

DOI: https://doi.org/10.48009/1_iis_2026_147

Auditing web application vulnerabilities using the Open Worldwide Application Security Project (OWASP) Top Ten

Jose Lineros, *University of North Texas*, jose.lineros@unt.edu

Abstract

Web application vulnerabilities create significant cybersecurity and control risk in systems that support transactions, reporting, and operational processes. Although the Open Worldwide Application Security Project (OWASP) Top Ten is widely used to classify application-security risks, it does not provide a complete audit method for translating those risks into control objectives, audit procedures, or evidence requirements. Using design science research, this paper develops an audit-oriented artifact that maps OWASP Top Ten categories to risk conditions, control failures, audit procedures, evidence sources, and illustrative testing techniques. The artifact integrates OWASP with information technology (IT) audit and control logic drawn from the Control Objectives for Information and Related Technologies (COBIT) framework, the American Institute of Certified Public Accountants (AICPA) Trust Services Criteria, and the National Institute of Standards and Technology (NIST) Cybersecurity Framework. It is demonstrated through retrospective applications to Equifax, SolarWinds, Capital One, LinkedIn, and Target to show how application vulnerabilities can be interpreted as auditable control failures. The paper contributes a structured framework for IT audit practice and accounting information systems (AIS) education while identifying opportunities for future empirical validation and artificial intelligence (AI)-assisted audit support.

Keywords: information technology audit, web application security, Open Worldwide Application Security Project (OWASP) Top Ten, design science research, cybersecurity controls, audit evidence

Introduction

Web application vulnerabilities remain a significant source of cybersecurity risk, particularly in systems that support financial transactions, reporting processes, customer-facing services, and core operational activities. Common weaknesses such as broken access control, injection, insecure configurations, weak authentication, and inadequate logging continue to be actively exploited, often resulting in unauthorized access, data exposure, system disruption, and control breakdowns. These weaknesses frequently arise not only from coding errors but also from deficiencies in control design, secure configuration, change governance, monitoring, and management oversight, making them directly relevant to information systems auditors (ISACA, 2018; Open Worldwide Application Security Project [OWASP], 2025).

The Open Worldwide Application Security Project (OWASP) Top Ten provides a widely recognized taxonomy for the most critical categories of web application security risk (OWASP, 2025). For developers and security teams, the taxonomy is valuable because it provides a common language for discussing

vulnerabilities and prioritizing remediation. For auditors, however, the OWASP categories are only the starting point. Audit work requires further translation from technical risk categories into control objectives, testable control failures, audit procedures, and evidence that can support a defensible conclusion.

Existing information technology (IT) governance and assurance frameworks provide useful but incomplete guidance for this translation. The Control Objectives for Information and Related Technologies (COBIT) framework provides governance and control objectives for enterprise technology management (ISACA, 2018), the Trust Services Criteria issued by the American Institute of Certified Public Accountants (AICPA, 2023) provide assurance criteria for security, availability, confidentiality, privacy, and processing integrity, and the National Institute of Standards and Technology (NIST, 2024) Cybersecurity Framework provides a broader structure for identifying, protecting, detecting, responding to, recovering from, and governing cybersecurity risk. None of these frameworks, by themselves, give auditors a direct OWASP category-to-procedure-to-evidence artifact for application-security audit work.

This paper addresses that gap by developing an audit-oriented framework that aligns OWASP Top Ten categories with risk conditions, control failures, audit procedures, and evidence sources. The objective is not to redefine OWASP or claim that the underlying vulnerability categories are new. Rather, the objective is to operationalize those categories for audit planning, fieldwork, reporting, and education so that application-layer vulnerabilities can be evaluated as control issues rather than treated only as technical findings.

