Jamaica public website security assessment

Framework informed public response review and owner verification plan

StarApple AI
Lead researcher Adrian Dunkley
Evidence snapshot 24 October 2026 America Jamaica

The immediate task is to obtain serving-runtime patch evidence for three PHP observations, then review 21 cookie-attribute concerns and two HTTP resource declarations. No exploitable target vulnerability, successful attack chain or breach was demonstrated. Four chatbot assessment leads remain separate from the 26 website follow-up records.

Evidence measure	Result
Directory entries / primary entries	248 / 242
Public HTML responses / access needs verification	187 / 61
PHP patch verification tasks	3 high follow-up priority
Other follow-up tasks	23 medium follow-up priority
Chatbot assessment leads	4; no LLM architecture attested
Validation kit	27 offline/mocked tests passed; no new target requests in this phase

High follow-up priority is an investigation deadline, not a confirmed High or Critical vulnerability severity. The PHP advisories describe critical vendor flaws; the deployed package state and backports remain unknown. A low public index value is also not proof that an application is secure.

The requested white-box template contains no new code, API contract or server configuration. This report therefore applies the requested methodologies to the available black-box snapshots. Exact vulnerable server file/line locations, site-specific exploit requests and validated source patches cannot be established from these artifacts.

How to use the deliverables

Read sections 1 to 4 in order. The accompanying kit contains every mapped record, exact evidence JSON pointers, conditional configuration examples, a bounded response verifier, offline tests and an owner staging test plan. The existing Excel register retains its identity and its 26 original actions. No agency server, account or code was changed.

1 Attack surface summary

The inventory covers public directory-discovered government domains and government-listed external portals. It is not a certified government asset inventory. Observations are tied to requested URLs, final public responses, timestamps and response hashes. Alias entries remain in coverage but do not create duplicate follow-up actions.

Public source feature	All entries	Primary entries
Valid root HTML responses	187	181
Form declarations	193	186
Entries containing forms	122	117
Forms containing password fields	18	18
Forms containing file inputs	1	1
Third-party script reference occurrences	400	382

These are source declarations per reviewed entry, not unique live endpoints, input controls, vendors or successful submissions. Counts include only successful root HTML responses and exclude linked JavaScript: parsing JavaScript strings as HTML can create phantom forms. No form was submitted.

Visible and unknown trust boundaries

Boundary	Visible evidence	Not established
Visitor to public edge	Ordinary HTTP/HTTPS entry response; public URL/query names and declared forms.	Server parsing, authorization and business rules.
Edge to application runtime	Reported PHP/CMS markers and response flags.	Origin server identity, serving package/SAPI, backports and backend topology.
Application to database or cloud	No internal configuration or data-flow evidence supplied.	SQL/NoSQL sinks, cloud roles, storage permissions, credentials or metadata access.
Browser to third-party services	Script reference occurrences and selected chat integration markers.	Provider-side configuration, LLM enablement, data retention and privilege scope.
Chatbot to retrieved data and tools	Four qualified service/project leads.	Model architecture, authorized retrieval, private-data access and transactional tools.

All API routes, object IDs and client inputs remain untrusted until the server authenticates the caller and authorizes the requested resource and action. This is an architectural principle for owner review, not evidence that any named application currently violates it.

1 Assessment coverage and methodology

The requested approaches are applied together as an evidence and coverage crosswalk. Mapping a test category does not imply that its entire procedure was executed. PTES exploitation/post-exploitation, an OSSTMM STAR/RAV measurement and full authenticated testing were not completed.

Methodology	Application in this report	Remaining boundary
PTES	Inventory, trust boundaries, threat hypotheses, analysis and reporting.	Exploitation and post-exploitation were not performed.
OWASP WSTG v4.2	Version-pinned test IDs for the observed classes.	Only the public-response portions are evidenced.
OWASP API Security 2023	API1 to API10 owner verification categories.	No API-specific vulnerability established by homepage inspection.
MITRE ATTACK Enterprise and Cloud	Conditional behavior mappings and prerequisites.	No adversary technique execution or cloud foothold observed.
NIST SP 800 115	Scope, evidence analysis, mitigation and retest structure.	No certification or exhaustive security assurance.
OSSTMM 3	Explicit tested/untested scope and control observations.	No RAV score, STAR completion or operational-security certification.

