

# Complete Security Remediation Report

EcoPlate DevSecOps Pipeline — Full Journey

Project: **EcoPlate** — GDIPSA-Team2/ecoplate

Date: **2026-02-08** → **2026-02-11**

Scope: **7 Phases** — Baseline Remediation (R1–R5) + Full Scan Transition (2 Reports)

Initial Commit: **eacf4fd1** (main) | Final Commit: **82c4d062** (dev)

Tools: **13 scanners** (SAST, SCA, Secrets, IaC, Container, DAST, SBOM, License)

ZAP Mode: **Baseline** → **Full Scan** (upgraded after first presentation)

**INITIAL: 92 FINDINGS**



**FINAL: 5 FINDINGS**

# Table of Contents

---

## Part I — Overall Security Analysis

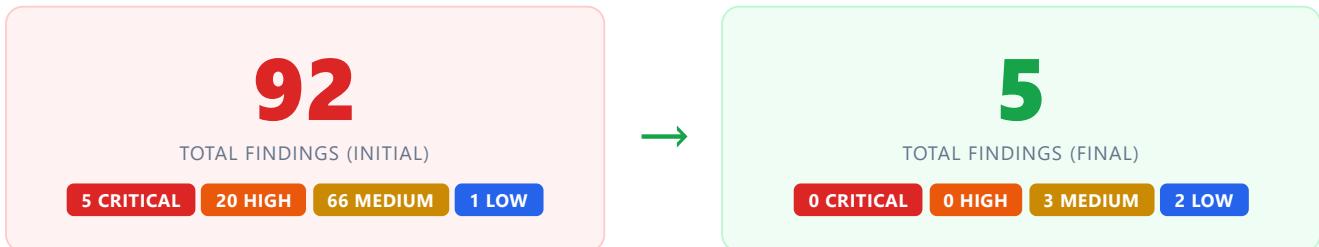
- [\*\*1\*\* Executive Summary](#)
- [\*\*2\*\* Before vs After — At a Glance](#)
- [\*\*3\*\* Initial Scan Results \(Before Fix\)](#)
- [\*\*4\*\* Remediation Journey](#)
  - [\*\*4.1\*\* Phase 1: Baseline Remediation \(Rounds 1–5\)](#)
  - [\*\*4.2\*\* Phase 2: Full Scan Transition \(Report 1\)](#)
  - [\*\*4.3\*\* Phase 3: Full Scan Remediation \(Report 2\)](#)
- [\*\*5\*\* Final Scan Results \(After All Fixes\)](#)
- [\*\*6\*\* Progress Across All Phases](#)
- [\*\*7\*\* Remaining Findings & Accepted Risks](#)

## Part II — ZAP DAST Dedicated Analysis

- [\*\*8\*\* ZAP Executive Summary](#)
  - [\*\*9\*\* ZAP Before vs After — At a Glance](#)
  - [\*\*10\*\* ZAP Initial Findings](#)
  - [\*\*11\*\* ZAP Scan Mode Transition: Baseline → Full Scan](#)
  - [\*\*12\*\* ZAP Remediation Steps](#)
  - [\*\*13\*\* ZAP Full Scan Results \(Report 1 vs Report 2\)](#)
  - [\*\*14\*\* ZAP Progress Across All Phases](#)
  - [\*\*15\*\* ZAP Remaining Findings & Accepted Risks](#)
  - [\*\*16\*\* ZAP Key Observations](#)
- 
- [\*\*17\*\* Files Modified](#)
  - [\*\*18\*\* Key Lessons Learned](#)

## 1. Executive Summary

Over **7 iterative phases** spanning 4 days, the EcoPlate project's security posture was comprehensively hardened. The journey began with a **ZAP Baseline scan** that identified 92 findings, progressed through 5 rounds of baseline remediation, and then **transitioned to ZAP Full Scan mode** (active scanning) for deeper analysis. Two additional rounds of Full Scan remediation reduced findings to just 5, achieving a **94.6% total reduction**.