This study makes three contributions to information systems audit research, practice, and education. First, it contributes a design artifact that translates OWASP Top Ten categories into audit-testable risk conditions, control failures, audit procedures, and evidence sources. Second, it extends IT audit practice by connecting application-security weaknesses to control evaluation rather than treating them only as isolated technical vulnerabilities. Third, it supports accounting information systems (AIS) and IT audit education by providing a structured bridge between cybersecurity concepts and audit evidence. These contributions are intentionally design oriented. The study does not claim that the OWASP categories are new but instead shows how they can be translated into a repeatable audit approach.

Background and Literature Review

Web applications support financial reporting, transaction processing, customer portals, administrative workflows, and integrations with cloud and third-party services. From an AIS perspective, these applications are part of the broader environment through which organizations capture, process, store, and communicate accounting and operational data (Romney et al., 2021). Because these systems process, validate, store, and transmit organizational data, application-layer vulnerabilities can create risks involving unauthorized access, data manipulation, service disruption, fraud exposure, privacy violations, and regulatory consequences.

The OWASP Top Ten 2025 release identifies ten categories that are especially relevant to web application security: A01 Broken Access Control; A02 Security Misconfiguration; A03 Software Supply Chain Failures; A04 Cryptographic Failures; A05 Injection; A06 Insecure Design; A07 Authentication Failures; A08 Software or Data Integrity Failures; A09 Security Logging and Alerting Failures; and A10 Mishandling of Exceptional Conditions (OWASP, 2025). These categories provide a current application-risk taxonomy, but the various taxonomies must be translated into audit work before they can guide control testing.

COBIT, the AICPA Trust Services Criteria, and the NIST Cybersecurity Framework provide complementary control perspectives. COBIT supports governance, risk management, change control, and control-monitoring logic. The AICPA Trust Services Criteria support an assurance orientation by emphasizing security, availability, confidentiality, privacy, and processing integrity. The NIST Cybersecurity Framework supports a lifecycle view of cybersecurity risk through identify, protect, detect, respond, recover, and govern functions. These frameworks are useful for audit planning, but they operate at a broader control level than individual OWASP categories.

Taken together, these frameworks provide complementary but incomplete guidance for application-security auditing. OWASP identifies important web application risk categories, while COBIT, the AICPA Trust Services Criteria, and the NIST Cybersecurity Framework provide broader control and governance concepts. However, auditors still need a more operational bridge between vulnerability categories and audit work, specifically what control failures should be considered, what procedures should be performed, and what evidence should be retained. This gap justifies the development of a design artifact that translates application-security taxonomies into audit-executable procedures.

Design science research is appropriate for this problem because the research objective is the development and communication of a purposeful artifact for a recognized professional problem (Hevner et al., 2004; Peffers et al., 2007). In this context, the artifact is not a theory test or statistical model. It is a structured framework intended to help auditors translate technical vulnerability categories into control evaluation logic.

Table 1. Relationship of Source Frameworks to This Study

Framework	Primary role	Relationship to this study
OWASP Top Ten	Application-risk taxonomy	Provides the vulnerability categories used as the artifact's organizing structure.
COBIT	IT governance and control framework	Supports the audit-control logic used to organize control concerns and procedures.
AICPA Trust Services Criteria	Assurance/control criteria	Supports the security, availability, confidentiality, privacy, and processing-integrity orientation.
NIST Cybersecurity Framework	Cybersecurity risk-management framework	Supports the broader identify, protect, detect, respond, recover, and govern framing.

Research Questions

This study is guided by two research questions. Research Question 1 (RQ1): How can OWASP Top Ten web application vulnerability categories be translated into audit-testable control failures, procedures, and evidence sources? Research Question 2 (RQ2): How can the resulting artifact be demonstrated using public cybersecurity incidents involving application-layer or related control failures?

Methodology

This study follows a design science research (DSR) approach because its objective is to develop and communicate a purposeful artifact for a recognized professional problem. Specifically, the problem is that auditors may recognize OWASP vulnerability categories but lack a repeatable method for translating those

categories into control failures, audit procedures, and evidence requirements. The methodology follows the DSR emphasis on problem identification, objective definition, artifact design and development, demonstration, evaluation, and communication (Hevner et al., 2004; Peffers et al., 2007).