Coverage matrix part one

Requested area	Coverage	Evidence and missing prerequisite
Public entry URLs and source parameters	Partial	Existing GET snapshots and public link/form declarations; no exhaustive route crawl.
Client to server trust boundaries	Partial	Browser/public edge and third-party widget candidates are visible. Server authorization and processing are unknown.
Third-party scripts and services	Partial	Source reference occurrences and selected chatbot integration markers only; no provider security certification.
Database cloud and message sinks	Not assessed	No application/data-flow configuration, cloud roles, database clients or private API contracts supplied.
BOLA IDOR and BFLA	Not assessed	Need object/role model, documented API routes and two synthetic accounts with distinct ownership and roles.
JWT password OAuth and session fixation	Not assessed	No token validation implementation, identity-provider configuration, reset flow or authenticated-session test evidence.
Cookie attributes	Partial	Anonymous response flags on selected roots; actual authentication purpose and browser/session behavior unverified.
DOM reflected and stored XSS	Not assessed	No payloads or source-to-sink tracing; unsafe markup or exploitable rendering not established.
CSRF	Not assessed	No state-changing route exercised or token/Origin/SameSite enforcement reviewed.

1 Coverage matrix continued

Requested area	Coverage	Evidence and missing prerequisite
SQL NoSQL injection and prototype pollution	Not assessed	No server query/merge implementation or mutation/injection tests supplied.
SSRF	Not assessed	No outbound-fetch handler, egress policy or controlled callback/metadata request tested.
Race conditions and business logic	Not assessed	No synthetic transactional workflows, concurrency runs, idempotency controls or database constraints reviewed.
Rate limiting and resource consumption	Not assessed	No load, flooding or quota tests; static headers cannot establish effective per-user/API limits.
HTTP headers and CORS	Partial	Selected public response headers only; CORS authenticated origin behavior and effective browser policy were not validated.
Origin TLS and cryptography	Not assessed	HTTPS interception prevents origin-certificate/cipher conclusions; application crypto/key management unavailable.
Error handling	Limited	Only naturally returned responses. Automated diagnostic/source/parser candidates were excluded after review.
Audit logs and incident response	Not assessed	No internal logs, events, access policies, alerting or retention records supplied.
Chatbot prompt injection and tools	Not assessed	Four qualified assessment leads; no LLM architecture attested and no bot prompts or tool requests made.

Inputs required for a true white box or gray box follow up

Owners should provide the authorized repositories and commit identifiers; framework/server versions; API specifications and route inventory; user/role/object ownership model; synthetic test accounts; relevant database/cloud policies; PHP web-SAPI package and backport records; and a staging environment containing disposable data. These inputs determine the exact source patches and safe functional tests. They are not needed to preserve the completed public evidence review.

The scope remains ordinary public-response examination. No input mutation, authentication bypass, form submission, exploit payloads, memory exhaustion, production concurrency/load tests, guessed diagnostics/configuration files or chatbot prompts were executed. Origin TLS expiry and cipher conclusions remain unavailable because HTTPS is intercepted in the review environment.

2 Prioritized vulnerability report

Severity grouping	Current conclusion
Critical confirmed	None established. Vendor-critical PHP issues require owner package confirmation.
High confirmed	None established. Three records carry High follow-up priority.
Medium conditional	21 cookie-attribute records and two HTTP active-resource declarations; actual sensitive-session or browser exposure remains unverified.
Low informational	Reported component/version strings and prior header-hardening observations are inventory/configuration signals; not additional confirmed exploit findings.

High follow up PHP observations

Common proof of observation: an ordinary GET and repeat returned the reported PHP string, which was compared with the vendor release notice. No exploit-triggering request was sent. An affected deployed build and reachable vulnerable path must be established before concluding target vulnerability. Database credentials, cloud hosting and lateral access remain unknown.

PHP vendor fixes published on October 21 are 8.2.35, 8.3.36, 8.4.27 and 8.5.12. Six other observed PHP banners at these levels are not flagged for the two named October flaws. An older string can retain a vendor/source backport. [PHP October notices; PHP end-of-life table]

JA-0034-PHP patch verification Admissions to Caribbean Maritime University (CMU)

Mapping: WSTG-v42-INFO-08, WSTG-v42-CONF-02. Location: https://cmu.edu.jm/. Origin source file and line: unavailable.

Response reports X-Powered-By PHP/7.4.33. The same banner was repeated. HTTP 503; this may describe an error-serving layer.

Exploit proof: not performed. Potential impact is critical only if the serving package lacks applicable fixes and a reachable vulnerable path exists.

Fix and retest: Run the bounded response verifier to observe the banner, then verify actual web SAPI packages, vendor fix/backport and every restarted backend. Do not use an RCE or memory-exhaustion probe on production.

