AMERICAN DREAM
CYBER SECURITY



CONFIDENTIAL

Penetration Test Report

Web Application, Cloud & API Security Assessment

Sample report — demonstrates the structure, depth and quality of an American Dream Cyber Security engagement.

Client	NovaPay Financial Technologies (fictitious)
Prepared by	American Dream Cyber Security Labs Inc.
Engagement	Grey-Box Penetration Test — Web / Cloud / API
Report version	1.0 (Final)
Report date	29 October 2026
Classification	CONFIDENTIAL — Restricted Distribution
Contact	info@americandream-cybersecurity.com

This document contains confidential information about the security posture of the client named above and is intended solely for authorised recipients. This is a SAMPLE report: all targets, hostnames, accounts and data are fictitious.

Confidentiality & Legal Notice

This report is the product of a security assessment performed by American Dream Cyber Security Labs Inc. ("American Dream", "the Consultant") for the client. It is provided on a strictly confidential basis and is protected by the terms of the engagement agreement and any applicable non-disclosure agreement.

- The findings describe the security posture of the in-scope systems **as observed during the testing window only**. Security is dynamic; later changes to the environment may introduce risks not reflected here.
- Penetration testing is a point-in-time, best-effort activity. The absence of a finding does not guarantee that no vulnerability exists; testing is bounded by the agreed scope, time and rules of engagement.
- All testing was performed with the prior written authorisation of the client. Proof-of-concept activity was deliberately constrained to demonstrate risk without causing damage or accessing data beyond what was approved.
- This document, including its proof-of-concept material, is shared for the client's remediation purposes and must not be used against any system without explicit authorisation.

Purpose of this Document

This sample report shows the format, depth and quality of the deliverable a client receives at the end of a combined Web / Cloud / API penetration test. It follows a fictitious engagement for a fictional fintech, "NovaPay", with findings at every severity level (Critical, High, Medium, Low and Informational). Each finding has a description, business impact, proof of concept (PoC) and remediation guidance.

Note. All company names, domains, IP ranges, credentials, identifiers and data in this report are invented for demonstration purposes. The proof-of-concept sections show methodology and evidence, not working exploit code.

Table of Contents

Confidentiality & Legal Notice 2
Purpose of this Document 2

1 About American Dream Cyber Security 4
1.1 Who We Are 4
1.2 Our Services 4

2 Engagement Scope & Approach 5
2.1 Scope of Work 5
2.2 Rules of Engagement 5
2.3 Methodology 5
2.4 Risk Rating Methodology 6

3 Executive Summary 7
3.1 Findings at a Glance 7
3.2 Summary of Findings 8
3.3 Strategic Recommendations 8

4 Detailed Findings 9
4.1 WEB-01 — SQL Injection in Unauthenticated Merchant Search (Critical) 9
4.2 CLD-01 — Over-Privileged IAM Role Across Tenant Boundary (Critical) 11
4.3 API-01 — Broken Object-Level Authorization (BOLA / IDOR) (High) 12
4.4 API-02 — Excessive Data Exposure in Auth Response (High) 13
4.5 WEB-02 — Stored Cross-Site Scripting in Display Name (High) 14
4.6 CLD-02 — Publicly Writable S3 Bucket (High) 15
4.7 API-03 — Missing Rate Limiting on Auth / OTP (Medium) 16
4.8 API-04 — SSRF via Webhook URL Validation Bypass (Medium) 17
4.9 WEB-03 — Session Token Not Rotated on Privilege Change (Medium) 18
4.10 WEB-04 — Verbose Errors Disclose Stack Traces (Medium) 19
4.11 WEB-05 — Missing Security Headers (Low) 20
4.12 WEB-06 — User / E-mail Enumeration (Low) 21
4.13 CLD-03 — CloudTrail Logging Disabled in a Region (Low) 22
4.14 INF-01 — TLS Supports a Legacy Cipher Suite (Informational) 23
4.15 INF-02 — No API Versioning / Deprecation Policy (Informational) 24
4.16 INF-03 — Cloud Resources Missing Consistent Tagging (Informational) 25

5 Remediation Roadmap 26
5.1 Retesting 26

6 Appendix 27
A. Severity & CVSS Reference 27
B. Tooling 27
C. Document Control 27

Contact American Dream Cyber Security 28

1 About American Dream Cyber Security

NEXTGEN CYBER SECURITY LABS.

1.1 Who We Are

American Dream Cyber Security Labs Inc. combines a powerful proprietary security platform with elite cyber services: real-time global threat mitigation, penetration testing and next-generation security scanners. We protect businesses of every size. **We find your security weaknesses and fix them for you.**

Our penetration testing practice delivers black, grey and white-box assessments of web applications, APIs, cloud infrastructure, networks, mobile, AI systems and IoT. Every engagement pairs rigorous manual testing with the American Dream Cyber Scanner and ends with developer-ready remediation guidance and complimentary retesting.

Assessments follow recognised standards (OWASP WSTG, OWASP API Security Top 10, PTES, MITRE ATT&CK) and are rated with CVSS 3.1, so results can be benchmarked and tracked over time.



<2h
INITIAL RESPONSE SLA

<24h
FIRST FINDINGS REPORTED

<48h
RAPID PROJECT KICK-OFF

1.2 Our Services

PENTEST
Penetration Testing
Black, grey and white-box testing of web, AI, mobile, cloud and IoT systems.
Black/White/Gray Box · Mobile · Code Review · AI · Network · API

XDR
Cyber Defense
24/7 continuous monitoring and automated threat mitigation for networks and endpoints.
Incident Response · 24/7 Monitoring · Endpoint · WAF Mitigation

FORENSICS
Computer Forensics & Malware Analysis
Cyber investigations, reverse engineering and evidence recovery for APTs and zero-day malware.
Reverse Engineering · Malware Analysis · Digital Forensics

R&D
Cyber Research
In-depth vulnerability research and intelligence on emerging threat vectors.
Red & Blue Teaming · Security Trainings · Vulnerability Research

OUR UNIQUE PRODUCT
American Dream Cyber Scanner
AI-powered scanning plus elite human review across code, web, network, cloud and API.
PTaaS · AI-Driven · API Security · SAST & DAST

OTHER
Custom Engineering
Tailored cyber security solutions engineered for your specific architecture.
Custom Architectures · Security Engineering · Deep Tech

2 Engagement Scope & Approach

2.1 Scope of Work

The assessment covered three connected environments that together make up the NovaPay payment platform. Testing was conducted from the internet and, for cloud, with read-only audit credentials during the agreed window.

Assessment area	In-scope targets	Testing style
Web Application	app.novapay.example — customer & merchant portal (authenticated + unauthenticated)	Grey-box
API	api.novapay.example — REST API v1 & v2 (OpenAPI spec provided)	Grey-box
Cloud (AWS)	Production AWS account 4411-xxxx-8820 — IAM, S3, EC2, RDS, CloudTrail	Configuration review (read-only)

Out of scope

- Denial-of-service and volumetric load testing.
- Physical security and social engineering of staff.
- Third-party and payment-processor systems not owned by the client.
- Source-code review beyond the files provided for grey-box context.

2.2 Rules of Engagement

- Testing was limited to the targets above during the authorised window.
- Destructive actions were prohibited; proof-of-concept activity was kept to the minimum needed to demonstrate impact.
- Critical findings were reported to the client immediately instead of waiting for the final report.
- Data access was minimised; where sensitive data could be reached, impact was proven against masked or purpose-created test records only.

2.3 Methodology

The engagement followed a structured, standards-aligned methodology. Automated scanning with the American Dream Cyber Scanner provided breadth, and manual testing provided depth. Every result was manually verified to remove false positives.

