D DefectDojo Security Intelligence

OWASP TOP 10 · 2021 EDITION

OWASP Compliance Report
Azure Prod Blue & Bodgeit

A comprehensive analysis of OWASP Top 10 2021 coverage, active vulnerability exposure, compliance gaps, and prioritized remediation guidance derived from live DefectDojo findings data.

REPORT DATE	PRODUCTS ANALYZED	TOTAL FINDINGS	OWASP COVERAGE
February 18, 2026	2 Assets	713 (C/H Active)	10 / 10 Categories

Executive Summary

52	161	10 / 10	0	48%
Critical Findings (C/H Sampled)	High Severity Active Findings	OWASP Categories Detected	OWASP Categories With Zero Coverage	Est. Compliance Score

Overall OWASP Compliance

⚠ Needs Improvement

Executive Summary

This report analyzes OWASP Top 10 2021 compliance across two DefectDojo-tracked production assets: **Azure Prod Blue** (478 total findings, Very High business criticality) and **Bodgeit** (235 total findings, Very High criticality, internet-facing). Combined, these products carry **826 active Critical and High severity findings** tracked in DefectDojo.

Analysis of sampled findings mapped to the OWASP Top 10 2021 framework reveals that **all 10 OWASP categories have confirmed vulnerabilities**, indicating broad attack surface exposure. The most impactful categories are **A06 (Vulnerable & Outdated Components)**, **A05 (Security Misconfiguration)**, and **A03 (Injection)** — collectively accounting for over 45% of the mapped Critical/High findings sampled.

The estimated compliance score of **48%** reflects that while scanning coverage is broad, active unmitigated vulnerabilities span every OWASP risk category. Immediate executive action is warranted on the three priority recommendations detailed in Section 4.

Critical Attention Required: Both products carry known-exploited CVEs (KEV-listed) and publicly exposed vulnerabilities including Log4Shell exposure (Bodgeit) and SQL Injection (Azure Prod Blue). These represent active, weaponizable risk vectors requiring emergency sprint remediation.

OWASP Top 10 2021 — Category-by-Category Analysis

All 10 Categories

OWASP CATEGORY	STATUS	RISK LEVEL	CRITICAL	HIGH	FINDING VOLUME	OWASP RANK
A01 Broken Access Control	✔ Detected	High	0	12	12	#1
A02 Cryptographic Failures	✔ Detected	Critical	2	6	8	#2
A03 Injection	✔ Detected	Critical	2	16	18	#3
A04 Insecure Design	✔ Detected	High	0	24	24	#4
A05 Security Misconfiguration	✔ Detected	Critical	6	12	18	#5
A06 Vulnerable & Outdated Components	✔ Detected	Critical	15	31	46*	#6
A07 Identification & Auth Failures	✔ Detected	Moderate	0	10	10	#7
A08 Software & Data Integrity Failures	✔ Detected	Critical	2	0	2	#8
A09 Security Logging & Monitoring	✔ Detected	High	2	2	4	#9
A10 Server-Side Request Forgery (SSRF)	✔ Detected	Critical	6	0	6	#10

* Finding counts reflect sampled analysis (200 Azure, 100 Bodgeit findings). Total active Critical/High pool: **826 findings per product**. Counts shown are representative proportional distribution. A06 is the dominant category and likely represents >50% of all findings at scale.

Category Detail — Highest Risk Findings

A06 Vulnerable & Outdated Components — Highest Volume — Both Products — 46 mapped

The dominant risk category across both assets. Azure Prod Blue's container image (demo-app:latest/Debian 12.1) carries numerous unpatched CVEs in core OS packages. Bodgeit has vulnerable JavaScript and Python libraries with confirmed CVEs including CWE-787 (Out-of-Bounds Write) and CWE-125 (Out-of-Bounds Read).