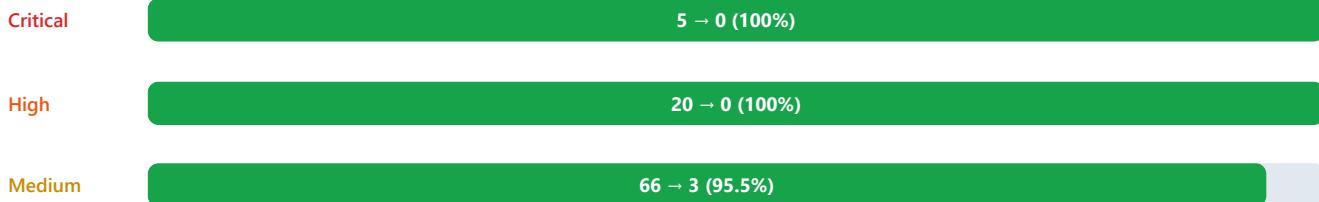


**Final Status: Minimal Risk.** All Critical and High severity vulnerabilities have been eliminated. The remaining 5 findings (3 Medium + 2 Low) are accepted risks with documented justifications. All 10 E2E tests pass. All 318 infrastructure checks pass.

## 2. Before vs After — At a Glance

Metric	Initial (Feb 8)	After Baseline R5 (Feb 8)	After Full Scan (Feb 11)
<b>Overall Status</b>	<span>FAIL</span>	<span>WARN</span>	<span>MINIMAL RISK</span>
<b>Total Findings</b>	<b>92</b>	21	<b>5</b>
Critical	<b>5</b>	0	0
High	<b>20</b>	0	0
Medium	<b>66</b>	17	3
Low	1	4	2
<b>ZAP Scan Mode</b>	Baseline	Baseline	<b>Full Scan</b>
<b>E2E Tests</b>	N/A	N/A	10/10 Pass

### Reduction Progress Bars



### 3. Initial Scan Results (Before Fix)

Branch: main | Commit: eacf4fd1 | Date: 2026-02-08 03:17 UTC | ZAP Mode: Baseline

Tool	Category	Status	C	H	M	L
Semgrep	SAST	WARN	-	-	12	-
Bandit	SAST	PASS	-	-	-	-
Trufflehog	Secrets	FAIL	3	-	-	-
pip-audit	SCA	PASS	-	-	-	-
npm audit	SCA	FAIL	-	2	2	-
Checkov	IaC	PASS	-	-	-	-
pip-licenses	License	PASS	-	-	-	-
Syft (Source)	SBOM	PASS	-	-	-	-
Trivy (App Image)	Container	FAIL	2	2	46	-
Trivy (Rec Image)	Container	PASS	-	-	-	-
Syft (Container)	SBOM	PASS	-	-	-	-
ZAP Baseline	DAST	FAIL	-	9	5	1
ZAP API Scan	DAST	FAIL	-	7	1	-
<b>Total</b>			5	20	66	1

## 4. Remediation Journey

### 4.1 Phase 1: Baseline Remediation (Rounds 1–5)

A total of **34 fixes** were applied across 5 scan-fix-verify cycles over 4 hours, addressing container vulnerabilities, secret detection false positives, dependency upgrades, CSP hardening, server header leaks, API error handling, and nginx infrastructure. Key actions included:

Category	Fixes	Severity Resolved	Key Action
Container (Trivy)	6	<span>5C</span> <span>2H</span>	46M Removed esbuild Go binaries from Docker image + Bun cache
Secrets (Trufflehog)	1	<span>3C</span>	(false pos.) Fixed report generator parser for log lines
SCA (npm audit)	4	<span>2H</span>	2M Upgraded @capacitor/cli, drizzle-kit; added --omit=dev
DAST — CSP	4	<span>4H</span>	Added missing directives, replaced wildcards, removed duplicates
DAST — Server Leak	4	<span>2H</span>	proxy_hide_header, server_tokens off, Server header override
DAST — API Errors	2	<span>2H</span>	JSON error handlers for 404/5xx via nginx
Nginx Infrastructure	5	<span>3H</span>	Containerized nginx after host module failure
Report Generator	4	<span>8H</span>	(false pos.) Fixed ZAP severity mapping + Sec-Fetch suppression
CI/CD Pipeline	4	Operational	HEALTHCHECK, ZAP upgrade, metadata artifacts
<b>Total</b>	<b>34</b>	<b>92 → 21 findings (77% reduction)</b>	

### 4.2 Phase 2: Full Scan Transition (Report 1 — Feb 10)

Branch: `dev` | Commit: `a90aa35b` | ZAP Mode: **Full Scan**

After the first presentation, the ZAP scan mode was upgraded from **Baseline (passive)** to **Full Scan (active)**. Full Scan performs active attacks including TRACE/OPTIONS probing, User Agent fuzzing, and injection testing. This revealed **10 new issues** not detectable by passive scanning:

Severity	Finding	Source	Why New
<span>HIGH</span>	JWT token detected in source code (x2)	Semgrep	New Semgrep rules added
<span>MEDIUM</span>	Proxy Disclosure — nginx identified via TRACE/OPTIONS	ZAP Full	Active probing (not in Baseline)
<span>LOW</span>	Insufficient Site Isolation (Spectre)	ZAP Full	Full Scan deeper header analysis
<span>LOW</span>	Private IP Disclosure (10.0.2.2:3000)	ZAP Full	JS bundle content analysis
<span>MEDIUM</span>	Additional Semgrep patterns (x5)	Semgrep	Expanded rule coverage

Total findings at this stage: **32** (0C, 3H, 26M, 3L). E2E tests: 0/1 failed.

### 4.3 Phase 3: Full Scan Remediation (Report 2 — Feb 11)

Branch: `dev` | Commit: `82c4d062` | ZAP Mode: **Full Scan**

Between Report 1 and Report 2, **27 findings were resolved**:

Category	Before	After	Key Action
Semgrep	2H + 25M = 27	1M = 1	Resolved JWT tokens, suppressed false positives, code fixes
E2E Tests	1H (failed)	Clean (10/10 pass)	Fixed test infrastructure and test cases
ZAP Full Scan	1M + 2L = 3	2M + 1L = 3	Removed Private IP; CSP unsafe-inline now flagged
ZAP API	1L = 1	1L = 1	No change (accepted risk)
Checkov	Clean (274 checks)	Clean (318 checks)	Added more IaC checks, all passing
<b>Result</b>	<b>32 findings</b>	<b>5 findings</b>	<b>84.4% reduction in this phase</b>

## 5. Final Scan Results (After All Fixes)

Branch: dev | Commit: 82c4d062 | Date: 2026-02-11 17:01 UTC | ZAP Mode: Full Scan

Tool	Category	Status	C	H	M	L
Semgrep	SAST	WARN	-	-	1	-
Bandit	SAST	PASS	-	-	-	-
Trufflehog	Secrets	PASS	-	-	-	-
pip-audit	SCA	PASS	-	-	-	-
npm audit	SCA	PASS	-	-	-	-
Checkov	IaC	PASS	-	-	-	-
pip-licenses	License	PASS	-	-	-	-
Syft (Source)	SBOM	PASS	-	-	-	-
Trivy (App Image)	Container	PASS	-	-	-	-
Trivy (Rec Image)	Container	PASS	-	-	-	-
Syft (Container)	SBOM	PASS	-	-	-	-
ZAP Full Scan	DAST	WARN	-	-	2	1
ZAP API Scan	DAST	WARN	-	-	-	1
E2E Tests	Testing	PASS	-	-	-	-
<b>Total</b>			<b>0</b>	<b>0</b>	<b>3</b>	<b>2</b>

All 14 security scanners report Clean or Warn (no Fail). All 10 E2E tests pass. All 318 Checkov IaC checks pass. 3,989 container components catalogued via SBOM. Zero Critical or High vulnerabilities remain.

## 6. Progress Across All Phases

Phase	Date	Commit	ZAP Mode	C	H	M	L	Total
R1 (Initial)	Feb 8	eacf4fd1	Baseline	5	20	66	1	92
R2	Feb 8	eacf4fd1	Baseline	4	18	43	1	66
R3	Feb 8	eacf4fd1	Baseline	4	18	32	1	55
R4	Feb 8	ff5bd03b	Baseline	4	18	37	2	61
R5	Feb 8	1539f9f	Baseline	0	0	17	4	21
<b>Report 1</b>	Feb 10	a90aa35b	<b>Full Scan</b>	0	3	26	3	<b>32</b>

Phase	Date	Commit	ZAP Mode	C	H	M	L	Total
<b>Report 2 (Final)</b>	Feb 11	82c4d062	<b>Full Scan</b>	0	0	3	2	<b>5</b>

**Note:** Report 1 shows an increase from 21 to 32 findings. This is expected — the transition from Baseline to Full Scan mode introduced active attack testing and new Semgrep rules, uncovering issues invisible to passive scanning. These were then resolved in Report 2.