Phase	Activities	Primary standards
1. Reconnaissance	Attack-surface mapping, technology fingerprinting, OpenAPI and cloud inventory review	OSINT, PTES
2. Threat modelling	Trust-boundary and data-flow analysis, attacker profiling (STRIDE, attack trees)	STRIDE
3. Vulnerability analysis	Automated and manual testing of web, API and cloud configuration	OWASP WSTG, OWASP API Top 10
4. Exploitation	Controlled, ROE-bounded validation of real impact	PTES, MITRE ATT&CK
5. Post-exploitation	Assessment of reachable data and lateral movement (non-destructive)	MITRE ATT&CK

Phase	Activities	Primary standards
6. Reporting	Risk rating (CVSS 3.1), remediation guidance, retest support	CVSS 3.1

2.4 Risk Rating Methodology

Each finding is rated with the Common Vulnerability Scoring System (CVSS) v3.1. The base score (0.0–10.0) maps to the qualitative severity used throughout this report. Severity reflects technical risk; business context agreed with the client may adjust remediation priority.

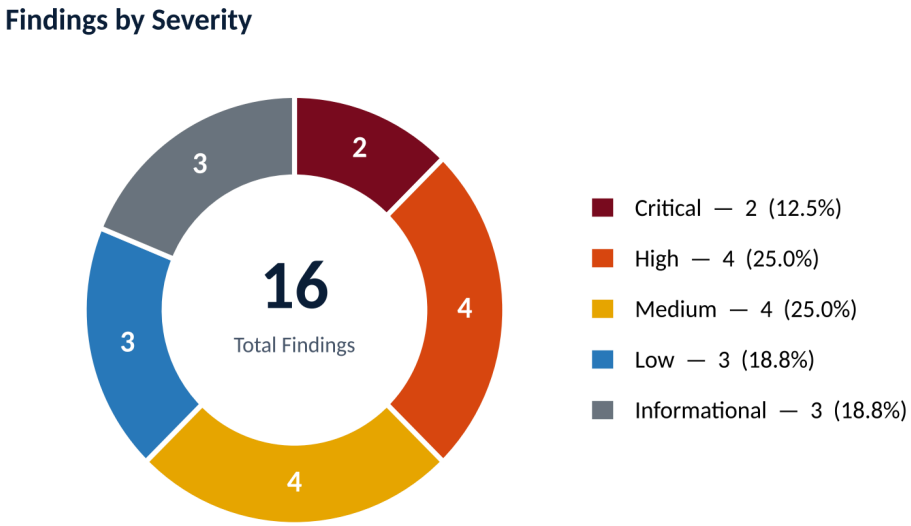
Severity	CVSS 3.1 base	Expected response
Critical	**9.0 – 10.0**	Emergency fix: remediate immediately
High	**7.0 – 8.9**	Remediate within days / the current sprint
Medium	**4.0 – 6.9**	Plan remediation in the near term
Low	**0.1 – 3.9**	Fix as part of routine maintenance
Informational	**0.0**	Best-practice hardening; no direct risk

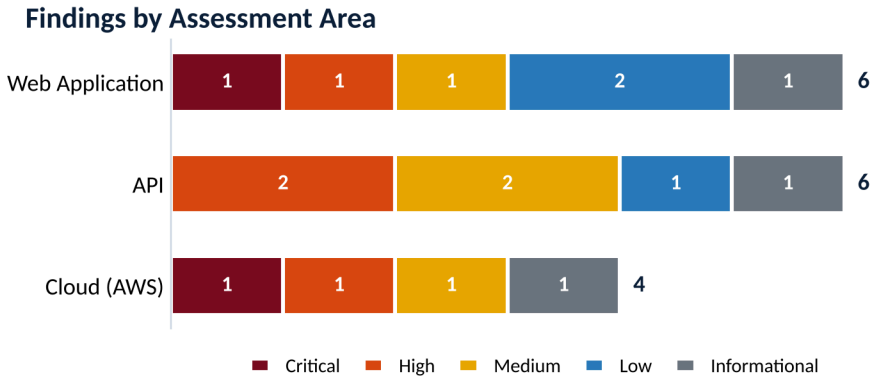
3 Executive Summary

American Dream Cyber Security performed a grey-box penetration test of NovaPay's web application, API and AWS cloud environment. The assessment identified **16 findings**, including **2 Critical** and **4 High** issues that need prompt attention. The most serious would let an unauthenticated attacker read sensitive customer data (SQL injection) and could give full administrative control of the production cloud account (an over-privileged, broadly trusted IAM role).

The platform has a solid baseline in places, including modern TLS, MFA and a largely parameterised data layer. A small number of high-impact gaps in input handling, authorization and cloud privilege management raise the organisation's risk significantly. None of the Critical or High findings is difficult to fix; each has a clear, well-understood remediation in Sections 4 and 5.

3.1 Findings at a Glance





2	4	4	3	3
Critical	High	Medium	Low	Informational

3.2 Summary of Findings

Every finding with its identifier, severity, CVSS base score and affected area. Full detail starts in Section 4.

ID	Severity	CVSS	Area	Finding
WEB-01	Critical	9.8	Web Application	SQL Injection in Unauthenticated Merchant Search
CLD-01	Critical	9.6	Cloud (AWS)	Over-Privileged IAM Role Assumable Across Tenant Boundary
API-01	High	8.1	API	Broken Object-Level Authorization (BOLA / IDOR) on Invoice API
API-02	High	7.5	API	Excessive Data Exposure in Authentication Response
WEB-02	High	8.7	Web Application	Stored Cross-Site Scripting in Merchant Display Name
CLD-02	High	7.4	Cloud (AWS)	Publicly Writable S3 Bucket Serving Application Assets
API-03	Medium	6.5	API	Missing Rate Limiting on Authentication and OTP Endpoints
API-04	Medium	6.5	API	Server-Side Request Forgery via Webhook URL Validation Bypass
WEB-03	Medium	5.9	Web Application	Session Token Does Not Rotate After Privilege Change
WEB-04	Medium	5.3	Web Application	Verbose Error Messages Disclose Stack Traces and Framework Versions
WEB-05	Low	3.7	Web Application	Missing Security Headers (HSTS, CSP, X-Content-Type-Options)
WEB-06	Low	4.3	Web Application	User/E-mail Enumeration via Distinct Responses
CLD-03	Low	3.1	Cloud (AWS)	CloudTrail Logging Disabled in a Secondary Region
INF-01	Informational	0.0	Web Application	TLS Configuration Supports a Legacy Cipher Suite
INF-02	Informational	0.0	API	API Lacks a Documented Versioning & Deprecation Policy
INF-03	Informational	0.0	Cloud (AWS)	Cloud Resources Missing Consistent Tagging

3.3 Strategic Recommendations

- Treat the two Critical findings as emergencies: parameterise the vulnerable query (WEB-01) and apply least privilege to the production IAM role and its trust policy (CLD-01).
- Introduce centralised, enforced authorization checks in the API so object-level access control cannot be skipped per endpoint (API-01).
- Adopt secure-by-default output encoding and allow-listed response serialisation to remove the XSS and data-exposure classes (WEB-02, API-02).
- Harden the cloud baseline: block public access to storage, enforce multi-region logging and make least-privilege IAM the standard (CLD-02, CLD-03).
- Add security regression tests to CI/CD so fixed issues cannot silently return.

4 Detailed Findings

Each finding includes a metadata summary, a description of the issue and its root cause, the business impact, a proof of concept demonstrating the issue, and prioritised remediation guidance. Findings are ordered from most to least severe.

4.1 WEB-01 — SQL Injection in Unauthenticated Merchant Search

Finding ID	**WEB-01**	**Severity**	**Critical (CVSS 9.8)**
Category	OWASP A03:2021 – Injection	**CWE**	CWE-89
Affected area	Web Application	**CVSS vector**	CVSS:3.1/AV:N/AC:L/PR:N/UI:N/S:U/C:H/I:H/A:H
Affected asset	`app.novapay.example – GET /api/public/merchants/search`		

Description

The public merchant-directory search accepts a `q` parameter that is incorporated into a backend PostgreSQL query without parameterisation. The endpoint requires no authentication. Our testing confirmed the input reaches the database as code rather than data, allowing the query logic to be altered.