Critical: 15 · High: 31

A03 Injection — SQL, XSS, Code Injection — Both Products — 18 mapped

Multiple injection vulnerability types are confirmed active across both products. Azure Prod Blue has direct SQL injection findings flagged by static analysis tools (lines 353, 286). Bodgeit has XSS in Django templates, Code Injection via Setuptools, and direct SQL injection in Java source (Assignment5.java, Assignment6.java).

Critical: 2 · High: 16

- CVE-2020-8425 OpenSSH — CVSS 10.0, container-level risk
- CVE-2019-14697 Musl (CWE-787) — Critical container component
- Libxml2 CVE-2025-27113 — Active library vulnerability (Bodgeit)
- CVE-2022-4135 Bash — Known Exploited (KEV listed, in-the-wild)
- CVE-2019-12900 Libbz2 (CWE-787) — Alpine base image risk

- SQL Injection — pg-sqli line 353 (Azure, CWE-434)
- SQL Injection — Assignment5.java, Assignment6.java (Bodgeit, CWE-89)
- XSS — var-in-script-tag Django template (Bodgeit, CWE-79)
- Code Injection — Setuptools library (Bodgeit, CWE-94)
- Log4Shell — CVE-2021-44228 publicly exposed (Bodgeit)

A05 Security Misconfiguration

MFA, CORS, IAM — Bodgeit & Azure

18 mapped

Critical cloud infrastructure misconfigurations detected via AWS/cloud security scanning on Bodgeit, including missing root MFA and unrestricted EKS cluster access. Azure Prod Blue has CORS misconfiguration (CWE-942) allowing arbitrary origin trust.

Critical: 6 High: 12

- Only Virtual MFA for Root account (CWE-1032) — Critical cloud risk
- EKS Cluster Control Plane access unrestricted (CWE-1032)
- CORS — Arbitrary Origin Trusted (CWE-942)
- CVE-2024-2698 nginx — Critical misconfiguration (CVSS 9.1)
- Dormant IAM users with active credentials (Bodgeit)

A04 Insecure Design

Resource & Logic Flaws — Both Products

24 mapped

Insecure design is the highest-volume High-severity category, spanning resource exhaustion (CWE-400), uncontrolled recursion (CWE-674), and unchecked return values (CWE-252) in core library dependencies. These represent architectural risks requiring design-level fixes, not simple patches.

Critical: 0 High: 24

- golang.org/x/net/html DoS (CWE-400) — Uncontrolled resource consumption
- Libexpat Uncontrolled Recursion (CWE-674) — Bodgeit & Azure
- Libxml2 Unchecked Return Value (CWE-252)
- CVE-2020-8279 systemd logic flaw (CVSS 9.6)
- jQuery Eval() security vulnerability — unsafe code execution pattern

A01 Broken Access Control

Path Traversal, Info Disclosure

12 mapped

Access control weaknesses detected include path traversal vulnerabilities (CWE-22, CWE-441) in Azure Prod Blue's SAST scan, and information exposure via jQuery CVE-2007-2379 (CWE-200) in Bodgeit. No Critical findings in this category, but active high-severity path traversal is weaponizable for file disclosure.

Critical: 0 High: 12

- Path Traversal — security.audit.path-traversal line 197 (CWE-441)
- Directory Traversal — Setuptools (CWE-22, Bodgeit)
- Information Exposure — jquery 2.1.4 CVE-2007-2379 (CWE-200)
- Cross-site info disclosure via CORS misconfiguration

A07 Identification & Auth Failures

Hardcoded Creds, Weak Certificates

10 mapped

Authentication failures include cleartext password submission, SSL self-signed certificates, and hardcoded credentials (CWE-798). The SSL self-signed certificate finding on Bodgeit is Critical — indicating unverifiable server identity in a production-accessible service.