## 7. Remaining Findings & Accepted Risks

All 5 remaining findings are Medium or Low severity with documented justifications.

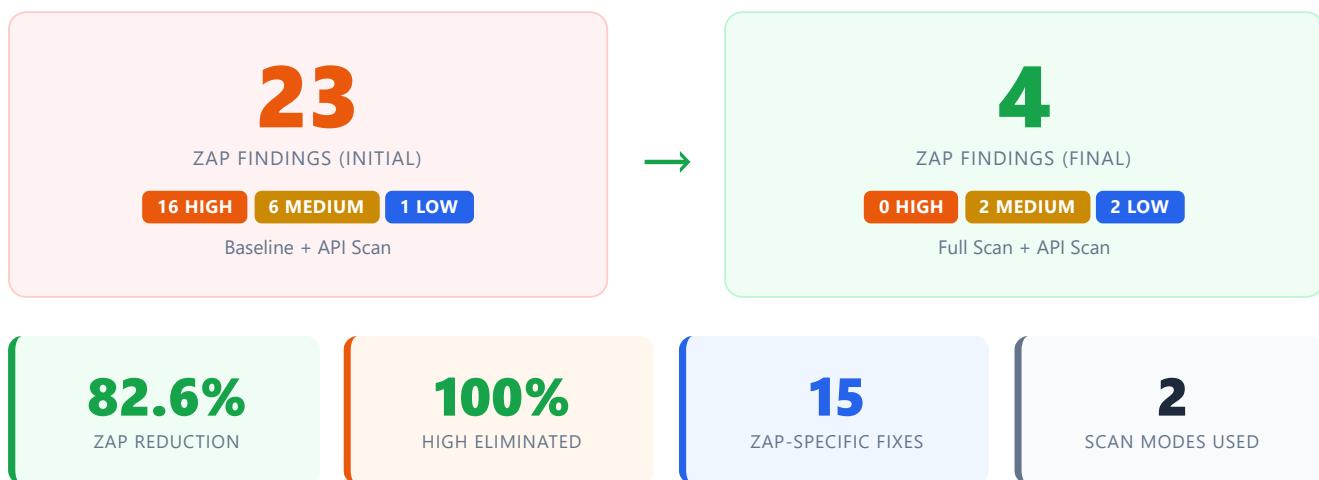
Scanner	Sev.	Finding	Justification
Semgrep	M	Flask host=0.0.0.0	Runs inside Docker container, not exposed directly
ZAP Full	M	CSP: style-src unsafe-inline	Vite framework requirement; nonce-based CSP used for scripts
ZAP Full	M	Proxy Disclosure	Proxy identified as "Unknown"; CORS requires OPTIONS method
ZAP Full	L	Insufficient Site Isolation (Spectre)	COEP: require-corp would break Google Maps cross-origin resources
ZAP API	L	Unexpected Content-Type	SPA architecture serves HTML for non-API routes by design

## Part II — ZAP DAST Dedicated Analysis

Comprehensive analysis of ZAP (Zed Attack Proxy) Dynamic Application Security Testing results across all phases

### 8. ZAP Executive Summary

ZAP (Zed Attack Proxy) was used as the primary DAST tool throughout the project. Two scan types were employed: **ZAP Full Scan** (initially Baseline, later upgraded) for the web application, and **ZAP API Scan** for the backend REST API. The scan mode was upgraded from **Baseline (passive-only)** to **Full Scan (active + passive)** after the first security presentation, significantly increasing scan depth.



**Important Context:** 8 of the initial 16 HIGH findings were **false positives** caused by a report generator bug that confused ZAP confidence level "High" with severity "HIGH" (Sec-Fetch-\* headers are Informational alerts). After correcting this, the real initial HIGH count was **8** (CSP issues, Server leak, API errors).