The gray-box code review confirmed the root cause: this one endpoint builds its query by string concatenation, while the rest of the codebase uses the parameterised ORM. The vulnerability is therefore isolated and straightforward to remediate.

Business Impact

An anonymous attacker on the internet can influence the database query and retrieve data the application account can access — which in the test environment included customer contact records and password hashes — and, because that account also has write access, potentially alter stored records. This represents a direct threat to customer-data confidentiality and transaction integrity. In accordance with the Rules of Engagement, we confirmed read access against a small number of masked test rows and did not extract production data or attempt any modification.

Proof of Concept

PROOF OF CONCEPT *evidence summary — sensitive values redacted*

1. Confirm the parameter is injectable

```
// A single-quote probe in `q` changed the server response from a normal
// result set to a database error, indicating the input breaks out of the query.
GET /api/public/merchants/search?q=nova%27  HTTP/1.1
Host: app.novapay.example
✓ HTTP/1.1 500 – unhandled database exception (syntax error) returned to client
```

2. Confirm attacker control of query logic (boolean test)

```
// Two logically opposite conditions produced two different, consistent result
// counts – a true condition returned the full listing, a false one returned none.
// This proves the query evaluates attacker-supplied logic.
q = <search term plus an always-true condition>   -> full result set
q = <search term plus an always-false condition>  -> empty result set
✓ Differential responses confirm injection without needing to extract any data
```

3. Demonstrate data exposure (ROE-limited)

```
// With client approval, a read-only query was used to retrieve the database
// version banner and a COUNT() of one non-sensitive reference table only.
✓ Postgres 14.x banner returned in the response body
✓ Row-count of public `countries` table returned (no customer data accessed)
```

Remediation

- Rewrite the endpoint to use parameterised queries / prepared statements (bound parameters) exactly as the rest of the application does; never concatenate user input into SQL.
- Run the application under a least-privilege database role that is read-only for directory data and cannot reach customer or credential tables.
- Add a server-side allow-list on the q parameter (length and character class) as defence in depth.
- Deploy a WAF rule to alert on injection patterns, and add a regression test that asserts this endpoint rejects injection probes.

References

OWASP Top 10 2021 – A03 Injection • OWASP SQL Injection Prevention Cheat Sheet • CWE-89

4.2 CLD-01 — Over-Privileged IAM Role Assumable Across Tenant Boundary

Finding ID	CLD-01	Severity	Critical (CVSS 9.6)
Category	OWASP Cloud / IAM Misconfiguration	CWE	CWE-269
Affected area	Cloud (AWS)	CVSS vector	CVSS:3.1/AV:N/AC:L/PR:L/UI:N/S:C/C:H/I:H/A:N
Affected asset	AWS account 4411-xxxx-8820 — role NovaPay-App-Exec		

Description

The execution role attached to the primary application compute carries an inline policy granting `*:*` (all actions on all resources). Separately, the role's trust policy allows `sts:AssumeRole` from any principal within the organisation rather than from a specific, pinned service principal.

The combination means a foothold in any lower-trust workload in the organisation can assume a role that effectively administers the entire production account — a privilege-escalation and tenant-isolation failure (scope change from the compromised component to the whole account).

Business Impact

A low-privileged presence in any workload sharing the organisation can escalate to full administrative control of the production AWS account: reading every S3 bucket and secret, creating backdoor identities, and disabling logging. This is a crossing of the trust boundary between a single workload and the entire cloud estate.

Proof of Concept

PROOF OF CONCEPT *evidence summary — sensitive values redacted*

1. Enumerate the role policy

```
// Read-only IAM review (client-provided audit credentials) showed the inline policy:
✓ Effect: Allow   Action: "*"   Resource: "*"
// i.e. unrestricted administrative permissions attached to app compute
```

2. Inspect the trust relationship

```
// The role trust policy permitted assumption from a broad org-wide principal
// rather than a single pinned service role.
✓ Principal scope: entire organisation  (expected: one specific service principal)
```

3. Confirm reachability (non-destructive)

```
// From a deliberately low-privileged test identity, an sts:AssumeRole call to
// NovaPay-App-Exec succeeded and returned temporary admin-level credentials.
✓ AssumeRole succeeded — session issued; no further action taken per ROE
```

Remediation

- Replace the `*:*` policy with least-privilege permissions scoped to only the services, actions and resource ARNs the workload actually needs.
- Pin the trust policy to the specific service principal(s) that must assume the role; add `aws:SourceArn`/`aws:SourceAccount` or an `ExternalId` condition.
- Separate production into its own account/OU with explicit boundaries; apply Service Control Policies (SCPs) to cap maximum privileges.
- Enable and alert on CloudTrail `AssumeRole` events for high-privilege roles.

References

AWS IAM Best Practices – least privilege • CIS AWS Foundations Benchmark 1.x • CWE-269

4.3 API-01 — Broken Object-Level Authorization (BOLA / IDOR) on Invoice API

Finding ID	API-01	Severity	High (CVSS 8.1)
Category	OWASP API Top 10 2023 – API1 (BOLA)	CWE	CWE-639
Affected area	API	CVSS vector	CVSS:3.1/AV:N/AC:L/PR:L/UI:N/S:U/C:H/I:H/A:N
Affected asset	api.novapay.example — GET/PUT /v2/invoices/{invoiceId}		

Description

The invoice API authenticates the caller's token but does not verify that the authenticated user is entitled to the requested `invoiceId`. Any valid, low-tier account can read and modify invoices belonging to other tenants by changing the identifier in the path.

Invoice identifiers are sequential integers, making other tenants' records trivially discoverable.

Business Impact

Any registered user can enumerate and read other customers' invoices (amounts, bank/beneficiary details, line items) and, via the PUT method, alter invoice fields such as the payable amount or payee reference — a confidentiality and integrity breach across the entire tenant base.

Proof of Concept

PROOF OF CONCEPT *evidence summary — sensitive values redacted*

1. Note the caller's own object

```
// Account A (test tenant) legitimately owns invoice 100500.
GET /v2/invoices/100500   Authorization: Bearer <token-A>   -> 200 OK (own data)
```

2. Request another tenant's object

```
// Using the SAME token, request a neighbouring sequential ID owned by Account B.
GET /v2/invoices/100501   Authorization: Bearer <token-A>
✓ 200 OK — returned Account B's invoice; no ownership check performed
```

3. Confirm write access (ROE-limited)

```
// A PUT to a dedicated second test tenant's invoice was accepted, changing a
// benign description field only. No financial field on a real record was altered.
✓ 200 OK — cross-tenant modification accepted on the test object
```

Remediation

- Enforce object-level authorization on every request: verify the authenticated principal owns, or is explicitly granted access to, the requested `invoiceId` before returning or modifying it.
- Replace sequential integer IDs with unguessable UUIDs to reduce enumeration.
- Centralise access-control checks in middleware so no endpoint can omit them; add automated tests that assert cross-tenant requests return 403.
- Log and rate-limit access patterns indicative of ID enumeration.

References

OWASP API Security Top 10 2023 – API1:2023 • CWE-639 • CWE-284

4.4 API-02 — Excessive Data Exposure in Authentication Response

Finding ID	API-02	Severity	High (CVSS 7.5)
Category	OWASP API Top 10 2023 – API3 (Broken Property Level Authz)	CWE	CWE-213
Affected area	API	CVSS vector	CVSS:3.1/AV:N/AC:L/PR:N/UI:N/S:U/C:H/I:N/A:N
Affected asset	`api.novapay.example – POST /v2/auth/login`		

Description

The login endpoint returns a serialized user object that includes internal fields never used by the client: the password hash, a password-reset token, MFA seed material and internal role flags. The front-end simply ignores these fields, but they are fully visible in the raw HTTP response.