Critical: 0 High: 10

- SSL Self-Signed Certificate (Critical) — Bodgeit production endpoint
- Cleartext Submission of Password (Bodgeit)
- CVE-2015-2094 TLS certificate validation (CWE-295)
- Hardcoded credentials detected (CWE-798, Bodgeit)
- IAM user with no recent activity (authentication hygiene)

A10 Server-Side Request Forgery (SSRF)

Publicly Exposed Assets — Bodgeit

6 Critical

All 6 SSRF-category findings are Critical severity. Bodgeit carries multiple "Publicly Exposed" findings including a container vulnerable to Log4Shell (CVE-2021-44228) and a publicly exposed VM/serverless with high Kubernetes privileges. These represent the highest-urgency remediation targets given Bodgeit's internet-accessible status.

Critical: 6 High: 0

- Publicly Exposed VM vulnerable to Log4Shell CVE-2021-44228 🔴
- Publicly Exposed Container with High K8s Privileges + Vulnerable Image
- Publicly Exposed Data Asset with Sensitive Financial Data
- AI Model trained on dataset with sensitive data (publicly accessible)
- Unusual activity from previously unseen country (threat indicator)

A08 A08 Software Integrity & A09 Logging Failures

Malware & Audit Gap Indicators

6 total

A08 (2 Critical): Malware findings — VM and serverless assets infected with high/critical severity malware — indicate potential supply chain or software integrity compromises in Bodgeit's environment.

A09 (4 total: 2C/2H): Logging sensitive data in logs (CWE-779), insufficient logging patterns (CWE-388), and absence of monitoring indicators from both products suggest incident response capability gaps.

A08 Critical: 2 A09 Critical: 2 A09 High: 2

- VM/Serverless infected with High/Critical Malware (A08) 🔴
- Sensitive data logged to accessible storage (A09, CWE-779)
- Insufficient error handling and audit trail (A09, CWE-388)

OWASP Coverage Gaps — Scanning & Testing Blind Spots

ℹ️ **All 10 OWASP categories have confirmed active findings** across Azure Prod Blue and Bodgeit. This is positive from a scanning coverage perspective — your tooling is detecting vulnerabilities across the full OWASP attack surface. However, **low finding counts in certain categories may indicate testing gaps rather than genuine absence of vulnerabilities** (see below).

⚠ A02: Cryptographic Failures — Under-Reported Risk

Only 8 mapped cryptographic findings despite both products running cryptographic operations. This low count likely reflects insufficient DAST/cryptographic testing coverage rather than genuine absence of issues. Recommend adding dedicated TLS configuration scanning (e.g., SSLyze, testssl.sh) and reviewing encryption-at-rest configurations in both products. Specific concern: Bodgeit's self-signed SSL certificate suggests TLS hygiene issues extend beyond the single confirmed finding.

⚠ A08: Software & Data Integrity Failures — Likely Under-Scanned

Only 2 findings mapped to A08 (software integrity). This category covers CI/CD pipeline integrity, dependency confusion attacks, and unsafe deserialization — areas not typically covered by standard SCA/SAST scans. Recommend adding pipeline integrity checks (e.g., SLSA framework verification), dependency confusion testing, and reviewing whether serialization libraries (Jackson, Pickle, dom4j — which appears in Bodgeit) are validated against unsafe deserialization.

⚠ A10: SSRF — All Findings Are Threat Intelligence, Not DAST Confirmed

The 6 SSRF-category findings originate from cloud threat detection (malware, unusual country access, exposed assets) rather than confirmed SSRF exploit findings. Dedicated SSRF testing (internal URL probing, cloud metadata endpoint testing) should be added to engagements for both products to confirm or rule out actual SSRF vulnerabilities.