JA-0078-PHP patch verification NWC- iPAY Payment Portal

Mapping: WSTG-v42-INFO-08, WSTG-v42-CONF-02. Location: https://ipay.nwcjamaica.com/. Origin source file and line: unavailable.

Response reports X-Powered-By PHP/8.3.25. The same banner was repeated.

Exploit proof: not performed. Potential impact is critical only if the serving package lacks applicable fixes and a reachable vulnerable path exists.

Fix and retest: Run the bounded response verifier to observe the banner, then verify actual web SAPI packages, vendor fix/backport and every restarted backend. Do not use an RCE or memory-exhaustion probe on production.

JA-0149-PHP patch verification mset.gov.jm

Mapping: WSTG-v42-INFO-08, WSTG-v42-CONF-02. Location: https://www.mset.gov.jm/. Origin source file and line: unavailable.

Response reports X-Powered-By PHP/7.4.33. The same banner was repeated.

Exploit proof: not performed. Potential impact is critical only if the serving package lacks applicable fixes and a reachable vulnerable path exists.

Fix and retest: Run the bounded response verifier to observe the banner, then verify actual web SAPI packages, vendor fix/backport and every restarted backend. Do not use an RCE or memory-exhaustion probe on production.

2 Medium conditional cookie observations

Mapping for every row: WSTG-v42-SESS-02. Vulnerable server source location is unavailable. The exact observed location is the listed public URL and its evidence/SiteID.json#/root/cookies attribute summary in the kit. Cookie names and values were discarded; session-like classes are heuristics.

Record site ID	Exact response URL	First response flags	Repeat
JA-0004	https://www.agrilinkages.gov.jm/	Secure absent 2; HttpOnly absent 1	Match
JA-0007	https://apply.heart-nta.org/	Secure absent 2; HttpOnly absent 0	Match
JA-0021	https://www.caribpesticides.net/index1.asp	Secure absent 1; HttpOnly absent 1	Match
JA-0031	https://clientportal.reb.gov.jm/login	Secure absent 2; HttpOnly absent 1	Match
JA-0052	https://www.elearningja.gov.jm/	Secure absent 1; HttpOnly absent 1	Match
JA-0054	https://employer.nht.gov.jm/login	Secure absent 3; HttpOnly absent 3	Match
JA-0070	https://www.heart-nsta.org/	Secure absent 1; HttpOnly absent 0	No match
JA-0107	https://www.jipo.gov.jm/	Secure absent 1; HttpOnly absent 1	Match
JA-0111	https://jobs.heart-nsta.org/	Secure absent 2; HttpOnly absent 0	Match
JA-0122	https://lmip.heart-nsta.org/Default.aspx	Secure absent 2; HttpOnly absent 0	Match
JA-0123	https://lmip.heart-nsta.org:443/	Secure absent 2; HttpOnly absent 0	Match

2 Cookie observations continued

Record site ID	Exact response URL	First response flags	Repeat
JA-0140	https://www.moca.gov.jm/	Secure absent 0; HttpOnly absent 1	Match
JA-0177	https://www.nwcjamaica.com/	Secure absent 1; HttpOnly absent 1	Match
JA-0182	https://online.nht.gov.jm/Account/Login?ReturnUrl=%2f	Secure absent 2; HttpOnly absent 0	Match
JA-0199	https://portal.jcf.gov.jm/	Secure absent 0; HttpOnly absent 1	Match
JA-0204	https://ptsqueryonline.fsl.org.jm/PTSOnlineWeb/ptsquery.jsp	Secure absent 1; HttpOnly absent 0	Match
JA-0205	https://publications.rada.gov.jm/publicationslist	Secure absent 1; HttpOnly absent 0	Match
JA-0209	https://recruit.moey.gov.jm/	Secure absent 1; HttpOnly absent 0	Match
JA-0224	https://srhadl-portal.srha.gov.jm/	Secure absent 0; HttpOnly absent 1	Match
JA-0228	https://supportjamaica.gov.jm/	Secure absent 0; HttpOnly absent 1	Match
JA-0237	https://trs.fsl.org.jm/Central_System_WEB/updateRegistration.jsp	Secure absent 1; HttpOnly absent 0	Match

Mechanics: an ordinary unauthenticated GET returned the flag patterns; the repeat supports at least one pattern where indicated, not necessarily identity of every cookie. No sensitive cookie theft, replay, XSS or CSRF was demonstrated. If a sensitive cookie is JavaScript-readable, an independently established same-origin XSS flaw could increase impact. Missing Secure has consequences only under additional transport conditions. These are conditional chains, not findings of account compromise.