Business Impact

Any user who logs in — or any attacker who captures a single login response — obtains the account's password hash and a valid password-reset token, enabling offline cracking and account takeover without further interaction. The MFA seed exposure undermines the second factor entirely.

Proof of Concept

PROOF OF CONCEPT *evidence summary — sensitive values redacted*

1. Authenticate and inspect the raw response

```
POST /v2/auth/login   { "email": "<test user>", "password": "<test pw>" }
// The JSON response body contained far more than a session token:
✓ fields present: passwordHash, resetToken, mfaSeed, isInternalAdmin
// none of which are referenced by the web or mobile client
```

Remediation

- Return only the fields the client needs (e.g. display name, session token, non-sensitive profile data) using an explicit response DTO / serializer allow-list — never serialize the ORM entity directly.
- Keep credential, reset-token and MFA-seed material server-side only.
- Add a schema/contract test that fails if a response contains disallowed properties.

References

OWASP API Security Top 10 2023 – API3:2023 • CWE-213 • CWE-359

4.5 WEB-02 — Stored Cross-Site Scripting in Merchant Display Name

Finding ID	WEB-02	Severity	High (CVSS 8.7)
Category	OWASP A03:2021 – Injection (XSS)	CWE	CWE-79
Affected area	Web Application	CVSS vector	CVSS:3.1/AV:N/AC:L/PR:L/UI:R/S:C/C:H/I:H/A:N
Affected asset	app.novapay.example — merchant profile / admin console		

Description

The merchant display name is stored without output encoding and rendered verbatim in two places: the public profile page and the internal admin console that staff use to review merchants. A script placed in the display name executes in the browser of anyone viewing it, including administrators (scope change into the privileged admin context).

Business Impact

A self-registered merchant can store a payload that runs in an administrator's authenticated session when staff review the account, allowing session/token theft or actions performed as the admin. Because it also renders on public pages, ordinary visitors are exposed as well.

Proof of Concept

PROOF OF CONCEPT *evidence summary — sensitive values redacted*

1. Store the payload

```
// During merchant signup, the display-name field accepted HTML/script markup
// instead of treating it as plain text.
✓ A benign marker script was saved in the display name (no real payload used)
```

2. Confirm execution in the admin view

```
// When the test merchant was opened in the admin review console, the stored
// marker executed, proving script runs in the privileged context.
✓ Marker executed in admin console session (confirmed via controlled callback)
```

Remediation

- Apply context-aware output encoding wherever the display name is rendered (HTML body, attributes); prefer a framework that auto-escapes by default.
- Validate/normalise the field on input (reject markup), as defence in depth.
- Deploy a strict Content-Security-Policy to limit script execution, and set `HttpOnly`/`SameSite` on session cookies.
- Re-encode historical stored values when deploying the fix.

References

OWASP A03:2021 • OWASP XSS Prevention Cheat Sheet • CWE-79

4.6 CLD-02 — Publicly Writable S3 Bucket Serving Application Assets

Finding ID	**CLD-02**	**Severity**	**High (CVSS 7.4)**
Category	Cloud Storage Misconfiguration	**CWE**	CWE-732
Affected area	Cloud (AWS)	**CVSS vector**	`CVSS:3.1/AV:N/AC:H/PR:N/UI:N/S:U/C:H/I:H/A:N`
Affected asset	`s3://novapay-public-assets`		

Description

The bucket that serves the web application's static JavaScript and CSS has a bucket policy allowing object writes from unauthenticated principals. The same bucket is the origin for the production front-end, so its contents are executed in every user's browser.

Business Impact

An attacker can overwrite the application's JavaScript with malicious code that then runs for every visitor — effectively a site-wide compromise and a supply-chain style attack (credential harvesting, redirection, card skimming). Confidentiality and integrity are both at high risk; the High access complexity reflects the conditions required to locate and weaponise the writable path.

Proof of Concept

PROOF OF CONCEPT *evidence summary — sensitive values redacted*

1. Review bucket permissions

```
// Read-only audit of the bucket policy and ACLs showed:
✓ Public write (PutObject) permitted to an anonymous principal
✓ Bucket is the CloudFront origin for app.novapay.example static assets
```

2. Confirm writability (benign, ROE-limited)

```
// With approval, a harmless text file was uploaded to a scratch key and then
// deleted. No application asset was touched.
✓ PutObject to novapay-public-assets/_pentest_marker.txt succeeded (then removed)
```

Remediation

- Remove all public write permissions; restrict writes to a specific CI/CD deployment role.
- Enable S3 Block Public Access at the account and bucket level.
- Enable bucket versioning and object-level logging so tampering is detectable and recoverable.
- Serve assets via CloudFront with Origin Access Control and consider Subresource Integrity (SRI) on critical scripts.

References

AWS S3 Block Public Access • CIS AWS Benchmark – S3 • CWE-732

4.7 API-03 — Missing Rate Limiting on Authentication and OTP Endpoints

Finding ID	API-03	Severity	Medium (CVSS 6.5)
Category	OWASP API Top 10 2023 – API4 (Unrestricted Resource Consumption)	CWE	CWE-307
Affected area	API	CVSS vector	CVSS:3.1/AV:N/AC:L/PR:N/UI:N/S:U/C:L/I:L/A:N
Affected asset	api.novapay.example – /v2/auth/login, /v2/auth/otp/verify		

Description

Neither the password login nor the OTP-verification endpoint enforces rate limiting, account lockout or progressive delays. Automated submission of many attempts is accepted without throttling or CAPTCHA.

Business Impact

The login endpoint is exposed to credential-stuffing and password-spraying; the six-digit OTP endpoint is brute-forceable within its validity window, undermining MFA. Impact is rated Medium because success still depends on weak or reused credentials.

Proof of Concept

PROOF OF CONCEPT *evidence summary — sensitive values redacted*

1. Submit repeated attempts

```
// A controlled burst of sequential login attempts for one test account was sent.
✓ All requests returned normal 401s with no throttling, lockout or delay
// OTP verify behaved the same – no limit on verification attempts observed
```

Remediation

- Enforce per-account and per-IP rate limits with exponential back-off and temporary lockout after repeated failures.
- Cap OTP verification attempts per code and shorten code validity; rotate on failure.
- Add CAPTCHA / proof-of-work after a threshold and alert on distributed attempts.
- Monitor for credential-stuffing signatures at the edge/WAF.

References

OWASP API Security Top 10 2023 – API4:2023 • CWE-307 • NIST SP 800-63B

4.8 API-04 — Server-Side Request Forgery via Webhook URL Validation Bypass

Finding ID	API-04	Severity	Medium (CVSS 6.5)
Category	OWASP API Top 10 2023 – API7 (SSRF)	CWE	CWE-918
Affected area	API	CVSS vector	CVSS:3.1/AV:N/AC:L/PR:N/UI:R/S:U/C:H/I:N/A:N
Affected asset	api.novapay.example — POST /v2/webhooks		

Description

The webhook registration feature fetches a user-supplied URL to validate it. The allow-list check is applied to the originally submitted hostname but not re-checked after HTTP redirects, so a request to an attacker-controlled host can redirect the server into the internal network, including the cloud instance metadata service.

Business Impact

An authenticated user can coax the backend into making requests to internal-only services and the cloud metadata endpoint, potentially retrieving instance credentials. Rated Medium rather than High because exploitation requires a specific redirect setup and the metadata service enforces session-based protections in the test environment.