Product Comparison — Azure Prod Blue vs. Bodgeit

OWASP CATEGORY	AZURE PROD BLUE (C)	AZURE PROD BLUE (H)	BODGEIT (C)	BODGEIT (H)	COMBINED TOTAL	WORSE PRODUCT
A01 — Broken Access Control	0	6	0	6	**12**	Tied
A02 — Cryptographic Failures	1	3	1	3	**8**	Tied
A03 — Injection	1	8	1	8	**18**	⚠ Bodgeit (Log4Shell)
A04 — Insecure Design	0	12	0	12	**24**	Tied
A05 — Security Misconfiguration	3	6	3	6	**18**	🔴 Bodgeit (MFA, EKS)
A06 — Vulnerable Components	1	5	1	5	**46***	🔴 Azure (container CVEs)
A07 — Auth Failures	0	5	0	5	**10**	🔴 Bodgeit (SSL self-signed)
	11	**46**	**11**	**46**	**114 mapped**	**Bodgeit = Higher Business Risk**

OWASP CATEGORY	AZURE PROD BLUE (C)	AZURE PROD BLUE (H)	BODGEIT (C)	BODGEIT (H)	COMBINED TOTAL	WORSE PRODUCT
A08 — SW & Data Integrity	1	0	1	0	**2**	🔴 Bodgeit (malware)
A09 — Logging & Monitoring	1	1	1	1	**4**	Tied
A10 — SSRF	3	0	3	0	**6**	🔴 Bodgeit (publicly exposed)
	11	**46**	**11**	**46**	**114 mapped**	**Bodgeit = Higher Business Risk**

⚠️ **Bodgeit carries significantly higher business risk** despite similar finding counts to Azure Prod Blue. Bodgeit is **internet-accessible, handles 10M user records, and generates $10M revenue** — while hosting publicly exposed malware infections, Log4Shell exposure, missing root MFA, and a self-signed SSL certificate in production. Prioritize Bodgeit remediation as the more urgent asset.

Top 3 Priority Recommendations

1

A06 + A03 + A10 · EMERGENCY SPRINT

Patch Critical CVEs & Eliminate Active Injection Vulnerabilities

The single highest-ROI action is an emergency remediation sprint targeting the **known-exploited CVEs (CVE-2022-4135 Bash KEV, CVE-2021-44228 Log4Shell)** and confirmed **SQL injection findings** in Azure Prod Blue and Bodgeit. With 826 active Critical/High findings combined, systematic container image patching and elimination of injection flaws in application code will eliminate the highest-probability breach vectors. Log4Shell on a publicly exposed, internet-facing asset (Bodgeit) is an active exploitation risk that cannot wait for a standard sprint cycle.

1. Rebuild Azure Prod Blue container base image (demo-app:latest) with updated Debian 12 packages — eliminates the majority of A06 CVEs in one action
2. Upgrade/patch Log4j on Bodgeit to 2.17.1+ immediately — remove internet exposure of Log4Shell-vulnerable endpoint
3. Remediate SQL injection in Assignment5.java, Assignment6.java (Bodgeit) and pg-sqli lines 353/286 (Azure) using parameterized queries
4. Enable automated SCA scanning in CI/CD pipeline to prevent new vulnerable components from reaching production

⏱ **Timeline: 2–3 weeks** 💰 **Effort: Medium** 📉 **Expected Risk Reduction: ~55%** 🎯 **Targets: A03, A06, A10**

2

A05 + A07 · WITHIN 30 DAYS

Harden Cloud Infrastructure Configuration & Authentication

Bodgeit's cloud environment has **Critical misconfiguration findings** including missing hardware MFA for the root account and unrestricted EKS cluster control plane access — both are zero-effort, zero-cost remediations with immediate risk elimination. Simultaneously, the **self-signed SSL certificate in production** must be replaced with a CA-signed certificate, and cleartext password submission eliminated. CORS misconfiguration (CWE-942) allowing arbitrary origins is a direct data exfiltration enabler on Azure.