The artifact was developed in five steps. First, the OWASP Top Ten 2025 categories were selected as the organizing risk taxonomy because they represent a widely used and current vocabulary for web application risk. Second, each category was interpreted through IT audit and control concepts drawn from COBIT, the AICPA Trust Services Criteria, and the NIST Cybersecurity Framework. Third, each category was mapped to common control failures that could be evaluated during an audit engagement. Fourth, audit procedures and evidence sources were identified for each category, so the artifact would support audit planning, fieldwork, and reporting. Fifth, the artifact was demonstrated through retrospective applications to Equifax, SolarWinds, Capital One, LinkedIn, and Target.

The cases were selected because they represent distinct patterns of application-layer or related control failures. Among them were unpatched components and vulnerability remediation weaknesses, software supply chain compromises, cloud/application misconfigurations, weak credential protections, and third-party access and monitoring failures. The demonstration is a preliminary observational evaluation rather than a summative validation. The cases show how the artifact can be applied to public incidents to identify risk conditions, control failures, audit procedures, and evidence sources. The study therefore evaluates the artifact's practical plausibility and communicability, but it does not claim empirical proof that the artifact improves audit effectiveness.

Table 2. Design Science Research Application

DSR element	Application in this study
Problem identification	OWASP risks are not consistently translated into audit procedures, control failures, and evidence requirements.
Objective of the solution	Develop an audit artifact mapping OWASP categories to risk conditions, controls, procedures, and evidence.
Design and development	Synthesize OWASP categories with IT audit control logic and evidence-oriented procedures.
Demonstration	Apply the artifact retrospectively to Equifax, SolarWinds, Capital One, LinkedIn, and Target.
Preliminary evaluation	Assess practical plausibility and communicability through observational case application.
Communication	Present implications for audit research, practice, AIS design, and audit education.

Presentation of the Main Contribution

The main contribution of this paper is an OWASP-based audit artifact that translates application-security categories into audit workpaper logic. The artifact was designed around four principles. First, auditability means that each OWASP category must be translated into a condition that can be tested or evaluated. Second, evidence orientation means that each procedure should identify the documents, logs, configurations, scans, approvals, or other evidence needed to support an audit conclusion. Third, framework alignment means that the artifact should complement rather than replace established governance and

assurance frameworks. And fourth, educational transferability means that the artifact should be understandable to AIS and IT audit learners while remaining useful to practitioners.

Table 3 summarizes the proposed artifact. It links each OWASP category to a related risk condition, primary control concern, example audit procedure, and example evidence source. The table is intended to function as a planning aid, testing guide, and teaching artifact.

Table 3. Summary of Proposed OWASP-Based Audit Artifact

OWASP category	Risk condition	Primary control concern	Example audit procedure	Example evidence
A01 Broken Access Control	Unauthorized access to data or functions	RBAC and authorization enforcement	Test access across user roles and attempt IDOR scenarios	Role matrices, access logs, authorization rules
A02 Security Misconfiguration	Exposure from insecure settings	Configuration hardening	Review baselines and scan for missing headers or open ports	Configuration files, scan reports
A03 Software Supply Chain Failures	Vulnerable or compromised components	Dependency and pipeline governance	Review SBOM and SCA scan results	SBOM, CVE reports, CI/CD logs
A04 Cryptographic Failures	Exposure of sensitive data	Encryption and key management	Test TLS and review key controls	TLS reports, key management logs
A05 Injection	Malicious input execution	Input validation and query handling	Test input fields and review parameterized queries	Test results, validation rules
A06 Insecure Design	Design-level vulnerabilities	Secure SDLC and threat modeling	Review threat models and misuse cases	SDLC documentation, design records
A07 Authentication Failures	Unauthorized access through weak authentication	MFA, password, and session controls	Test MFA, lockout, and session expiration	Authentication logs, password policies
A08 Software or Data Integrity Failures	Use of untrusted code or data	Integrity validation	Review code signing and update validation	Signing records, update logs
A09 Security Logging and Alerting Failures	Failure to detect or respond to security events	Logging, SIEM, and alerting	Trigger events and verify alerts	Logs, alerts, escalation records
A10 Mishandling of Exceptional Conditions	Exploitable error states	Error handling and resilience	Perform negative testing and review rollback controls	Error logs, rollback records

