Evaluating Automated Web Security Assessment Accuracy

A Reproducible Regression Study Using OMARINO AmanCheck

OMARINO IT Services inh. Omar Zaror, Germany

Project 2026-001 | 3 November 2026 | Publication edition

Abstract

Automated security assessments can confuse missing evidence with a finding, treat an indicator as proof, or report a correct finding at an ineffective severity. We evaluated seven AmanCheck probes using 32 controlled local scenarios and three executions per version. The inspected baseline and a separately corrected research variant produced 96 observations each. On 31 binary fixture claims, the baseline yielded 14 true positives, six false positives, six false negatives and five true negatives: precision and recall were both 70.0%. A separate resolver-failure case required abstention. Corrections resolved every selected claim, severity label and exact redirect-count assertion in the targeted suite. These results establish reproducible regressions and local fixes, not population-level scanner accuracy. Source, expected outcomes, independently acquired observations, raw JSON/CSV, hashes and local test instructions accompany the paper. Repository history was audited and sensitive deployment material was excluded from the prepared publication package.

Keywords: defensive security, scanner validation, false positives, evidence quality, reproducibility, TLS, web configuration.

1. Research question and contribution

How reliably do the implemented probes distinguish a selected configuration or technology claim from its absence, and when does the scanner overstate or understate the evidence? The contribution is an executable source-informed regression corpus, a measured original baseline, independently acquired fixture observations, and a transparent before/after correction record. The study does not evaluate AI-generated reports, exploitation capability or the full AmanCheck product.

The unit of analysis is a scenario-specific claim. Some claims represent security misconfiguration; others represent redirect tracking or technology identification. Pooling them is useful for describing this suite but must not be interpreted as an estimate of general vulnerability detection accuracy. Perfect agreement after fixing tests selected from the same source is an expected regression-engineering outcome, not evidence of perfect real-world scanning.

2. Materials, scope and authorization

The laboratory work began on 2 November 2026 and final reproduction/package verification continued on 3 November 2026 (Europe/Berlin). Online version checks were performed on 2 November.

The user authorized inspection and local defensive testing of OMARINO-DE/AmanCheck and OMARINO-DE/whmcs-kazawallet. AmanCheck was private and remained private. Its inspected commit was `2108bc91bcf089699ef46eadf75e09ddf1be1e5e`. Only seven C# probe files, their small contracts/enums and the upstream license notice were copied into this research package. Baseline source hashes are recorded individually and checked by the summarizer.

The upstream API/worker targets .NET 8. The extracted probes were compiled without source changes against .NET / ASP.NET Core 10.0.14, SDK 10.0.403, on macOS 27.0.1 Arm64. The same runtime and fixture implementation were used for both variants. Application services, authentication, database, queue, background worker orchestration,

reporting, Python utility tests and frontend were not executed. Findings therefore concern these isolated probe implementations on the stated runtime.

All scanner targets were local Kestrel HTTP/HTTPS listeners or the local UDP DNS fixture. There were no customer targets, real credentials, production scans or payment operations. Ephemeral CA/leaf keys were generated for the laboratory and no root certificate was installed in the system trust store. The native run used process-level fixture separation and loopback scope; it did not have an operating-system-enforced network namespace. A Docker runner with network disabled is provided, but no Docker service was running on the host and that route was not executed.

3. Experimental design and independent checks

The corpus contains four TLS cases, four header cases, four cookie cases, four DNS cases, six exposure cases, five performance cases and five technology cases. Expected outcomes and the severity rubric were encoded in the fixture definitions before baseline execution. Cases deliberately include a secure/clean outcome where feasible, an insecure or positive signal, and parser/transport edge cases. Selection was guided by source inspection; there was no random sampling, blinded annotation or held-out test set.

Each scenario was executed three times per variant. The fixtures emit controlled responses, and a separate acquisition path records raw evidence before invoking the probe. Independent HTTP requests disable automatic redirects and record status, headers, body and measured duration. Exposure controls include both the candidate file and a deliberately missing path. A separate SslStream handshake records certificate policy errors and a custom-root X509Chain check, allowing expiry and hostname problems to be distinguished from the intentionally untrusted lab issuer. Raw UDP DNS queries collect TXT and CAA fixture answers, or a SERVFAIL response.

These checks are independent of the scanner parsers but remain part of the same author-controlled laboratory and runtime. They are not an independent third-party validation. Mozilla HTTP Observatory is a relevant header-assessment reference [4]; its external scanning service was not run against local fixtures. OWASP WSTG cookie and exposure guidance [1,2] informed the distinction between the observed attribute/file and its security interpretation. NIST SP 800-115 [3] provides context for planned scope, evidence analysis and mitigation; this study is not a conformance assessment against that publication.