Remediation: use the PHP reference only where PHP manages the affected cookies, or patch the actual framework cookie setter. Validate the cookie role and appropriate HttpOnly, Secure and SameSite settings in authenticated staging flows. Row identifiers map to the full per-record proof, impact, fix and verification fields in findings.csv and assessment.json.

2 Medium resource observations and AI leads

JA-0110-HTTP asset declarations Jamaica National Heritage Trust

Mapping: WSTG-v42-CRYP-01. Location: https://jnht.com/. The HTML source/template file and line were not supplied.

2 HTTP active-resource declarations in HTTPS page source. Enforcing upgrade directive detected: False. Browser upgrade/blocking was not assessed.

Proof and mechanics: 1. HTTPS page source contained an HTTP active-resource declaration. 2. The declaration recurred on a repeat. 3. No explicit upgrade directive was detected by the bounded source check. 4. Browser network verification is still required; browsers may block the resource. No content substitution was attempted.

Conditional impact: Only if a resource is actually fetched insecurely and an adversary can alter it could page integrity be affected. Browser blocking can instead cause functional failure. No successful injection or credential capture was established.

Fix: Change each identified active-resource URL in the source/template/CMS to its validated HTTPS equivalent. The server-side source file was not supplied. Retest: In owner staging, load the page and inspect the browser network/console. Confirm affected active resources resolve over HTTPS and legitimate features work; source-string replacement alone is not complete validation.

JA-0172-HTTP asset declarations Nutrition Products Limited

Mapping: WSTG-v42-CRYP-01. Location: https://npl.gov.jm/. The HTML source/template file and line were not supplied.

1 HTTP active-resource declarations in HTTPS page source. Enforcing upgrade directive detected: False. Browser upgrade/blocking was not assessed.

Proof and mechanics: 1. HTTPS page source contained an HTTP active-resource declaration. 2. The declaration recurred on a repeat. 3. No explicit upgrade directive was detected by the bounded source check. 4. Browser network verification is still required; browsers may block the resource. No content substitution was attempted.

Conditional impact: Only if a resource is actually fetched insecurely and an adversary can alter it could page integrity be affected. Browser blocking can instead cause functional failure. No successful injection or credential capture was established.

Fix: Change each identified active-resource URL in the source/template/CMS to its validated HTTPS equivalent. The server-side source file was not supplied. Retest: In owner staging, load the page and inspect the browser network/console. Confirm affected active resources resolve over HTTPS and legitimate features work; source-string replacement alone is not complete validation.

2 Chatbot assessment leads

Lead	Evidence	Assessment status
Passport, Immigration and Citizenship Agency	Freshworks loader/fcWidget and Freshchat provider hostname independently observed in homepage source. Widget operation was not exercised.	Chat/bot integration lead; current architecture unverified. LLM architecture and injection vulnerability are unverified.
Accountant General's Department	Tawk_API and live-chat source text independently observed. No active widget or conversation was tested.	Live-chat integration candidate; automation unknown. LLM architecture and injection vulnerability are unverified.
Jamaica Tourist Board	Homepage returned HTTP200; no recognised chat signature was identified in this static source pass. That does not establish absence.	Officially reported AI chatbot; current deployment unverified. LLM architecture and injection vulnerability are unverified.
JamBot project JA-T1234	IDB project record lists Implementation. No public bot endpoint was sought or contacted.	Project architecture/design review lead. LLM architecture and injection vulnerability are unverified.

These four leads are counted separately from the 26 website records. The prompt-injection threat applies only if a model/architecture capable of following untrusted instructions is present. Private data, retrieval permissions and tool access must be established before rating impact. Use the 14 AI owner controls already added to the Excel register. No bot messages or tool requests were sent. [OWASP LLM01 and LLM06]

2 Chained impact hypotheses

The following are illustrative owner-validation scenarios. They are not exploit paths proven on any listed agency. Do not assign a target CVSS score by combining unknown conditions.

Runtime compromise and cloud data

Transition	Required prerequisite	Current evidence
T1190 Exploit Public-Facing Application	Affected serving build and a reachable exploitable application/runtime path.	Reported PHP banners only; package/backport and exploitable path not established.
T1552.005 Cloud Instance Metadata API	Cloud instance hosting and a permitted path to metadata credentials despite controls.	Cloud hosting, metadata reachability and credentials not established.
T1530 Data from Cloud Storage	Usable permissions and access to the specific storage objects.	No cloud role, storage access or collection established.