Note: RBAC = role-based access control; IDOR = insecure direct object reference; SBOM = software bill of materials; SCA = software composition analysis; CVE = common vulnerabilities and exposures; CI/CD = continuous integration/continuous deployment; TLS = transport layer security; SDLC = system development life cycle; MFA = multi-factor authentication; SIEM = security information and event management.

Auditors can use the artifact during planning to identify relevant application-layer risks during fieldwork when selecting their control tests and evidence sources. During reporting, the artifact can help them connect technical findings to control deficiencies. For example, an injection finding would not be reported solely as

a technical vulnerability, but instead connected to input validation, query handling, test evidence, affected business processes, and corrective control recommendations.

The artifact can also be applied by grouping audit procedures into preventive, detective, and corrective control domains. Preventive procedures focus on whether applications are designed, configured, and deployed to reduce the likelihood of exploitable weaknesses. Detective procedures focus on whether vulnerabilities, unauthorized activity, and abnormal patterns are identified in a timely manner. Corrective procedures focus on whether the organization remediates vulnerabilities, responds to incidents, and restores secure operations.

Table 4. Example Control Domains, Audit Procedures, and Evidence Sources

Control domain	Control objective	Example audit procedures	Example evidence
Preventive	Access control enforcement	Evaluate RBAC, least privilege, application programming interface (API) authorization, and IDOR resistance.	Role matrices, provisioning records, access logs, authorization rules
Preventive	Secure input handling	Test input fields, review parameterized query use, and inspect validation rules.	Test results, code/configuration extracts, validation standards
Preventive	Secure configuration and cryptography	Compare configurations to baselines, test TLS, review key controls, and assess security headers.	Configuration snapshots, TLS reports, key-management logs, baseline exceptions
Preventive	Secure development and supply chain governance	Review threat models, SBOMs, SCA results, build controls, and release approvals.	SDLC records, SBOMs, SCA reports, CI/CD logs, code-signing records
Detective	Logging, monitoring, and vulnerability detection	Trigger test events, review SIEM rules, examine vulnerability scans, and assess escalation workflows.	Log samples, alert records, scan reports, incident tickets
Corrective	Incident response and recovery	Review response plans, post-incident analyses, remediation status, rollback controls, and recovery evidence.	Response plans, remediation tickets, after-action reports, recovery logs

Testing Techniques and Evidence Collection

Testing techniques should be selected because they support audit evidence, not because a tool list appears comprehensive. Configuration and header testing may include TLS and security-header scans; input validation testing may use controlled payloads and request/response capture; authentication and session testing may examine lockout thresholds, MFA enforcement, cookie attributes, and session expiration; access-control testing may compare user roles and API permissions; dependency scanning may identify vulnerable components and remediation status; and log testing may verify that relevant events generate alerts and escalation records. Examples of tools include Burp Suite, OWASP Zed Attack Proxy (ZAP),

Nmap, Qualys Secure Sockets Layer (SSL) Labs, Snyk, OWASP Dependency-Check, Nessus, Qualys, Splunk, Microsoft Sentinel, SonarQube, and Checkmarx.

These examples are illustrative rather than exhaustive. Evidence retained from these procedures should support both the technical finding and the related control conclusion. Strong evidence may include system-generated logs, configuration exports, screenshots of controlled tests, vulnerability scan reports, source-code review results, SIEM alerts, ticket histories, exception approvals, and remediation records. The audit emphasis remains on whether the control is designed, implemented, and operating effectively, not merely whether a vulnerability can be identified.