TLS-02 uses a currently valid leaf signed by a CA absent from the system store; it is intentionally an untrusted-chain case, not a production-trusted HTTPS control. TLS-03 adds expiry, and TLS-04 adds a wrong hostname. A production-trusted positive TLS control and HTTPS-cookie cases are outside this corpus. The separate payment-module study has a configured-root trusted control, but its results are not pooled into AmanCheck metrics.

4. Metric definitions

A prediction is derived from the actual result using the claim mapping in the runner. For TLS, a certificate repair recommendation counts as detection even if the assigned severity is too low; severity is evaluated separately. Header and cookie warnings/failures are positive findings. Exposure uses the exposed-endpoint count. DNS uses assessment status and recommendations. Performance uses observed redirect count or recommendations. Technology uses the number of detected signals.

Precision = TP/(TP+FP), recall = TP/(TP+FN), false-positive rate = FP/(FP+TN), and accuracy = (TP+TN)/(TP+FP+FN+TN). DNS-04 has unknown truth because its resolver fails and is excluded from binary counts. It is retained in the 32-case primary agreement measure, where only abstention is correct. Severity agreement is the proportion matching the study's explicit rubric; it is not CVSS or a consensus severity standard. Exact redirect-count accuracy covers only the two count fixtures.

The three repetitions test execution consistency. They do not turn 32 scenarios into 96 independent configurations. We report confusion counts on unique scenarios; repeated-observation counts are available in the data. No statistical significance or confidence about an external population is claimed. Probe elapsed milliseconds are real local wall-clock measurements and include client setup within ExecuteAsync; they exclude oracle acquisition and server

setup. They are not estimates of internet scan throughput.

5. Results

Measure	Baseline	Corrected research variant
Unique scenarios	32	32
Executed observations	96	96
TP / FP / FN / TN on binary claims	14 / 6 / 6 / 5	20 / 0 / 0 / 11
Binary claim precision	70.0%	100.0%
Binary claim recall	70.0%	100.0%
False-positive rate	54.5%	0.0%
Binary claim accuracy	61.3% (19/31)	100.0% (31/31)
Primary agreement including abstention	59.4% (19/32)	100.0% (32/32)
Study severity agreement	56.3% (18/32)	100.0% (32/32)
Exact redirect-count assertions	0/2	2/2

All three repetitions agreed on predicted claim, observed severity and redirect count for each scenario in each variant. This is consistency within the current fixture environment, not external replication. The corrected DNS resolver-failure case abstained rather than declaring missing security controls. Both versions detected the TLS invalid-certificate cases, but the baseline assigned low severity to untrusted-chain and hostname problems that the rubric classified high.

Probe	Cases	Baseline claim agreement	Fixed claim agreement	Baseline severity agreement	Fixed severity agreement
TLS/HTTPS	4	4/4	4/4	2/4	4/4
HTTP headers	4	2/4	4/4	2/4	4/4
Cookies	4	3/4	4/4	1/4	4/4
DNS evidence	4	2/4	4/4	2/4	4/4
Public exposure	6	2/6	6/6	2/6	6/6
Redirects/performance	5	3/5	5/5	4/5	5/5
Technology signals	5	3/5	5/5	5/5	5/5

6. Reproduced defects and local corrections

TLS/HTTPS. The baseline accepted certificates for analysis, collected chain/name errors and added repair recommendations, but usually returned low severity. The local correction returns a high-severity failure for certificate policy errors. The analytic callback's acceptance is deliberately retained to inspect the certificate; this is scanner evidence gathering, not a recommendation to disable verification in application clients. Protocol/cipher enumeration and revocation behavior are not measured.

HTTP headers. Presence-only checks accepted an invalid X-Frame-Options value and whitespace CSP. The correction validates the two supported frame values and rejects empty CSP. This is a limited repair: it does not fully parse CSP or establish browser enforcement. A restrictive CSP frame-ancestors policy can also be relevant to framing protection; the current corpus does not test every policy combination.

Cookies. The baseline searched Cookie.Comment for SameSite and used an integer-division threshold that elevated a single cookie even without an HttpOnly issue. The correction matches the cookie's raw Set-Cookie field and parses semicolon-delimited attributes, then requires at least one HttpOnly issue before applying that threshold. A Comment containing the text SameSite is not accepted as an attribute. Duplicate cookie names/paths, redirect-set cookies, SameSite=None with missing Secure and browser-specific enforcement remain unvalidated.