Breakpoints include current vendor patches, least privilege, isolation between application and cloud management paths, constrained instance roles and resource-level storage permissions. The official Cloud Matrix is a platform view within MITRE ATTACK Enterprise; it is not evidence of cloud deployment. [MITRE T1190, T1552.005 and T1530]

Cookie exposure with a separate script flaw

A missing HttpOnly flag can matter if the cookie is sensitive and independent script execution in the same origin is established. The public check establishes neither the script flaw nor the cookie role. Server-side authorization, context-appropriate output encoding and session-cookie hardening address separate prerequisites. A generic CORS wildcard on public content would not, by itself, prove a sensitive-data chain.

Prompt injection with privileged tools

If a chatbot uses an LLM, ingests untrusted pages/documents and can reach private data or consequential tools, manipulated instructions could cross a trust boundary. The model must not authorize database or API operations. Enforce identity, ownership, action scope and approval in trusted downstream code. For these four leads, model type, tools and permissions remain unverified. [OWASP LLM01 and LLM06]

3 Remediation code and headers

The code below is a reference for owner integration. There is no supplied agency source tree to refactor and no verified common web server across the inventory. Apply only after identifying the actual runtime/framework, staging the change and checking legitimate flows. Each of the 26 records has a remediation_reference and a verification_test in assessment.json.

PHP runtime patch and closure workflow

For the three PHP records, confirm the package identifier, vendor security advisory or source backport and effective web SAPI. Apply the vendor-fixed package or migrate an unsupported branch. Restart every relevant worker/backend. Retain deployment evidence and confirm legitimate flows. CLI php -v and a response banner can describe different runtimes; a hidden banner is not closure.

PHP HTTPS session and production error reference

```
; kit/php/production-hardening.ini.example
; This does not patch the PHP runtime.
expose_php = Off
display_errors = Off
display_startup_errors = Off
log_errors = On
session.use_only_cookies = 1
session.use_strict_mode = 1
session.cookie_secure = 1
session.cookie_httponly = 1
session.cookie_samesite = Lax

; Conditional October 23 advisory; validate application prerequisites.
session.upload_progress.enabled = 0
```

SameSite=Lax is an example, not a universal setting. Cross-site identity and payment flows may require another deliberate policy; SameSite=None requires Secure. Application/framework overrides and FPM pool settings must be checked. Use HttpOnly for sensitive session cookies that do not require JavaScript access, rather than applying it blindly to every cookie.

The October 23 session notice requires an unusual serializer configuration/runtime switch with upload progress enabled. The notice and owner configuration determine applicability. Disabling upload progress is separate from applying the October 21 critical runtime fixes. No production setting was changed. [PHP October notices and PHP secure session guidance]

3 Web server and resource remediation

Nginx reference for applicable HTTPS applications

```
# Only if Nginx returns these responses. Test actual location inheritance.
add_header X-Content-Type-Options "nosniff" always;
add_header Referrer-Policy "strict-origin-when-cross-origin" always;

# Only if third-party framing is not a legitimate requirement.
add_header X-Frame-Options "SAMEORIGIN" always;
add_header Content-Security-Policy "frame-ancestors 'self'" always;

# Enable after durable HTTPS and host readiness are confirmed.
# Start short and raise deliberately after validation.
# add_header Strict-Transport-Security "max-age=300" always;
```

The complete example in kit/nginx/security-headers.conf.example includes a trial Report-Only policy. Report-Only does not enforce protection or send telemetry without a reporting endpoint. Build the real source allowlist from required resources and validate forms, scripts, identity flows and widgets before enforcement. Do not add includeSubDomains or preload until domain readiness is established. [OWASP Secure Headers]

Two HTTP active resource records

Locate the source/template/CMS field generating each HTTP declaration, confirm that the resource supports HTTPS and replace the declaration with the validated HTTPS URL. Inspect the staging browser network panel and console for actual delivery or blocking, and verify the feature. A server header change or source replacement alone does not prove that browser exposure has been fixed.

CORS and API boundaries

Same-origin services generally need no CORS permission. If a documented API needs cross-origin browser access, allow only approved origins and methods, set credential behavior deliberately and maintain independent authentication/authorization on every request. Do not treat CORS as authorization. No current API CORS exploit is established by the supplied evidence.