Demonstration and Findings

To demonstrate the artifact's practical applicability, this section applies the framework to five retrospective cybersecurity incidents: Equifax, SolarWinds, Capital One, LinkedIn, and Target. The cases are not presented as statistical validation or proof of effectiveness. Rather, they serve as observational case applications showing how the artifact can be used to identify risk conditions, diagnose control failures, select audit procedures, and define evidence expectations.

Equifax: Unpatched Component and Input Exploitation

Equifax experienced a major breach after a known Apache Struts vulnerability in an internet-facing system was not remediated in a timely manner, exposing sensitive personal information of more than 140 million individuals (U.S. Government Accountability Office, 2018; U.S. House of Representatives, 2018). Applied through the artifact, the incident is not only an injection or component-risk event; it is an audit problem involving asset inventory, vulnerability scanning, patch governance, escalation, and evidence of remediation.

Table 5. Artifact Application to Equifax

Framework element	Application to Equifax
Risk condition	Known vulnerable web application component exposed to attack.
Control failure	Patch management, asset inventory, vulnerability remediation, and escalation controls.
OWASP categories	A03 Software Supply Chain Failures; A05 Injection; A02 Security Misconfiguration.
Audit procedures	Review vulnerability scanning, patch service-level expectations, asset inventory, exception reporting, and escalation procedures.
Evidence	Vulnerability tickets, patch logs, scan results, asset inventory, exception approvals, and escalation records.

SolarWinds: Software Supply Chain Compromise

The SolarWinds compromise involved malicious code inserted into the Orion software build process and distributed through trusted software updates, illustrating a software supply chain and integrity control failure (Cybersecurity and Infrastructure Security Agency, 2020; MITRE ATT&CK, 2023). Applied through the artifact, the case directs audit attention toward build-pipeline access, code-signing governance, release approvals, repository activity, and update validation.

Table 6. Artifact Application to SolarWinds

Framework element	Application to SolarWinds
Risk condition	Compromised software build and update process.
Control failure	Weak CI/CD integrity, code-signing governance, release controls, and update validation.
OWASP categories	A03 Software Supply Chain Failures; A08 Software or Data Integrity Failures.
Audit procedures	Review build pipeline access, release approvals, code signing, dependency integrity, repository activity, and update distribution controls.
Evidence	CI/CD logs, signing records, release approvals, repository access logs, dependency records, and update validation records.

Capital One: Misconfiguration and Data Exposure

The Capital One breach involved unauthorized access to cloud-hosted data and has been discussed as a case involving cloud configuration, access control, and monitoring weaknesses (Capital One, 2019; Office of the Comptroller of the Currency, 2020). Applied through the artifact, the case directs audit work toward IAM (identity and access management) permissions, web application firewall configuration, storage access controls, anomaly detection, and incident-response evidence.

Table 7. Artifact Application to Capital One

Framework element	Application to Capital One
Risk condition	Unauthorized access to cloud-hosted data.
Control failure	Misconfiguration, excessive permissions, weak monitoring, and incident-response gaps.
OWASP categories	A02 Security Misconfiguration; A01 Broken Access Control; A09 Security Logging and Alerting Failures.
Audit procedures	Review cloud IAM permissions, WAF (web application firewall) configuration, storage access controls, anomaly alerts, and incident-response records.
Evidence	IAM policies, configuration snapshots, access logs, SIEM alerts, incident tickets, and remediation records.

LinkedIn: Weak Credential Protection

LinkedIn experienced exposure of user credentials after compromise of password storage mechanisms, and the company confirmed that some compromised passwords corresponded to LinkedIn accounts and required affected members to reset their passwords (Silveira, 2012; Huh et al., 2017). Applied through the artifact, the incident directs audit attention toward credential storage, hashing and salting, password-reset governance, MFA adoption, authentication monitoring, and detection of unusual login activity.