DNS. The original TXT/CAA/DMARC/DNSSEC methods returned absent placeholders and could report high-severity absence without querying those record types. The correction introduces an explicit DNS record reader and distinguishes successful empty answers from unavailable evidence. In the lab, a local UDP reader supplies actual TXT/CAA responses. Without a configured reader, or on resolver failure, it returns an unknown informational assessment. It does not claim DNSSEC or DKIM validation. A production resolver integration, complete SPF syntax evaluation, CAA inheritance and DNSSEC cryptographic validation remain future work; the new interface alone is not that implementation.

Public exposure. The baseline included robots.txt/security.txt in the vulnerability endpoint list, accepted catch-all 200 bodies as exposure, depended on Content-Length and skipped GET when HEAD failed. The correction keeps informational files separate, compares a missing-path control, examines bounded GET content and handles chunked/no-length responses and HEAD-405 behavior. Synthetic .env contents demonstrate the selected cases. Dynamic soft-404s, authentication transitions and archive signatures are not fully classified; ambiguous administration HTML is left unassessed.

Redirects/performance. Automatic redirect following concealed hops from the baseline's manual tracking loop. Disabling automatic following allows the existing loop to record the two- and six-hop fixtures correctly. The 503 and injected 1200 ms delay controls also behaved as expected. Timeouts, redirect-limit edges, trusted HTTPS-to-HTTP downgrade and every intermediate protocol transition were not tested, and broader redirect-policy hardening remains outside the measured repair.

Technology fingerprinting. Broad word matching interpreted WordPress discussion text and the word reaction as technology evidence. Selected WordPress/React rules now require structured HTML/asset signals. The read path was also changed to request headers first and read at most 50 KiB instead of downloading the whole body before truncation; the download bound is a source-level improvement, not a separately benchmarked resource result. Other technology signatures remain heuristic and spoofable; a signal never proves a vulnerable installed component.

7. Reproducibility and evidence integrity

The package includes baseline/fixed JSON and CSV, full synthetic independent observations, expected scenarios, runtime metadata, metric definitions, source provenance and individual patches. The native run command rebuilds each probe variant, executes it against temporary fixtures and recomputes metrics. The summarizer checks scenario/repetition counts, metric consistency, baseline file hashes and all corrected assertions. SHA256SUMS covers delivered files; it is an integrity inventory, not a digital signature or trusted timestamp.

The lab does not use NuGet packages: it relies on the installed .NET shared frameworks. Re-execution regenerates ports, timestamps, certificate identities and timings. Reproducibility means matching semantic outcomes under the specified fixture conditions, not byte-identical evidence. Docker build/run instructions are included as a future portability route and explicitly marked unexecuted. The original application build, upstream deployment and reporting stack are separate integration tasks.

8. Sanitization, disclosure and publication status

Both repositories were reviewed through all locally fetched reachable history. The custom blob pass covered 33 commits/171 unique blobs for AmanCheck and 12 commits/42 unique blobs for the payment module. Gitleaks 8.30.1 independently flagged one AmanCheck private key and no payment-module matches in its scanned history. Internal infrastructure references and deployment material were excluded. Candidate regex matches were triaged without testing any credential. Historical test-script credential literals in the payment module were placeholders, not verified production secrets.

The prepared publication is assembled from selected files without upstream Git history. It contains no private key or production configuration. The detailed redacted audit and the payment-module advisory/patch are retained separately for owner review. Secret scanners have incomplete coverage: unreachable server objects, unfetched refs, Actions artifacts, release assets and remote LFS content were not examined. Absence of a match is not proof of no sensitive data. Existing source license text, including its legacy licensed-work label, is preserved. Publication does not relicense the original source; no additional license grant for the research artifacts is asserted.

This publication edition is released by OMARINO IT Services on 3 November 2026 at https://www.omarino.de/studies-publications . The existing private repository remains private. No upstream commit/PR, scanner production deployment or advisory submission is claimed. The separate payment-module workstream remains outside this public package pending coordinated remediation. The report and accompanying evidence may support a defensive-security evidence submission, but do not establish acceptance into Anthropic's Cyber Verification Program [5].

9. Threats to validity and next experiments

The corpus is deliberately small, source-informed and used to develop the fixes. There is no holdout, population sampling, independent annotation, browser execution, third-party replication or full-product integration test. The severity rubric is local policy. Binary metrics combine heterogeneous claims; technology/redirect detection should not be presented as vulnerability detection. Synthetic fixtures omit CDN behavior, realistic authentication, DNS hierarchies, diverse public CA chains and network variability. Running the extracted original source on .NET 10 may differ from the upstream .NET 8 deployment.