Application source patches awaiting code

BOLA/BFLA needs resource-ownership and role checks in the real route handlers; SQL/NoSQL needs parameterized values and validated structures; JWT/OAuth needs the actual verifier/provider configuration; SSRF needs the outbound-fetch and egress implementation; mass assignment needs explicit input fields; concurrency needs authoritative transactions/constraints/idempotency. Without those artifacts, presenting invented filenames, line numbers or framework-specific middleware as a patched agency implementation would be misleading.

4 Verification tests

Executed local verification

Fourteen supplied unit tests and 13 independent mocked tests passed. The tests cover patch-floor boundaries, nonstable/vendor version strings, cookie-value suppression, missing flag observations, report-only and malformed framing primitives, exact URL scope, local/private destination rejection, DNS preflight behavior and a single mocked GET that does not read a body or follow redirects. These tests validate the reference verifier, not any agency remediation.

```
# From the kit directory. These commands do not contact government sites.
python -m unittest discover -s verification -p 'test_verifier*.py' -v
python verification/verify_public_response.py \
  --evidence-json evidence/JA-0078.json
```

Optional owner run bounded response check

```
# One explicitly allowlisted HTTPS URL; no redirects or chat messages.
python verification/verify_public_response.py \
  --url https://ipay.nwcjamaica.com/ \
  --approved-urls verification/approved_urls.json
```

The script emits selected observations only. It discards cookie values and does not inspect the response body. PHP package patch status always remains an owner-verification requirement. Its framing field reports tokens only and cannot establish browser policy enforcement. DNS preflight is not DNS pinning/rebinding protection and does not attest a proxy route. This is a bounded owner-operated checker, not an SSRF security boundary.

Equivalent curl capture without cookie output

```
curl --silent --show-error --proto '=https' \
  --connect-timeout 5 --max-time 15 --max-redirs 0 \
  --dump-header - --output /dev/null \
  'https://ipay.nwcjamaica.com/' | \
awk 'tolower($0) ~ /^(http\/|x-powered-by:|strict-transport-security:)/ {print}'
```

Expected outcome: the response observation can be compared with the dated snapshot. The owner must separately confirm the serving package/backport and deployment. A 200 response or a fixed-level/absent banner is insufficient to mark the site secure. The report does not contain a validated exploit payload for an unproven flaw.

4 Owner staging verification plan

The following tests are planned and unexecuted. Use the actual route/schema/role contracts and disposable test records in an owner-controlled environment. The expected assertions describe the required control without inventing a target API path or production payload.

Test	Expected verification	Required artifact
Object ownership	Given synthetic users A and B with different objects, when A accesses B objects through every documented method, expect denial and no data or state change.	Staging accounts, route contract, authorization code
Function authorizati on	Given ordinary and administrative roles, exercise documented functions under each role and anonymous state; expect policy-consistent denial.	Role matrix and isolated accounts
JWT validation	Validate rejection of expired, wrong-issuer, wrong-audience, disallowed-algorithm and untrusted-key fixtures locally; do not accept unsigned fallback.	Actual verifier and test keys
Password reset and OAuth	Verify single-use/expiry, account binding, safe redirect allowlists, state and PKCE where applicable; confirm no account enumeration or session confusion.	Source and identity provider test tenant
Session lifecycle	Use synthetic logins to confirm rotation on authentication/privilege change, invalidation on logout and reset, and cookie attributes across all paths.	Owner staging and synthetic accounts
CSRF	A missing/invalid token or disallowed Origin must fail for documented state-changing requests; legitimate application and federated flows must remain functional.	State-changing routes and trusted-origin policy
XSS	Trace untrusted input through templates/DOM sinks; encode for its context and verify harmless marker text remains data in isolated test pages.	Client/server code and sandbox renderer
SQL and NoSQL	Use the actual repository to verify parameterized values, allowlisted fields/operators and rejection of unexpected structural input.	Query code and disposable database
SSRF	Use a controlled local stub to prove scheme/host/port and redirect/egress policies block disallowed destinations without contacting real metadata endpoints.	Outbound handler and isolated network