Table 8. Artifact Application to LinkedIn

Framework element	Application to LinkedIn
Risk condition	Exposure of user credentials through weak or compromised credential-protection mechanisms.
Control failure	Weak password hashing, insufficient salting, inadequate credential-protection governance, and limited account-compromise monitoring.
OWASP categories	A04 Cryptographic Failures; A07 Authentication Failures; A09 Security Logging and Alerting Failures.
Audit procedures	Review password hashing and salting standards, credential-reset procedures, MFA adoption, authentication event logging, and unusual-login monitoring.
Evidence	Password-storage standards, hashing configuration evidence, credential-reset records, MFA enrollment reports, authentication logs, and account-compromise alerts.

Target: Third-Party Access and Monitoring Failure

Target experienced a breach involving unauthorized access to payment systems through compromised third-party credentials, illustrating vendor access, segmentation, monitoring, and escalation weaknesses (Keskin et al., 2021; U.S. Senate Committee on Commerce, Science, and Transportation, 2014). Applied through the artifact, the case directs audit attention toward third-party access governance, least-privilege enforcement, network segmentation, alert triage, escalation procedures, and incident-response timeliness.

Table 9. Artifact Application to Target

Framework element	Application to Target
Risk condition	Unauthorized access to payment systems through compromised third-party credentials.
Control failure	Weak vendor access controls, inadequate segmentation, insufficient monitoring, and failure to respond effectively to alerts.
OWASP categories	A01 Broken Access Control; A02 Security Misconfiguration; A09 Security Logging and Alerting Failures.
Audit procedures	Review vendor access provisioning, least-privilege enforcement, network segmentation, monitoring rules, alert escalation, and incident-response records.
Evidence	Vendor access lists, IAM records, segmentation diagrams, firewall rules, SIEM alerts, escalation tickets, and incident-response documentation.

Cross-Case Findings

Across the five cases, the artifact shifts analysis from incident description to audit evaluation. Equifax illustrates how a known vulnerability becomes an audit issue when asset inventory, patch governance, vulnerability scanning, and escalation controls fail. SolarWinds illustrates how software supply chain risk becomes auditable through build-pipeline access, code-signing controls, release approvals, and update validation evidence. Capital One illustrates how cloud misconfiguration and excessive permissions can be translated into tests of IAM policies, WAF configuration, storage access controls, monitoring rules, and

incident-response records. LinkedIn illustrates how credential exposure becomes auditable through password hashing, salting, credential-reset governance, MFA adoption, and monitoring for account takeover activity. Target illustrates how third-party access risk becomes auditable through vendor access provisioning, least-privilege enforcement, network segmentation, alert escalation, and incident-response timeliness. The cases therefore demonstrate that OWASP categories are most useful to auditors when they are connected to control failures and evidence expectations rather than treated only as technical labels.

Discussion of Findings

The demonstration suggests that an OWASP-based audit artifact can improve the structure of application-security audit work by connecting risk categories to control failures, procedures, and evidence. This contribution is most important where audit teams already understand cybersecurity terminology but lack a repeatable way to convert technical vulnerability categories into audit workpaper logic. The artifact also supports communication between technical specialists and auditors because it links vulnerability language to familiar control-evaluation concepts.

Implications for Research

For information systems research, the paper contributes a design artifact that can be refined and evaluated in future studies. The artifact creates a basis for comparing structured and unstructured audit planning, examining whether auditors using the artifact identify broader risk coverage, collect stronger evidence, or produce more specific recommendations. It also creates a foundation for studying how technical cybersecurity taxonomies can be translated into professional assurance practice.

Implications for Practice

For practice, the artifact supports more consistent application-security audit procedures across systems, business units, and engagements. It encourages auditors to move beyond vulnerability identification toward control evaluation by asking why a vulnerability exists, which control failed, what evidence supports the conclusion, and how remediation should be monitored. The artifact also expands the evidence base for application-security audits by emphasizing logs, scans, configurations, build records, access rules, alert histories, and remediation tickets in addition to interviews and policy documents.

