Critical OWASP Top 10 Findings in Legacy Web Applications

To: CISO & Application Development Management

From: GRC Audit Team Date:

Date: 9/25/2025

1. Executive Summary

The assessment of the OWASP server revealed multiple severe web application vulnerabilities mapped to the OWASP Top 10 (2021). These issues highlight weaknesses in input validation, authentication, and secure configuration, exposing the applications to risks of data breaches, account compromise, and system takeover.
The findings demonstrate the absence of robust secure coding practices and monitoring controls, which, if left unresolved, could be exploited by attackers to compromise sensitive data and disrupt business operations.

2. Key Findings & Risks

- **Critical Risk – Broken Access Control (A01:2021):**
  Administrative interfaces and functions lacked sufficient authorization checks. Exploiting this could allow privilege escalation and unauthorized access to sensitive operations.
- **High Risk – Cross-Site Scripting (A03:2021):**
  Reflected and DOM-based XSS vulnerabilities were identified, which could allow attackers to steal session tokens, perform actions on behalf of users, or inject malicious code.
- **High Risk – Cryptographic Failures (A02:2021):**
  Weak TLS configuration and lack of proper certificate validation expose login credentials and session data to interception.
- **High Risk – Identification & Authentication Failures (A07:2021):**
  Weak/default credentials (e.g., 12345) were accepted without enforcement of strong password policies, leading to account compromise risk.
- **High Risk – Software & Data Integrity Failures (A08:2021):**
  Outdated components (ASP.NET, PHP frameworks) contained known CVEs, potentially allowing remote code execution or full system compromise.

- **Medium Risk – Security Logging & Monitoring Failures (A09:2021):**
  Applications lacked monitoring or alerts for brute-force and authentication anomalies, enabling attackers to operate undetected.

Triage and Risk Assessment; Risk Assessment table mapped to Owasp top 10

OWASP Top 10 Category	Vulnerability	Example Affected Application	Inherent Risk (L/M/H)	Compliance Violation (e.g., PCI DSS, NIST)	Business Impact
A01: Broken Access Control	Unprotected admin/customer pages accessible without proper restrictions	WebGoatCoins login page	H	PCI DSS 7.2.1 (restrict access to system components)	Unauthorized access, data leakage, privilege escalation
A02: Cryptographic Failures	Credentials transmitted in plaintext (no HTTPS, weak session cookies)	CustomerLogin.aspx (POST request, plaintext creds)	H	PCI DSS 4.1, NIST CSF PR.DS-2	Credential theft, account takeover, espionage
A03: Injection	Unsanitized user inputs (possible SQL injection vectors in login form)	CustomerLogin.aspx username/password fields	H	PCI DSS 6.5.1	Full database compromise, data theft
A05: Security Misconfiguration	Directory indexing enabled, verbose error messages, outdated server headers	WebGoatCoins / directories	M	CIS 11.2, NIST CSF PR.IP-1	Information disclosure, attacker reconnaissance
A07: Identification & Authentication Failures	Weak/Default passwords accepted without enforcement	Login form, Burp intercepted request with weak creds (12345)	H	PCI DSS 8.2.3	Unauthorized access, compromised accounts
A08: Software & Data Integrity Failures	Outdated frameworks (ASP.NET, PHP versions) exposing known CVEs	WebGoat server components	H	CIS 7.1 (patch mgmt), PCI DSS 6.2	Remote code execution, full system compromise

A09: Security Logging & Monitoring Failures	Lack of monitoring/alerting for brute force/login attempts	WebGoatCoins login	M	PCI DSS 10.2	Prolonged undetected attacks, delayed response

3. Recommended Actions

Immediate Mitigation

- Restrict access to vulnerable applications to trusted internal networks.
- Disable exposed administrative pages and enforce access controls immediately.

Remediation

- Fix vulnerabilities using OWASP Proactive Controls (input validation, output encoding, secure authentication).
- Enforce strong TLS configurations with valid certificates.
- Apply patch management to update outdated frameworks and libraries.

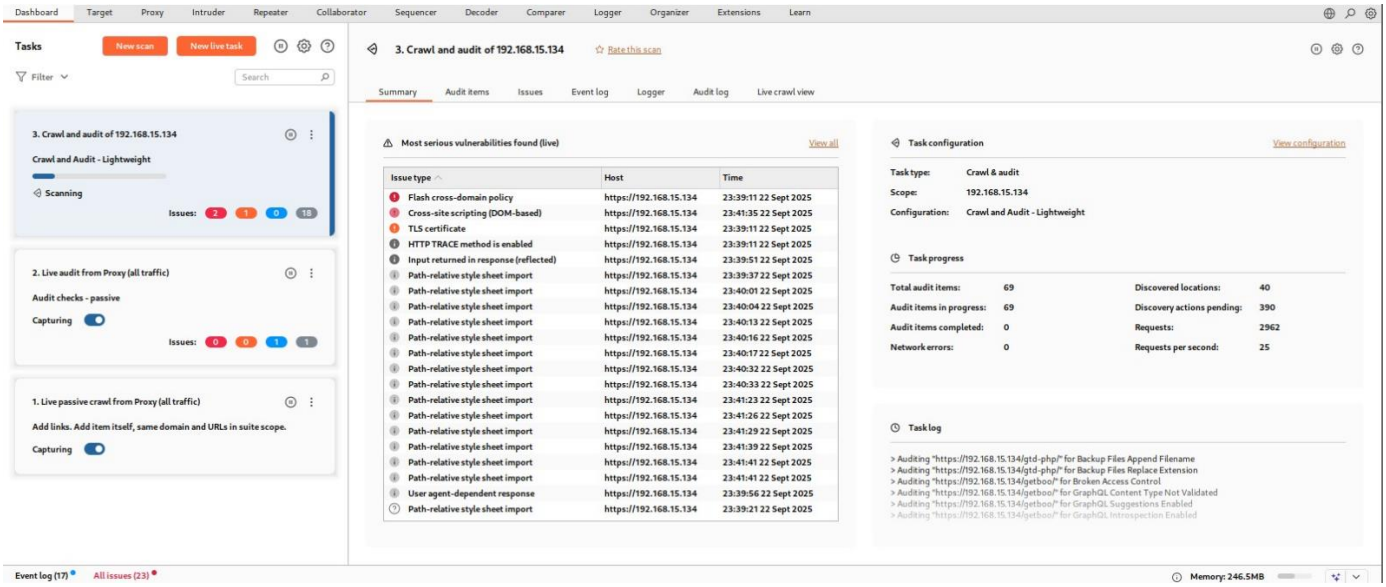
Process Improvement

- Train development teams on secure coding aligned with the OWASP Top 10.
- Integrate automated scanning (SAST/DAST) into the CI/CD pipeline to catch vulnerabilities early.

Ongoing Security Assurance

- Conduct regular penetration tests (internal and third-party).
- Establish centralized logging and monitoring with alerting for suspicious activities.
- Adopt a vulnerability management program for continuous risk reduction.

Appendices A; Dashboard view of Burbsuite showing the scanned results



Appendices B; Vulnerability test from the Repeater tab