4 Owner staging tests continued

Test	Expected verification	Required artifact
Prototype pollution	Verify object merge/path utilities reject reserved prototype-related keys and that untrusted input cannot affect unrelated objects in local tests.	Dependency/source inventory and local fixtures
Mass assignment	Send only synthetic documented fields plus disallowed role/state properties in staging; expect rejection/ignoring without privilege change.	Request schema and role model
Concurrency	Submit a bounded pair of synthetic duplicate actions in staging; verify transaction/constraint/idempotency logic allows only the intended transition.	Owner-approved staging workflow and disposable records
Price and state logic	Calculate authoritative amounts/state transitions on server from trusted records; synthetic client edits must not determine final price/privilege/state.	Business rules and synthetic transactions
Rate limiting	Use a bounded owner-approved test budget; verify per-user/tenant and resource controls without production flooding.	Agreed limits and isolated environment
CORS	Using a controlled test origin, verify only approved origins can read intended responses, credential policy is deliberate, and server authorization still applies.	API route list and non-production origin
Prompt injection	Use owner-scoped synthetic conversations/documents in staging to validate data/tool boundaries and downstream denial, without real private data or public-bot probing.	Confirmed model architecture, tools and isolated accounts
Audit evidence	Verify denied access, privileged actions and approvals generate useful redacted events; restrict log access and test retention.	Log schema, storage access and incident workflow

Closure evidence and acceptance

For a PHP record, retain the web-SAPI package/backport evidence, backend inventory, restart/deployment time, legitimate-flow checks and responsible owner. For a cookie record, identify the real sensitive cookie and show the intended flags after authentication and renewal without retaining its value. For a resource record, retain staging browser evidence that required active content is delivered over HTTPS and works. For AI leads, record model type, source trust, permissions and downstream authorization before moving from an assessment lead to a severity-rated finding.

A retest should close the specific demonstrated condition, preserve useful audit evidence and record residual uncertainty. No record should be closed solely because a component banner was removed. The 61 access/response limitations remain a verification backlog rather than confirmed public outages.

Framework crosswalk and evidence locations

Observation	Verified mapping	Execution boundary
PHP version reported in X-Powered-By; framework/CMS markers in HTML and cookies	WSTG-v42-INFO-08	Partial category coverage: public banner and source observation only, where evidence exists.
Web server type/version in Server response header	WSTG-v42-INFO-02	Partial category coverage: ordinary response banner only.
PHP production configuration and diagnostic exposure concerns	WSTG-v42-CONF-02	Limited public observations and owner checklist; server configuration review not completed.
Information in public HTML comments, metadata or linked client source	WSTG-v42-INFO-05	Partial source inspection only for retrieved resources.
PHP warnings, errors, paths or traces already present on normal responses	WSTG-v42-ERRH-01	Passive inspection of naturally returned errors only; full error-handling testing not completed.
Set-Cookie Secure, HttpOnly and SameSite attributes	WSTG-v42-SESS-02	Partial category coverage: attributes of cookies issued on reviewed unauthenticated responses.
HSTS response header and directives	WSTG-v42-CONF-07	Observed-header subset only where successful HTTPS responses exist.
HTTP active-resource references inside HTTPS pages; HTTP-to-HTTPS entry behavior	WSTG-v42-CRYP-01	Partial application-use subset: source references and ordinary entry requests; origin TLS and cipher assessment not completed.
Forms or sensitive-data paths pointing to unencrypted HTTP	WSTG-v42-CRYP-03	Form-action/source inspection only where present; no data transmission test.
Missing or permissive X-Frame-Options / CSP frame-ancestors	WSTG-v42-CLNT-09	Header-only subset; clickjacking exploit testing not completed.
General CSP, nosniff, Referrer-Policy and browser-header configuration	OWASP Secure Headers	Header observations; applicability-specific hardening review.

General CSP, nosniff and referrer-policy review is anchored to OWASP Secure Headers. WSTG v4.2 does not provide the newer CONF-12 label in its verified configuration index. API8:2023 and API9:2023 remain conditional owner/inventory categories where an API context exists. No API vulnerability is established merely by a homepage response.

Every record in the kit includes its exact public route, framework mapping, evidence JSON pointers, proof of observation, exploit-proof status, conditional impact, remediation file reference and retest criteria. These artifact pointers are evidence locations; they are not invented vulnerable agency source-code locations.

Source register

The following primary/vendor sources support definitions and interpretation. Risk priorities, staging acceptance criteria and effort estimates are researcher recommendations. Vendor advisory applicability is not equivalent to proof of a vulnerable installed package. URLs were checked during this review; access to official PTES documentation used its original repository after the standalone site timed out.