Proof of Concept

PROOF OF CONCEPT *evidence summary — sensitive values redacted*

1. Register a webhook pointing at an attacker-controlled validator

```
POST /v2/webhooks   { "url": "https://<external host under test control>/hook" }
// The initial host passed the allow-list check.
```

2. Redirect the validation fetch inward

```
// The external host answered the validation fetch with an HTTP redirect to an
// internal-only address. The server followed it without re-validating.
✓ Backend issued a request to an internal metadata address (link-local range)
// Observed via out-of-band interaction logs; no credentials were exfiltrated.
```

Remediation

- Re-apply the host allow-list on every hop, including after each redirect, or disable redirect-following for validation fetches.
- Block outbound requests to private, loopback and link-local ranges (including the metadata IP) at the egress layer.
- Enforce IMDSv2 (session-oriented metadata) and a minimal hop limit.
- Fetch user URLs from an isolated egress proxy with no internal network reachability.

References

OWASP API Security Top 10 2023 – API7:2023 • OWASP SSRF Prevention Cheat Sheet • CWE-918

4.9 WEB-03 — Session Token Does Not Rotate After Privilege Change

Finding ID	WEB-03	Severity	Medium (CVSS 5.9)
Category	OWASP A07:2021 – Identification & Authentication Failures	CWE	CWE-384
Affected area	Web Application	CVSS vector	CVSS:3.1/AV:N/AC:H/PR:H/UI:N/S:U/C:H/I:H/A:N
Affected asset	app.novapay.example — session management		

Description

Session identifiers are not reissued when a user authenticates or when their privileges are elevated (e.g. stepping up to an admin action). A token captured before the privilege change remains valid afterward, weakening protection against session fixation and stale-session misuse.

Business Impact

An attacker who fixes or captures a session before a privilege transition can continue to use it with the elevated privileges. Exploitation requires an existing high-privilege context and specific timing, hence the Medium rating.

Proof of Concept

PROOF OF CONCEPT *evidence summary — sensitive values redacted*

1. Record the pre-change token

```
// Session token value noted before the step-up/elevation event.
```

2. Compare after the privilege change

```
✓ Token value unchanged after re-authentication and after admin step-up
// The same identifier continued to grant access at the new privilege level
```

Remediation

- Issue a brand-new session identifier on every authentication and on any privilege elevation; invalidate the previous one server-side.
- Bind sessions to additional context and enforce absolute and idle timeouts.
- Set `HttpOnly`, `Secure` and `SameSite` on session cookies.

References

OWASP A07:2021 • OWASP Session Management Cheat Sheet • CWE-384

4.10 WEB-04 — Verbose Error Messages Disclose Stack Traces and Framework Versions

Finding ID	WEB-04	Severity	Medium (CVSS 5.3)
Category	OWASP A05:2021 – Security Misconfiguration	CWE	CWE-209
Affected area	Web Application	CVSS vector	CVSS:3.1/AV:N/AC:L/PR:N/UI:N/S:U/C:L/I:N/A:N
Affected asset	app.novapay.example — multiple endpoints		

Description

Several endpoints return full stack traces, internal file paths and exact framework/library versions when an unexpected input is supplied. Production error handling has not been enabled.

Business Impact

The disclosed detail hands an attacker a map of the technology stack, internal structure and specific component versions, significantly lowering the effort to find and chain further vulnerabilities. No data is directly exposed, so impact is limited to information disclosure.

Proof of Concept

PROOF OF CONCEPT *evidence summary — sensitive values redacted*

1. Trigger an unhandled error

```
// Supplying a malformed value to a typed parameter produced a 500 whose body
// contained the full exception:
✓ Stack trace, absolute server file paths and framework version returned to the client
```

Remediation

- Enable production error handling: return generic error messages with a correlation ID and log details server-side only.
- Remove version banners and debug flags from production responses.
- Add a test that asserts 4xx/5xx responses contain no stack traces or paths.

References

OWASP A05:2021 • CWE-209 • CWE-200

4.11 WEB-05 — Missing Security Headers (HSTS, CSP, X-Content-Type-Options)

Finding ID	WEB-05	Severity	Low (CVSS 3.7)
Category	OWASP A05:2021 – Security Misconfiguration	CWE	CWE-693
Affected area	Web Application	CVSS vector	CVSS:3.1/AV:N/AC:H/PR:N/UI:N/S:U/C:L/I:N/A:N
Affected asset	app.novapay.example — all responses		

Description

Responses omit several recommended security headers, including Strict-Transport-Security, Content-Security-Policy, X-Content-Type-Options and Referrer-Policy. These do not create a vulnerability on their own but remove layers of defence against downgrade, injection and MIME-sniffing attacks.

Business Impact

Absence of these headers makes other issues (e.g. XSS, protocol downgrade) easier to exploit and is commonly flagged in compliance reviews. Direct impact is low.

Proof of Concept

PROOF OF CONCEPT *evidence summary — sensitive values redacted*

1. Inspect response headers

```
// A standard request to the application root was examined:
✓ Strict-Transport-Security: (absent)
✓ Content-Security-Policy: (absent)
✓ X-Content-Type-Options: (absent)
```

Remediation

- Add `Strict-Transport-Security` with a long max-age and `includeSubDomains`.
- Deploy a tuned `Content-Security-Policy`, plus `X-Content-Type-Options: nosniff` and a sensible `Referrer-Policy`.
- Set these centrally at the edge/reverse proxy so all responses inherit them.

References

OWASP Secure Headers Project • CWE-693 • A05:2021

4.12 WEB-06 — User/E-mail Enumeration via Distinct Responses

Finding ID	WEB-06	Severity	Low (CVSS 4.3)
Category	OWASP A07:2021 – Identification & Authentication Failures	CWE	CWE-203
Affected area	Web Application	CVSS vector	CVSS:3.1/AV:N/AC:L/PR:L/UI:N/S:U/C:L/I:N/A:N
Affected asset	app.novapay.example – /register, /forgot-password		

Description

The registration and password-reset flows return observably different messages (and timing) for addresses that already exist versus those that do not, allowing an attacker to confirm which e-mail addresses hold accounts.

Business Impact

An attacker can compile a list of valid customer accounts to target with phishing or credential-stuffing. Low direct impact; it is an enabler for other attacks.

Proof of Concept

PROOF OF CONCEPT *evidence summary — sensitive values redacted*

1. Compare responses for known vs unknown addresses

```
✓ Existing address -> "an account with this e-mail already exists"
✓ Unknown address  -> "we could not find that e-mail"
// The differing messages confirm account existence
```

Remediation

- Return a single, identical, non-committal message for both cases (e.g. "if an account exists, we have sent instructions").
- Normalise response timing and apply rate limiting to these flows.

References

OWASP A07:2021 • CWE-203 • OWASP WSTG – Account Enumeration

4.13 CLD-03 — CloudTrail Logging Disabled in a Secondary Region

Finding ID	CLD-03	Severity	Low (CVSS 3.1)
Category	Cloud Logging & Monitoring Gap	CWE	CWE-778
Affected area	Cloud (AWS)	CVSS vector	CVSS:3.1/AV:N/AC:H/PR:N/UI:R/S:U/C:L/I:N/A:N
Affected asset	AWS account 4411-xxxx-8820 — region eu-west-2		

Description

Organisation CloudTrail is enabled in the primary region but a multi-region trail was not configured, leaving one active secondary region without management-event logging.

Business Impact

Actions taken in the unlogged region would not be recorded, hampering incident investigation and detection. There is no direct confidentiality or integrity impact, so the rating is Low.

Proof of Concept

PROOF OF CONCEPT *evidence summary — sensitive values redacted*

1. Review trail configuration

```
✓ CloudTrail trail present in primary region only; `IsMultiRegionTrail` = false
// eu-west-2 contained running resources but no management-event trail
```

Remediation

- Enable a multi-region organisation CloudTrail so every region is logged.
- Ship logs to a dedicated, access-restricted logging account with integrity validation enabled.
- Alert on any attempt to stop or delete trails.