The most useful next experiments are a frozen held-out corpus, browser-backed cookie/header checks, full DNS resolver integration with authenticated failure/absence distinctions, dynamic soft-404 controls, bounded redirect/timeout tests and a clean Docker replication. Owners should review production integration and release licensing before incorporating research patches. The measured contribution is an honest baseline with reproducible failures and repairs in this bounded setting.

Versions & Docs

Versions were verified against official release information on 2 November 2026 before code was written. New backend laboratory: .NET / ASP.NET Core 10.0.14 LTS, SDK 10.0.403 (https://learn.microsoft.com/dotnet/ ; https://learn.microsoft.com/aspnet/core/). Frontend selection if needed: React 19.3.1 (https://react.dev/) and Tailwind CSS 4.3.3 (https://tailwindcss.com/docs). Mobile selection if needed: Flutter 3.47.9 (https://docs.flutter.dev/). No frontend or mobile component was created. Exact release-evidence URLs are in Versions-Docs.md.

References

[1] OWASP. Web Security Testing Guide v4.2, WSTG-SESS-02: Testing for Cookies Attributes. https://wstg.owasp.org/v4.2/4-Web_Application_Security_Testing/06-Session_Management_Testing/02-Testing_for_Cookies_Attributes/

[2] OWASP. Web Security Testing Guide v4.2, WSTG-CONF-04: Review Old Backup and Unreferenced Files for Sensitive Information. https://wstg.owasp.org/v4.2/4-Web_Application_Security_Testing/02-Configuration_and_Deployment_Management_Testing/04-Review_Old_Backup_and_Unreferenced_Files_for_Sensitive_Information/

[3] NIST. SP 800-115: Technical Guide to Information Security Testing and Assessment, September 2008. https://csrc.nist.gov/pubs/sp/800/115/final

[4] Mozilla. HTTP Observatory. https://developer.mozilla.org/en-US/observatory . Referenced methodology context; no Observatory service result is claimed.

[5] Anthropic. Expanding the Cyber Verification Program, 6 October 2026. https://www.anthropic.com/news/cyber-verification-program . Supporting evidence does not guarantee qualification.

[6] Microsoft. .NET 10 downloads, stable runtime/ASP.NET Core 10.0.14, SDK 10.0.403. https://dotnet.microsoft.com/en-us/download/dotnet/10.0

All online references accessed 2 November 2026. Raw local results are the primary evidence for the reported experimental measurements.

Appendix A. Scenario register

ID	Probe	Expected claim	Severity	Fixture claim
TLS-01	tls	True	High	Cleartext URL is rejected
TLS-02	tls	True	High	Untrusted certificate chain is a high finding
TLS-03	tls	True	High	Expired certificate is a high finding
TLS-04	tls	True	High	Hostname mismatch is a high finding
HDR-01	headers	False	Info	All selected controls present
HDR-02	headers	True	High	Three critical headers missing
HDR-03	headers	True	Medium	X-Frame-Options ALLOWALL is ineffective
HDR-04	headers	True	Medium	Whitespace CSP is ineffective
CK-01	cookies	False	Info	HttpOnly SameSite=Lax cookie over lab HTTP
CK-02	cookies	True	Low	SameSite attribute absent
CK-03	cookies	True	Low	Comment is not a SameSite attribute
CK-04	cookies	False	Info	No cookies is informational
DNS-01	dns	False	Info	SPF DMARC enforcement and CAA are present
DNS-02	dns	True	High	Mail policies absent
DNS-03	dns	True	High	DMARC p=none has no enforcement
DNS-04	dns	unknown	Info	Resolver failure must abstain
EXP-01	exposure	False	Info	Sensitive paths return 404
EXP-02	exposure	True	High	Synthetic .env file exposed
EXP-03	exposure	False	Info	robots and security.txt are informational
EXP-04	exposure	False	Info	Catch-all success response is not exposure evidence
EXP-05	exposure	True	High	.env content without Content-Length
EXP-06	exposure	True	High	GET exposes .env even when HEAD is 405
PERF-01	performance	False	Info	Fast 200 has no redirects
PERF-02	performance	True	Info	Two redirects are tracked
PERF-03	performance	True	High	503 availability failure
PERF-04	performance	True	Low	Injected 1200 ms server delay
PERF-05	performance	True	Medium	Six redirects are tracked
TECH-01	technology	True	Info	Generator tag declares WordPress
TECH-02	technology	False	Info	Prose mentioning WordPress is not a stack signal
TECH-03	technology	True	Info	Script asset indicates React
TECH-04	technology	False	Info	Reaction prose is not React
TECH-05	technology	False	Info	Plain HTML contains no stack signals