1. Fingerprint Web Application Framework
https://wstg.owasp.org/v4.2/4-Web_Application_Security_Testing/01-Information_Gathering/08-Fingerprint_Web_Application_Framework/

2. Fingerprint Web Server
https://wstg.owasp.org/v4.2/4-Web_Application_Security_Testing/01-Information_Gathering/02-Fingerprint_Web_Server/

3. Test Application Platform Configuration
https://wstg.owasp.org/v4.2/4-Web_Application_Security_Testing/02-Configuration_and_Deployment_Management_Testing/02-Test_Application_Platform_Configuration/

4. Review Webpage Content for Information Leakage
https://wstg.owasp.org/v4.2/4-Web_Application_Security_Testing/01-Information_Gathering/05-Review_Webpage_Content_for_Information_Leakage/

5. Testing for Improper Error Handling
https://wstg.owasp.org/v4.2/4-Web_Application_Security_Testing/08-Testing_for_Error_Handling/01-Testing_For_Improper_Error_Handling/

6. Testing for Cookies Attributes
https://wstg.owasp.org/v4.2/4-Web_Application_Security_Testing/06-Session_Management_Testing/02-Testing_for_Cookies_Attributes/

7. Test HTTP Strict Transport Security
https://wstg.owasp.org/v4.2/4-Web_Application_Security_Testing/02-Configuration_and_Deployment_Management_Testing/07-Test_HTTP_Strict_Transport_Security/

8. Testing for Weak Transport Layer Security
https://wstg.owasp.org/v4.2/4-Web_Application_Security_Testing/09-Testing_for_Weak_Cryptography/01-Testing_for_Weak_Transport_Layer_Security/

9. Testing for Sensitive Information Sent via Unencrypted Channels
https://wstg.owasp.org/v4.2/4-Web_Application_Security_Testing/09-Testing_for_Weak_Cryptography/03-Testing_for_Sensitive_Information_Sent_via_Unencrypted_Channels/

10. Testing for Clickjacking
https://wstg.owasp.org/v4.2/4-Web_Application_Security_Testing/11-Client-side_Testing/09-Testing_for_Clickjacking/

11. OWASP Secure Headers Project; supplementary guidance
https://secure-headers-project.owasp.org/response-headers/

12. OWASP API Security Top 10 2023
https://owasp.org/API-Security/editions/2023/en/0x11-t10/

13. PTES original documentation
https://github.com/pentest-standard/docs/blob/master/source/Main_Page

14. NIST SP 800 115
https://csrc.nist.gov/pubs/sp/800/115/final

15. OSSTMM 3
https://www.isecom.org/OSSTMM.3.pdf

16. MITRE ATTACK Enterprise Cloud Matrix
https://attack.mitre.org/matrices/enterprise/cloud/

17. Exploit Public-Facing Application
https://attack.mitre.org/techniques/T1190/

18. Unsecured Credentials: Cloud Instance Metadata API
https://attack.mitre.org/techniques/T1552/005/

19. Data from Cloud Storage
https://attack.mitre.org/techniques/T1530/

20. PHP October 21 and 23 security notices
https://www.php.net/archive/2026.php

21. PHP advisory GHSA-w8hw-w9fx-j79w
https://github.com/php/php-src/security/advisories/GHSA-w8hw-w9fx-j79w

22. PHP advisory GHSA-6wmf-4rv5-p3rp
https://github.com/php/php-src/security/advisories/GHSA-6wmf-4rv5-p3rp

23. PHP unsupported branches
https://www.php.net/eol.php

24. PHP supported versions
https://www.php.net/supported-versions.php

25. PHP secure session settings
https://www.php.net/manual/en/session.security.ini.php

26. PHP error configuration
https://www.php.net/manual/en/errorfunc.configuration.php

27. MDN secure cookie configuration
https://developer.mozilla.org/en-US/docs/Web/Security/Practical_implementation_guides/Cookies

28. MDN mixed content
https://developer.mozilla.org/en-US/docs/Web/Security/Mixed_content

29. OWASP HTTP Headers Cheat Sheet
https://cheatsheetseries.owasp.org/cheatsheets/HTTP_Headers_Cheat_Sheet.html

30. OWASP LLM01:2025 Prompt Injection
https://genai.owasp.org/llmrisk/llm01-prompt-injection/

31. OWASP LLM06:2025 Excessive Agency
https://genai.owasp.org/llmrisk/llm062025-excessive-agency/

32. OWASP LLM02
https://genai.owasp.org/llmrisk/llm022025-sensitive-information-disclosure/

33. OWASP LLM07
https://genai.owasp.org/llmrisk/llm072025-system-prompt-leakage/

34. OWASP LLM08
https://genai.owasp.org/llmrisk/llm082025-vector-and-embedding-weaknesses/

35. OWASP LOG
https://cheatsheetseries.owasp.org/cheatsheets/Logging_Cheat_Sheet.html