References

CIS AWS Benchmark – Logging • AWS CloudTrail Best Practices • CWE-778

4.14 INF-01 — TLS Configuration Supports a Legacy Cipher Suite

Finding ID	INF-01	Severity	Informational (CVSS 0.0)
Category	Hardening / Best Practice	CWE	CWE-326
Affected area	Web Application	CVSS vector	CVSS:3.1/AV:N/AC:L/PR:N/UI:N/S:U/C:N/I:N/A:N
Affected asset	app.novapay.example — TLS endpoint		

Description

The TLS endpoint negotiates modern protocols (TLS 1.2/1.3) but still advertises one legacy CBC-mode cipher suite for backward compatibility. No practical attack was demonstrated; this is a hardening observation.

Business Impact

No direct impact. Removing legacy suites tightens the configuration and helps pass compliance scanners (PCI DSS, Mozilla Modern profile).

Proof of Concept

PROOF OF CONCEPT *evidence summary — sensitive values redacted*

1. Enumerate supported cipher suites

```
✓ One legacy CBC cipher suite still offered alongside modern AEAD suites
```

Remediation

- Restrict the server to TLS 1.2+ with AEAD cipher suites only (prefer TLS 1.3).
- Disable legacy CBC suites and align with the Mozilla "Modern" configuration.

References

Mozilla TLS Guidelines • CWE-326 • PCI DSS v4.0

4.15 INF-02 — API Lacks a Documented Versioning & Deprecation Policy

Finding ID	INF-02	Severity	Informational (CVSS 0.0)
Category	Governance / Best Practice	CWE	CWE-1059
Affected area	API	CVSS vector	CVSS:3.1/AV:N/AC:L/PR:N/UI:N/S:U/C:N/I:N/A:N
Affected asset	api.novapay.example		

Description

Both `/v1` and `/v2` of the API remain reachable, but there is no published deprecation schedule or inventory of which clients use the older version. Older versions can accumulate unpatched behaviour over time.

Business Impact

No immediate security impact, but undocumented legacy endpoints are a common source of drift where fixes applied to the current version are missed on the old one.

Proof of Concept

PROOF OF CONCEPT *evidence summary — sensitive values redacted*

1. Confirm multiple live versions

```
✓ Both /v1 and /v2 responded; no deprecation notice or sunset header present
```

Remediation

- Maintain an inventory of API versions and consumers.
- Publish a deprecation policy with `Sunset` headers and retire unused versions.
- Ensure security fixes are applied to, or the endpoint removed from, every supported version.

References

OWASP API9:2023 – Improper Inventory Management • CWE-1059

4.16 INF-03 — Cloud Resources Missing Consistent Tagging

Finding ID	INF-03	Severity	Informational (CVSS 0.0)
Category	Governance / Best Practice	CWE	CWE-1059
Affected area	Cloud (AWS)	CVSS vector	CVSS:3.1/AV:N/AC:L/PR:N/UI:N/S:U/C:N/I:N/A:N
Affected asset	AWS account 4411-xxxx-8820 — multiple services		

Description

A number of production resources (EC2, S3, RDS) lack owner, environment and data-classification tags. This is an operational and governance observation rather than a vulnerability.

Business Impact

Inconsistent tagging slows incident response (harder to identify owners and blast radius) and weakens the ability to enforce tag-based access controls. No direct security impact.

Proof of Concept

PROOF OF CONCEPT *evidence summary — sensitive values redacted*

1. Review resource tags

```
✓ Several production resources had no owner/environment/data-class tags
```

Remediation

- Define a mandatory tagging standard (owner, environment, data classification).
- Enforce it with SCPs / tag policies and remediate existing untagged resources.
- Use tags to drive least-privilege access and cost/incident ownership.

References

AWS Tagging Best Practices • CWE-1059

5 Remediation Roadmap

The roadmap groups remediation by priority to help the team sequence the work. Timeframes are indicative and should be adjusted to the organisation's risk appetite and release cadence.

Priority	Findings	Action	Target
P1 — Immediate	WEB-01, CLD-01	Parameterise the vulnerable query; apply least-privilege IAM and pin the trust policy	24–72 hours
P2 — Urgent	API-01, API-02, WEB-02, CLD-02	Enforce object-level authz, response allow-lists and output encoding; block public S3 write	Within 2 weeks
P3 — Planned	API-03, API-04, WEB-03, WEB-04	Rate limiting, SSRF egress controls, session rotation, production error handling	Within 30–60 days
P4 — Routine	WEB-05, WEB-06, CLD-03	Security headers, enumeration hardening, multi-region logging	Next maintenance cycle
P5 — Hardening	INF-01, INF-02, INF-03	TLS hardening, API inventory and deprecation policy, cloud tagging standard	Backlog / ongoing

5.1 Retesting

American Dream Cyber Security includes complimentary retesting of remediated findings. Once fixes are deployed, tell your engagement lead; we will verify each issue and issue an updated attestation showing the closed items.

6 Appendix

A. Severity & CVSS Reference

CVSS 3.1 base scores describe the intrinsic characteristics of a vulnerability: how it is reached, how hard it is to exploit, the privileges and user interaction required, and the impact on confidentiality, integrity and availability. This report uses the qualitative bands defined in Section 2.4.

B. Tooling

Testing combined manual techniques with industry-standard and proprietary tooling, including the American Dream Cyber Scanner (hybrid SAST / DAST / SCA with cloud and API analysis) alongside standard proxy, enumeration and cloud-audit utilities. Every automated result was manually verified before inclusion.

C. Document Control

Version	Date	Author	Notes
0.1	27 Oct 2026	Lead Consultant	Internal draft
0.9	28 Oct 2026	QA / Technical Review	Peer review and QA
1.0	29 Oct 2026	Engagement Manager	Final issued to client

D. Common Attack Vectors Tested

Our assessment methodology for web applications, APIs and cloud environments includes the attack categories and techniques below. They were applied to the in-scope systems as appropriate to the agreed scope, time frame and rules of engagement, using manual consultant-led testing supported by the American Dream Cyber Scanner. All automated results were manually verified before being reported.

Scope & limitations. The categories below describe the techniques included in our methodology. "No issue observed" means no vulnerability was identified by the techniques applied during the testing window; it is not a guarantee that the systems are free of vulnerabilities in that category. Testing is point-in-time and bounded by the agreed scope (see the Confidentiality & Legal Notice).

98	12	10/10	10/10	150+
TEST CASES IN CHECKLIST	TEST CATEGORIES	OWASP TOP 10 CATEGORIES ASSESSED	OWASP API TOP 10 CATEGORIES ASSESSED	ATTACK TECHNIQUES IN METHODOLOGY