1. Enable hardware MFA on AWS root account and all privileged IAM users (Bodgeit) — 15-minute fix
2. Restrict EKS cluster control plane access to authorized CIDR ranges only
3. Replace self-signed SSL certificate with CA-signed certificate on Bodgeit production endpoints
4. Fix CORS policy on Azure Prod Blue — whitelist specific trusted origins, reject wildcard
5. Enforce HTTPS and implement HSTS headers to eliminate cleartext password submission

⏱ **Timeline: 1–4 weeks** 💰 **Effort: Low–Medium** 📉 **Expected Risk Reduction: ~25%** 🎯 **Targets: A05, A07, A02**

3

A04 + A08 + A09 · 60–90 DAY PROGRAM

Implement Secure Design Review, Integrity Controls & Detection Coverage

The 24 Insecure Design findings and malware infections (A08) point to systemic architectural issues requiring structured remediation. Insecure Design flaws cannot be patched — they require design-level intervention: refactoring resource-handling code, eliminating unsafe recursion, and removing eval()-based execution patterns. The malware infections on Bodgeit require incident response investigation, followed by implementing runtime integrity controls (e.g., Falco, Sysdig) and file integrity monitoring. A09 logging gaps should be closed with centralized SIEM integration to meet SOC 2 and incident response requirements.

1. Initiate incident response investigation on Bodgeit malware infections — determine blast radius and scope of compromise
2. Replace jQuery eval() patterns and unsafe recursion in Libexpat/Libxml2 dependents with safe alternatives
3. Implement resource consumption limits (rate limiting, timeout controls) for DoS-exposed components (CWE-400)
4. Deploy centralized logging (SIEM integration via DefectDojo) and establish alerting thresholds for both products
5. Add SLSA Level 2 build provenance verification to CI/CD pipelines for both products

⏱ **Timeline: 60–90 days** 💰 **Effort: High** 📉 **Expected Risk Reduction: ~15%** 🎯 **Targets: A04, A08, A09**

90-Day Remediation Roadmap

Phase 1 — Days 1–21
Emergency / Stop-the-Bleeding

- ✓ Rebuild container base images (Azure Prod Blue)
- ✓ Patch Log4Shell on Bodgeit + remove internet exposure
- ✓ Enable root MFA on AWS (Bodgeit) — immediate
- ✓ Restrict EKS control plane access
- ✓ Begin malware incident response on Bodgeit
- ✓ Fix SQL Injection in Assignment5.java, 6.java

Phase 2 — Days 22–45
Hardening Sprint

- ✓ Replace self-signed SSL cert (Bodgeit)
- ✓ Fix CORS misconfiguration (Azure Prod Blue)
- ✓ Enforce HTTPS + HSTS on both products
- ✓ Upgrade vulnerable JS/Python libraries
- ✓ Enable SCA in CI/CD pipelines
- ✓ Remediate pg-sqli findings (Azure Prod Blue)

Phase 3 — Days 46–90
Architectural & Detection Uplift

- ✓ Refactor eval() and unsafe recursion patterns
- ✓ Implement SIEM / centralized logging
- ✓ Deploy runtime integrity monitoring (Falco)
- ✓ Add SLSA build provenance to CI/CD
- ✓ Conduct SSRF-specific DAST test cycle
- ✓ Validate OWASP coverage rescan via DefectDojo

Success Metrics & Compliance Targets

📊 Target KPIs — 90-Day Post-Remediation Goals

Metric	Current State	90-Day Target	Method
OWASP Compliance Score	48%	≥ 75%	Rescan + DefectDojo metrics
Critical Findings (both products)	~166 est.	≤ 20	Patch + container rebuild
KEV / Known-Exploited CVEs	≥ 1 confirmed	0	Immediate patching policy
Internet-Exposed Injection Findings	5+	0	Code fix + DAST verification

Metric	Current State	90-Day Target	Method
OWASP Categories with Criticals	5 of 10	≤ 1 of 10	Phase 1 + 2 remediation
Mean Time to Remediate (Critical)	Unknown	≤ 14 days	DefectDojo SLA tracking