<br>• Source IP: <b>203.0.113.45</b><br>• User ID involved: <b>1523</b></td></tr>
<tr><td>Vulnerability Name</td><td><b>SQL Injection Leading to Unauthorized Data Extraction</b></td></tr>
<tr><td>Description:</td><td>The application logs reveal a <b>classic and successful SQL Injection attack sequence</b> originating from IP <b>203.0.113.45</b> using user ID 1523.<br>The attacker began with benign activity (/login, /dashboard), then escalated to <b>injection attempts</b> against /dashboard/search by manipulating the ticker parameter.<br>Progression observed:<br>1. <b>Initial probes</b> with OR 1=1 and DROP TABLE payloads returned HTTP 403 (blocked by WAF).<br>2. <b>Bypass success:</b> Payload /*!50000OR*/ 1=1-- returned HTTP 200 and a significantly <b>larger response size (156,789 bytes)</b> — strong indicator of <b>successful SQLi</b>.<br>3. Immediately after, attacker accessed /dashboard/export?format=csv, returning a <b>massive file (892,341 bytes)</b> — likely <b>data exfiltration</b> of the queried table contents.<br>This sequence confirms exploitation and data extraction of sensitive database records.<br><b>Technical Evidence (Log Excerpts)</b>
<table>
<tr><th>Timestamp</th><th>User ID</th><th>Endpoint</th><th>Query Params</th><th>Response</th><th>Size (bytes)</th><th>IP</th><th>Notes</th></tr>
</table>
</td></tr>
</table>

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| 2024-10-15 09:20:30 | 1523 | /dashboard/search | `ticker=AAPL' OR 1=1--` | 403 | 567 | 203.0.113.45 | Blocked probe |
| 2024-10-15 09:21:15 | 1523 | /dashboard/search | `ticker=AAPL'; DROP TABLE users--` | 403 | 567 | 203.0.113.45 | Attempted destructive injection |
| 2024-10-15 09:22:00 | 1523 | /dashboard/search | `ticker=AAPL' UNION SELECT * FROM users--` | 403 | 567 | 203.0.113.45 | Data extraction attempt |
| 2024-10-15 09:23:45 | 1523 | /dashboard/search | `ticker=AAPL' /*!50000OR*/ 1=1--` | **200** | **156,789** | 203.0.113.45 | **Successful injection (bypass)** |
| 2024-10-15 09:24:10 | 1523 | /dashboard/export | `format=csv` | **200** | **892,341** | 203.0.113.45 | **Possible exfiltration** |

The spike in **response size** and **change in HTTP status code** from `403 → 200` precisely matches a successful SQLi bypass and data retrieval event.

- **Full compromise of database integrity and confidentiality.**
- Attack likely exposed internal `users` table data (names, credentials, emails, etc.).
- Exported CSV file (892 KB) suggests **bulk data leakage**.
- Potential **breach of customer information** and **GDPR/KVKK compliance violation**.
- Risk of **follow-up attacks** (credential stuffing, spear phishing).

## Root Cause

- The `/dashboard/search` endpoint **directly concatenates user input** (e.g., `ticker` parameter) into SQL queries.
- **No input sanitization** or **parameterized queries** used.
- Lack of **output encoding** or **prepared statements**.
- Inadequate WAF coverage — 403 blocks bypassed by comment-based obfuscation (`/*!50000OR*/`).
- `/dashboard/export` lacks **authorization control** or **query sanitization** before exporting results.

Recommendations:

## Immediate Remediation (0–4 hours)

1. **Take `/dashboard/search` and `/dashboard/export` offline** until validated safe.
2. **Rotate database credentials** and revoke any exposed API keys/tokens.
3. **Conduct full database audit** to confirm data integrity and leakage scope.
4. **Block IP `203.0.113.45`** at all firewalls and proxies.
5. **Purge sensitive exports** from file systems and logs.

6. **Alert DPO and Incident Response teams** — likely data breach under legal reporting requirements.

## Short-Term Fixes (1–7 days)

- Rewrite affected SQL queries to use **parameterized queries** (e.g., prepared statements).
- Implement **server-side input validation** for `ticker` (allow only alphanumeric ticker symbols).
- Encode all output before rendering search results.
- Patch WAF to detect MySQL comment-based bypass patterns (`/*!...*/`).
- Restrict export endpoints to **admin roles only**, protected via **MFA and audit logs**.
- Add response-size anomaly alerts to SIEM — large output following SQL query should trigger alarm.

## Long-Term Recommendations (7+ days)

- Integrate **Static Application Security Testing (SAST)** and **Dynamic Application Security Testing (DAST)** into CI/CD.
- Deploy **Web Application Firewall tuning** with adaptive learning and SQLi-specific ML signatures.
- Apply **role-based access control (RBAC)** and **principle of least privilege** on admin/export modules.
- Conduct **developer secure coding training** on SQL injection prevention and secure data handling.
- Implement **database activity monitoring (DAM)** for ongoing detection of SQL anomalies.

- The same attacker IP (`203.0.113.45`) also appeared in:
  - **WAF-01** (SQL injection probes, account enumeration)
  - **EMAIL-01** (phishing campaign to steal credentials)
  - Indicates a **multi-stage attack campaign** involving credential theft → web exploitation → data exfiltration.

This event represents a **confirmed SQL Injection compromise** with **successful data extraction**.
The attacker bypassed basic WAF filters using encoded payloads and leveraged the export functionality for exfiltration.
Immediate remediation and forensic review are required.
The incident highlights the need for stronger **input validation**, **query parameterization**, and **segregation of duties** for sensitive data access.