Attack category	Techniques included in the methodology	Observed findings
Injection	SQL injection (error, boolean, time-based, UNION, second-order), NoSQL and ORM injection, OS command injection, LDAP, XPath, XML / XXE, SSI, IMAP/SMTP, server-side template injection (SSTI), code injection, format string, expression-language injection	WEB-01
Cross-Site Scripting & client-side	Reflected, stored and DOM-based XSS, HTML and CSS injection, client-side URL redirect, JavaScript execution sinks, postMessage / web messaging, WebSockets, cross-site script inclusion (XSSI), clickjacking, browser storage exposure, reverse tabnabbing	WEB-02
Authentication	Credential stuffing and password spraying resistance, brute force and lockout, default credentials, MFA / OTP bypass, password reset and change flows, remember-me tokens, authentication bypass via forced browsing and parameter tampering, alternative-channel authentication	API-03
Session management	Session fixation, token entropy and predictability, cookie flags (Secure, HttpOnly, SameSite), CSRF, logout and timeout behaviour, concurrent sessions, session puzzling, JWT weaknesses (alg=none, weak secret, key confusion, missing expiry)	WEB-03
Access control	Horizontal and vertical privilege escalation, IDOR / BOLA, function-level authorization, forced browsing to admin functions, path traversal and file inclusion, multi-tenant isolation, role and permission matrix testing	API-01
API-specific	OpenAPI-driven endpoint enumeration, mass assignment, excessive data exposure, property-level authorization, rate limiting and resource consumption, undocumented / shadow endpoints, API versioning, content-type confusion, GraphQL introspection (where applicable)	API-02, API-03, INF-02
Server-side request & file handling	SSRF (direct, blind, redirect-based, DNS rebinding), cloud metadata access, file upload (type, content and size validation, polyglots, path manipulation), archive extraction, insecure deserialization	API-04
HTTP protocol attacks	HTTP request smuggling and response splitting, host header injection, cache poisoning and cache deception, HTTP verb tampering, HTTP parameter pollution, CORS misconfiguration, open redirects	No issue observed
Configuration & hardening	Security headers (HSTS, CSP, X-Content-Type-Options, Referrer-Policy), HTTP methods, debug and admin interfaces, backup and unreferenced files, directory listing, subdomain takeover, verbose errors and stack traces	WEB-04, WEB-05
Cryptography & transport	TLS protocol and cipher suites, certificate validation, HSTS, padding oracle, weak hashing and encryption, sensitive data over unencrypted channels, secrets in client-side code	INF-01
Business logic	Workflow circumvention, price and quantity tampering, race conditions and process timing, function-use limits, payment and payout flow integrity, request forgery, anti-automation controls	No issue observed
Information disclosure & recon	Search-engine and OSINT discovery, metafiles (robots.txt, sitemap, security.txt), technology fingerprinting, source-map and comment leakage, user and e-mail enumeration, error message leakage	WEB-06
Cloud (AWS)	IAM least privilege and trust policies, privilege-escalation paths, S3 bucket policies and ACLs, public snapshots and AMIs, security groups and network exposure, IMDSv2, secrets management, CloudTrail / logging coverage, resource tagging and governance	CLD-01, CLD-02, CLD-03, INF-03

Attack category	Techniques included in the methodology	Observed findings
Components & supply chain	Outdated frameworks and libraries (SCA / SBOM review), known CVEs in exposed services, third-party scripts and Subresource Integrity, dependency confusion indicators	**No issue observed**

E. OWASP Top 10 & OWASP API Security Top 10 Coverage

Our methodology maps to the two most widely recognised industry risk lists: the OWASP Top 10 (2021) for web applications and the OWASP API Security Top 10 (2023). Every category was assessed within the agreed scope. The tables show the findings associated with each category; "No issue observed" reflects the results of testing during the engagement window only.

OWASP Top 10 — 2021 (Web Applications)

ID	Category	Status	Observed findings
A01	Broken Access Control	✓ ASSESSED	API-01
A02	Cryptographic Failures	✓ ASSESSED	INF-01
A03	Injection (incl. Cross-Site Scripting)	✓ ASSESSED	WEB-01, WEB-02
A04	Insecure Design	✓ ASSESSED	API-03
A05	Security Misconfiguration	✓ ASSESSED	WEB-04, WEB-05, CLD-02
A06	Vulnerable and Outdated Components	✓ ASSESSED	No issue observed
A07	Identification and Authentication Failures	✓ ASSESSED	WEB-03, WEB-06
A08	Software and Data Integrity Failures	✓ ASSESSED	CLD-02
A09	Security Logging and Monitoring Failures	✓ ASSESSED	CLD-03
A10	Server-Side Request Forgery (SSRF)	✓ ASSESSED	API-04

OWASP API Security Top 10 — 2023

ID	Category	Status	Observed findings
API1	Broken Object Level Authorization	✓ ASSESSED	API-01
API2	Broken Authentication	✓ ASSESSED	API-03
API3	Broken Object Property Level Authorization	✓ ASSESSED	API-02
API4	Unrestricted Resource Consumption	✓ ASSESSED	API-03
API5	Broken Function Level Authorization	✓ ASSESSED	No issue observed
API6	Unrestricted Access to Sensitive Business Flows	✓ ASSESSED	No issue observed
API7	Server Side Request Forgery	✓ ASSESSED	API-04
API8	Security Misconfiguration	✓ ASSESSED	WEB-04, WEB-05
API9	Improper Inventory Management	✓ ASSESSED	INF-02
API10	Unsafe Consumption of APIs	✓ ASSESSED	No issue observed

F. Web Application Security Test Checklist

Our web application and API test checklist is aligned with OWASP industry testing guidance and contains **98 test cases** in **12 categories**. Each test case is listed below with the outcome recorded for the in-scope systems during the testing window. Test cases that resulted in a finding link to the corresponding entry in Section 4; test cases for technologies not present in scope are marked as not applicable.

98	79	15	4
TEST CASES	NO ISSUE OBSERVED	RAISED A FINDING	NOT APPLICABLE

Legend: **NO ISSUE OBSERVED** no vulnerability identified by the techniques applied **FINDING** vulnerability identified (see finding ID) **NOT APPLICABLE** technology not present in scope

Test ID	Test case	Result
Information Gathering 9 test cases		
INFO-01	Conduct Search Engine Discovery Reconnaissance for Information Leakage	NO ISSUE OBSERVED
INFO-02	Fingerprint Web Server	NO ISSUE OBSERVED
INFO-03	Review Webserver Metafiles for Information Leakage	NO ISSUE OBSERVED
INFO-04	Enumerate Applications on Webserver	NO ISSUE OBSERVED
INFO-05	Review Webpage Content for Information Leakage	NO ISSUE OBSERVED
INFO-06	Identify Application Entry Points	NO ISSUE OBSERVED
INFO-07	Map Execution Paths Through Application	NO ISSUE OBSERVED
INFO-08	Fingerprint Web Application Framework	NO ISSUE OBSERVED
INFO-10	Map Application Architecture	NO ISSUE OBSERVED
Configuration & Deployment Management Testing 12 test cases		
CONF-01	Test Network Infrastructure Configuration	NO ISSUE OBSERVED
CONF-02	Test Application Platform Configuration	NO ISSUE OBSERVED
CONF-03	Test File Extensions Handling for Sensitive Information	NO ISSUE OBSERVED
CONF-04	Review Old Backup and Unreferenced Files for Sensitive Information	NO ISSUE OBSERVED
CONF-05	Enumerate Infrastructure and Application Admin Interfaces	NO ISSUE OBSERVED
CONF-06	Test HTTP Methods	NO ISSUE OBSERVED
CONF-07	Test HTTP Strict Transport Security *Missing Security Headers (HSTS, CSP, X-Content-Type-Options)*	FINDING · WEB-05
CONF-08	Test RIA Cross Domain Policy	NO ISSUE OBSERVED
CONF-09	Test File Permission	NO ISSUE OBSERVED
CONF-10	Test for Subdomain Takeover	NO ISSUE OBSERVED
CONF-11	Test Cloud Storage *Publicly Writable S3 Bucket Serving Application Assets*	FINDING · CLD-02
CONF-12	Testing for Content Security Policy *Missing Security Headers (HSTS, CSP, X-Content-Type-Options)*	FINDING · WEB-05
Identity Management Testing 5 test cases		
IDNT-01	Test Role Definitions	NO ISSUE OBSERVED
IDNT-02	Test User Registration Process	NO ISSUE OBSERVED
IDNT-03	Test Account Provisioning Process	NO ISSUE OBSERVED
IDNT-04	Testing for Account Enumeration and Guessable User Account *User/E-mail Enumeration via Distinct Responses*	FINDING · WEB-06