Implications for AIS and IT Audit Education

For AIS and IT audit education, the artifact provides a structured bridge between technical security concepts and audit practice. Students can use the artifact to connect OWASP categories to control objectives, audit procedures, and evidence sources in systems that support transactions and reporting, which are central concerns in AIS education (Romney et al., 2021). Case exercises based on Equifax, SolarWinds, Capital One, LinkedIn, and Target can help learners practice translating incident narratives into audit plans, control findings, and evidence requests.

Limitations and Future Research

This study has several limitations. First, the artifact is conceptual and has not yet been tested across live audit engagements. Second, the case applications are retrospective and rely on public information; they demonstrate how the artifact can structure analysis but do not prove that it improves audit effectiveness. Third, the artifact's usefulness may vary based on auditor expertise, access to technical evidence, organizational maturity, tool availability, and the complexity of the application environment. Fourth, the framework is aligned with the OWASP Top Ten 2025 categories, so future OWASP revisions may require

updates to category labels, control mappings, and evidence examples. These limitations are important because the paper presents a design artifact and preliminary demonstration, not a completed field validation.

Future research should evaluate whether the artifact improves audit planning, control mapping, evidence selection, and audit documentation when compared with less structured approaches. One path is expert-panel validation with IT auditors, cybersecurity professionals, and AIS educators. A second path is field application in internal audit or IT assurance engagements. A third path is an experimental study in which students or auditors prepare audit plans with and without the artifact, allowing comparison of risk coverage, evidence quality, and recommendation specificity.

Future research should also examine how artificial intelligence and large language models can support control mapping, vulnerability triage, secure code review, evidence summarization, remediation guidance, patch validation, and audit workpaper preparation (Anthropic, 2026a, 2026b; Okdem & Okdem, 2024). Recent frontier cyber models, including Anthropic's Claude Fable 5 and Claude Mythos 5, illustrate the growing defensive potential of these tools in authorized settings. Anthropic's Project Glasswing, for example, describes the use of Mythos-class models to identify high-severity vulnerabilities, support defensive security work, assist with patch writing, and conduct pre-release checks intended to prevent vulnerabilities from reaching production (Anthropic, 2026a, 2026b).

For IT auditors, this development is important because AI-assisted security tools may shorten the path from vulnerability discovery to control evaluation, remediation, and evidence collection. However, auditors should not treat model output as conclusive evidence. AI-assisted audit support may help synthesize large volumes of technical evidence, but it also creates risks involving inaccurate outputs, confidentiality exposure, insecure recommendations, bias, and auditor overreliance. Future studies should examine how these tools can be governed, validated, and documented before they are embedded into formal audit workflows.

Conclusions

This paper develops an audit-oriented design artifact that translates OWASP Top Ten web application vulnerability categories into audit-testable control failures, procedures, and evidence sources. The artifact responds to a practical gap in IT audit and AIS education: auditors may recognize application-security risk categories but still need a structured method for evaluating the related controls. By integrating OWASP with IT audit and control logic, the paper shows how application vulnerabilities can be framed as auditable control issues rather than isolated technical findings. The retrospective demonstrations involving Equifax, SolarWinds, Capital One, LinkedIn, and Target illustrate the artifact's practical use while also identifying the need for future validation. The primary contribution is therefore a structured, evidence-oriented framework that can support audit planning, fieldwork, reporting, and education in web application security contexts.

Acknowledgements

The author used generative AI tools to support editorial revision and organization. However, the author reviewed and verified the final manuscript and takes full responsibility for its content, citations, analysis, and conclusions.

