EXECUTIVE SUMMARY

Target: http://testasp.vulnweb.com

Scan ID: AG-756F93A0

Scan Date: 2026-01-31 01:22:10

Findings: 1 vulnerabilities detected

Overall Status

VULNERABLE

- Detected 1 security issue(s) requiring attention.
- Immediate remediation recommended for critical findings.
- Review each finding below for detailed impact analysis.
- Prioritize fixes based on severity and exploitability.

DETAILED FINDINGS

Finding #1: Insecure Direct Object Reference (IDOR)

HIGH
CWE: CWE-639

CVSS Score: 8.6 (High)

Description:

- Application exposes internal object references without authorization checks.
- Attackers can access resources belonging to other users.
- Object IDs are predictable and not properly validated.

Impact:

- Unauthorized access to other users' data.
- Privacy breach affecting multiple users.
- Potential for mass data harvesting.
- Regulatory compliance violations (GDPR, etc.).

Forensic Analysis

Method: GET | Param: user_id
URL: /api/v1/user/1005
Analysis: The 'user_id' parameter is sequential and lacks authorization checks.

Table 1: Payload Details (Insecure Direct Object Reference)

```
RAW: 1005
ENC: 1005
```

Reproduction Command:

```
curl -X GET 'http://target/api/v1/user/1005' -H 'Authorization: Bearer <attacker
```

HTTP Traffic Snapshot:

Request:

```
GET /api/v1/user/1005 HTTP/1.1
Host: target
Authorization: Bearer <attacker_token>
```

Response:

```
HTTP/1.1 200 OK
Content-Type: application/json

{"id": 1005, "username": "victim_user", "email": "victim@corp.com", "role": "admin"}
```

Remediation:

- Implement proper authorization checks on all resource access.
- Use indirect references or UUIDs instead of sequential IDs.
- Validate user permissions before returning data.
- Log and monitor access patterns for anomalies.

Recommended Code Fix:

```
# VULNERABLE CODE:
@app.get("/user/{user_id}")
def get_user(user_id: int):
    return db.get_user(user_id)

# SECURE CODE:
@app.get("/user/{user_id}")
def get_user(user_id: int, current_user: User):
    if user_id != current_user.id and not current_user.is_admin:
        raise HTTPException(403, "Access denied")
    return db.get_user(user_id)
```

RECOMMENDATIONS

Remediation Strategy

- Remediation steps are provided within each specific finding above.
- Prioritize Critical and High severity issues.
- Re-scan after applying patches to verify fixes.

SCAN TIMELINE

- [Orchestrator] TARGET_ACQUIRED - 2026-01-31 01:18:51
- [Sigma] JOB_ASSIGNED - 2026-01-31 01:18:51
- [Beta] LOG - 2026-01-31 01:18:51
- [Gamma] VULN_CONFIRMED - 2026-01-31 01:18:51
- [Kappa] GI5_LOG - 2026-01-31 01:18:51