Test ID	Test case	Result
IDNT-05	Testing for Weak or Unenforced Username Policy	NO ISSUE OBSERVED
Authentication Testing 10 test cases		
ATHN-01	Testing for Credentials Transported over an Encrypted Channel	NO ISSUE OBSERVED
ATHN-02	Testing for Default Credentials	NO ISSUE OBSERVED
ATHN-03	Testing for Weak Lock Out Mechanism *Missing Rate Limiting on Authentication and OTP Endpoints*	FINDING · API-03
ATHN-04	Testing for Bypassing Authentication Schema	NO ISSUE OBSERVED
ATHN-05	Testing for Vulnerable Remember Password	NO ISSUE OBSERVED
ATHN-06	Testing for Browser Cache Weaknesses	NO ISSUE OBSERVED
ATHN-07	Testing for Weak Password Policy	NO ISSUE OBSERVED
ATHN-08	Testing for Weak Security Question Answer *Application does not use security questions*	NOT APPLICABLE
ATHN-09	Testing for Weak Password Change or Reset Functionalities	NO ISSUE OBSERVED
ATHN-10	Testing for Weaker Authentication in Alternative Channel	NO ISSUE OBSERVED
Authorization Testing 4 test cases		
ATHZ-01	Testing Directory Traversal File Include	NO ISSUE OBSERVED
ATHZ-02	Testing for Bypassing Authorization Schema *Broken Object-Level Authorization (BOLA / IDOR) on Invoice API*	FINDING · API-01
ATHZ-03	Testing for Privilege Escalation	NO ISSUE OBSERVED
ATHZ-04	Testing for Insecure Direct Object References *Broken Object-Level Authorization (BOLA / IDOR) on Invoice API*	FINDING · API-01
Session Management Testing 10 test cases		
SESS-01	Testing for Session Management Schema	NO ISSUE OBSERVED
SESS-02	Testing for Cookies Attributes	NO ISSUE OBSERVED
SESS-03	Testing for Session Fixation *Session Token Does Not Rotate After Privilege Change*	FINDING · WEB-03
SESS-04	Testing for Exposed Session Variables	NO ISSUE OBSERVED
SESS-05	Testing for Cross Site Request Forgery	NO ISSUE OBSERVED
SESS-06	Testing for Logout Functionality	NO ISSUE OBSERVED
SESS-07	Testing Session Timeout	NO ISSUE OBSERVED
SESS-08	Testing for Session Puzzling	NO ISSUE OBSERVED
SESS-09	Testing for Session Hijacking	NO ISSUE OBSERVED
SESS-10	Testing JSON Web Tokens	NO ISSUE OBSERVED
Input Validation Testing 19 test cases		
INPV-01	Testing for Reflected Cross Site Scripting	NO ISSUE OBSERVED
INPV-02	Testing for Stored Cross Site Scripting *Stored Cross-Site Scripting in Merchant Display Name*	FINDING · WEB-02
INPV-03	Testing for HTTP Verb Tampering	NO ISSUE OBSERVED
INPV-04	Testing for HTTP Parameter Pollution	NO ISSUE OBSERVED
INPV-05	Testing for SQL Injection (incl. NoSQL, ORM and client-side) *SQL Injection in Unauthenticated Merchant Search*	FINDING · WEB-01
INPV-06	Testing for LDAP Injection *No LDAP-backed functionality in scope*	NOT APPLICABLE
INPV-07	Testing for XML Injection (incl. XXE)	NO ISSUE OBSERVED
INPV-08	Testing for SSI Injection	NO ISSUE OBSERVED

Test ID	Test case	Result
INPV-09	Testing for XPath Injection	NO ISSUE OBSERVED
INPV-10	Testing for IMAP SMTP Injection	NO ISSUE OBSERVED
INPV-11	Testing for Code Injection (incl. LFI / RFI)	NO ISSUE OBSERVED
INPV-12	Testing for Command Injection	NO ISSUE OBSERVED
INPV-13	Testing for Format String Injection	NO ISSUE OBSERVED
INPV-14	Testing for Incubated Vulnerability	NO ISSUE OBSERVED
INPV-15	Testing for HTTP Splitting Smuggling	NO ISSUE OBSERVED
INPV-16	Testing for HTTP Incoming Requests	NO ISSUE OBSERVED
INPV-17	Testing for Host Header Injection	NO ISSUE OBSERVED
INPV-18	Testing for Server-side Template Injection	NO ISSUE OBSERVED
INPV-19	Testing for Server-Side Request Forgery *Server-Side Request Forgery via Webhook URL Validation Bypass*	FINDING · API-04
Testing for Error Handling 2 test cases		
ERRH-01	Testing for Improper Error Handling *Verbose Error Messages Disclose Stack Traces and Framework Versions*	FINDING · WEB-04
ERRH-02	Testing for Stack Traces *Verbose Error Messages Disclose Stack Traces and Framework Versions*	FINDING · WEB-04
Testing for Weak Cryptography 4 test cases		
CRYP-01	Testing for Weak Transport Layer Security *TLS Configuration Supports a Legacy Cipher Suite*	FINDING · INF-01
CRYP-02	Testing for Padding Oracle	NO ISSUE OBSERVED
CRYP-03	Testing for Sensitive Information Sent via Unencrypted Channels	NO ISSUE OBSERVED
CRYP-04	Testing for Weak Encryption	NO ISSUE OBSERVED
Business Logic Testing 9 test cases		
BUSL-01	Test Business Logic Data Validation	NO ISSUE OBSERVED
BUSL-02	Test Ability to Forge Requests	NO ISSUE OBSERVED
BUSL-03	Test Integrity Checks	NO ISSUE OBSERVED
BUSL-04	Test for Process Timing	NO ISSUE OBSERVED
BUSL-05	Test Number of Times a Function Can Be Used Limits *Missing Rate Limiting on Authentication and OTP Endpoints*	FINDING · API-03
BUSL-06	Testing for the Circumvention of Work Flows	NO ISSUE OBSERVED
BUSL-07	Test Defenses Against Application Misuse	NO ISSUE OBSERVED
BUSL-08	Test Upload of Unexpected File Types	NO ISSUE OBSERVED
BUSL-09	Test Upload of Malicious Files	NO ISSUE OBSERVED
Client-side Testing 13 test cases		
CLNT-01	Testing for DOM-Based Cross Site Scripting	NO ISSUE OBSERVED
CLNT-02	Testing for JavaScript Execution	NO ISSUE OBSERVED
CLNT-03	Testing for HTML Injection	NO ISSUE OBSERVED
CLNT-04	Testing for Client-side URL Redirect	NO ISSUE OBSERVED
CLNT-05	Testing for CSS Injection	NO ISSUE OBSERVED
CLNT-06	Testing for Client-side Resource Manipulation	NO ISSUE OBSERVED
CLNT-07	Testing Cross Origin Resource Sharing	NO ISSUE OBSERVED
CLNT-08	Testing for Cross Site Flashing *No Flash content (technology retired)*	NOT APPLICABLE

Test ID	Test case	Result
CLNT-09	Testing for Clickjacking	NO ISSUE OBSERVED
CLNT-10	Testing WebSockets	NO ISSUE OBSERVED
CLNT-11	Testing Web Messaging	NO ISSUE OBSERVED
CLNT-12	Testing Browser Storage	NO ISSUE OBSERVED
CLNT-13	Testing for Cross Site Script Inclusion	NO ISSUE OBSERVED
API Testing 1 test case		
APIT-01	Testing GraphQL *API is REST-only; no GraphQL endpoint exposed*	NOT APPLICABLE

Results reflect the in-scope systems as observed during the testing window, using the techniques permitted by the rules of engagement. A result of "No issue observed" is not a warranty that the system is free of vulnerabilities; security posture can change as systems are modified.

Ready to find your weaknesses — and fix them?

Elite human-led penetration testing, backed by the American Dream Cyber Scanner, for web applications, APIs, cloud, networks, mobile and AI systems.

E-MAIL info@americandream-cybersecurity.com

WEBSITE americandream-cybersecurity.com

BOOK **A free consultation — get a fixed price**

UNDER ATTACK? **24/7 Incident Response**

<2h	<24h	<48h
INITIAL RESPONSE SLA	FIRST FINDINGS REPORTED	RAPID PROJECT KICK-OFF

© 2026 American Dream Security Labs Inc. All rights reserved. Designed for high-assurance enterprise defense.