References

- American Institute of Certified Public Accountants. (2023). *2017 Trust Services Criteria for Security, Availability, Processing Integrity, Confidentiality, and Privacy (With Revised Points of Focus - 2022)*. <https://www.aicpa-cima.com/resources/download/2017-trust-services-criteria-with-revised-points-of-focus-2022>
- Anthropic. (2026a, June 9). Claude Fable 5 and Claude Mythos 5. <https://www.anthropic.com/news/claude-fable-5-mythos-5>
- Anthropic. (2026b, June 2). Expanding Project Glasswing. <https://www.anthropic.com/news/expanding-project-glasswing>
- Capital One. (2019). *Information on the Capital One cyber incident*. <https://www.capitalone.com/digital/facts2019/>
- Cybersecurity and Infrastructure Security Agency. (2020, December 17). *Advanced persistent threat compromise of government agencies, critical infrastructure, and private sector organizations (AA20-352A)*. <https://www.cisa.gov/news-events/cybersecurity-advisories/aa20-352a>
- Hevner, A. R., March, S. T., Park, J., & Ram, S. (2004). Design science in information systems research. *MIS Quarterly*, 28(1), 75-105. <https://doi.org/10.2307/25148625>
- Huh, J. H., Kim, H., Rayala, S. S. V. P., Bobba, R. B., & Beznosov, K. (2017). I'm too busy to reset my LinkedIn password: On the effectiveness of password reset emails. In *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems* (pp. 387-391). Association for Computing Machinery. <https://doi.org/10.1145/3025453.3025788>
- ISACA. (2018). *COBIT 2019 framework: Governance and management objectives*. <https://www.isaca.org/resources/cobit>
- Keskin, O. F., Caramancion, K. M., Tatar, I., Raza, O., & Tatar, U. (2021). Cyber third-party risk management: A comparison of non-intrusive risk scoring reports. *Electronics*, 10(10), Article 1168. <https://doi.org/10.3390/electronics10101168>
- MITRE ATT&CK. (2023). *SolarWinds compromise, Campaign C0024*. <https://attack.mitre.org/campaigns/C0024/>
- National Institute of Standards and Technology. (2024). *The NIST Cybersecurity Framework (CSF) 2.0 (NIST CSWP 29)*. U.S. Department of Commerce. <https://doi.org/10.6028/NIST.CSWP.29>
- Office of the Comptroller of the Currency. (2020, August 6). *OCC assesses \$80 million civil money penalty against Capital One*. <https://www.occ.gov/news-issuances/news-releases/2020/nr-occ-2020-101.html>
- Okdem, S., & Okdem, S. (2024). Artificial intelligence in cybersecurity: A review and a case study. *Applied Sciences*, 14(22), Article 10487. <https://doi.org/10.3390/app142210487>
- Open Worldwide Application Security Project. (2025). *OWASP Top 10:2025*. <https://owasp.org/Top10/2025/>
- Peffer, K., Tuunanen, T., Rothenberger, M. A., & Chatterjee, S. (2007). A design science research methodology for information systems research. *Journal of Management Information Systems*, 24(3), 45-77. <https://doi.org/10.2753/MIS0742-1222240302>
- Romney, M. B., Steinbart, P. J., Summers, S. L., & Wood, D. A. (2021). *Accounting information systems* (15th ed.). Pearson.

- Silveira, V. (2012, June 6). *An update on LinkedIn member passwords compromised*. LinkedIn Official Blog. <https://www.linkedin.com/blog/member/archive/linkedin-member-passwords-compromised>
- U.S. Government Accountability Office. (2018). *Data protection: Actions taken by Equifax and federal agencies in response to the 2017 breach* (GAO-18-559). <https://www.gao.gov/products/gao-18-559>
- U.S. House of Representatives, Committee on Oversight and Government Reform. (2018). *The Equifax data breach: Majority staff report, 115th Congress*. <https://oversight.house.gov/wp-content/uploads/2018/12/Equifax-Report.pdf>
- U.S. Senate Committee on Commerce, Science, and Transportation. (2014, March 26). *A "kill chain" analysis of the 2013 Target data breach: Majority staff report for Chairman Rockefeller*. <https://www.commerce.senate.gov/wp-content/uploads/media/doc/2014%200325%20Target%20Kill%20Chain%20Analysis.pdf>