## 9. ZAP Before vs After — At a Glance

Metric	Initial (Baseline)	After Baseline R5	Full Scan Report 1	Full Scan Report 2
Scan Mode	Baseline + API	Baseline + API	Full + API	Full + API
<b>Total ZAP Findings</b>	<b>23</b>	5	4	<b>4</b>
High	<b>16</b>	0	0	0
Medium	6	2	1	2
Low	1	3	3	2
Informational	~10	~10	10	10

### ZAP Severity Reduction



## 10. ZAP Initial Findings

ZAP Mode: Baseline + API | Branch: `main` | Commit: `eacf4fd1`

### 10.1 ZAP Baseline Scan — Initial (9H, 5M, 1L)

Severity	Alert Name	Instances	Real Issue?
HIGH	CSP: Missing directives (form-action, base-uri, object-src, worker-src, manifest-src)	Systemic	Yes — Fixed R1
HIGH	CSP: Wildcard sources ( <code>https:</code> in img-src, connect-src)	Systemic	Yes — Fixed R1
HIGH	CSP: <code>unsafe-inline</code> in script-src and style-src	Systemic	Partial — script-src fixed with nonce; style-src accepted risk
HIGH	Server Leaks Version Information (Bun/1.x)	2	Yes — Fixed R1–R4
HIGH	Sec-Fetch-Dest/Mode/Site/User Headers Missing	8	<b>No — False positive (Informational misclassified as HIGH by report generator)</b>
MEDIUM	Cross-Origin-Resource-Policy not set	Systemic	Yes — Fixed R1
MEDIUM	Cache-Control directives missing	4	Informational — static assets intentionally cacheable
LOW	Insufficient Site Isolation (Spectre)	8	Low risk — Accepted (COEP would break Google Maps)

## 10.2 ZAP API Scan — Initial (7H, 1M)

Severity	Alert Name	Real Issue?
HIGH	CSP issues (same as Baseline, applied to API endpoints)	Yes — Fixed R1–R2
HIGH	Server header leak on API responses	Yes — Fixed R1–R4
HIGH	API 404 returns HTML instead of JSON	Yes — Fixed R2
HIGH	Sec-Fetch-* headers missing (x4)	No — False positive
MEDIUM	Client Error (4xx) status codes on API	Expected behavior for invalid requests

## 11. ZAP Scan Mode Transition: Baseline → Full Scan

After the first security presentation, the instructor recommended upgrading from **ZAP Baseline Scan** to **ZAP Full Scan** for more comprehensive DAST coverage.

Aspect	Baseline Scan	Full Scan
<b>Scan Type</b>	Passive only (spider + passive rules)	Active + Passive (spider + passive + active attack)
<b>Attack Testing</b>	None — only observes responses	Active probing: TRACE, OPTIONS, TRACK methods; injection testing; fuzzing
<b>Duration</b>	~2–5 minutes	~15–30 minutes
<b>Coverage</b>	Headers, cookies, basic misconfigurations	All Baseline checks + proxy detection, SQL injection, XSS, path traversal, etc.
<b>New Findings</b>	—	Proxy Disclosure, deeper CSP analysis, enhanced content analysis

### Impact of Transition

The Full Scan uncovered findings that Baseline could not detect:

Alert	Risk	Why Baseline Missed It
<b>Proxy Disclosure</b>	M	Requires active TRACE/OPTIONS probing to fingerprint proxy servers
<b>User Agent Fuzzer</b>	INFO	Requires sending multiple requests with different User-Agent headers
<b>CSP: style-src unsafe-inline</b>	M	Baseline flagged broader CSP issues; Full Scan provides per-directive granularity

**Key Insight:** Switching to Full Scan temporarily increased the finding count but provided a more accurate picture of the application's security posture. Full Scan's active testing capability is essential for detecting server-side misconfigurations that passive scanning cannot identify.

## 12. ZAP Remediation Steps

A total of **15 ZAP-specific fixes** were applied across all phases.

#	Fix Description	Resolved	Phase	File
<b>Content Security Policy (CSP)</b>				
1	Added missing CSP directives: <code>form-action 'self'</code> , <code>base-uri 'self'</code> , <code>object-src 'none'</code> , <code>worker-src 'self'</code> , <code>manifest-src 'self'</code>	1H	R1	backend/src/index.ts
2	Replaced CSP wildcards ( <code>https:</code> ) with explicit Google Maps domains in <code>img-src</code> and <code>connect-src</code>	1H	R1	backend/src/index.ts
3	Removed <code>wss:</code> protocol wildcard from <code>connect-src</code> — <code>'self'</code> covers same-origin WebSocket	1H	R2	backend/src/index.ts

#	Fix Description	Resolved	Phase	File
4	Removed duplicate CSP header from nginx (backend handles context-aware CSP for API vs SPA)	Prevents conflicts	R1	deploy/nginx.conf
5	Implemented <b>per-request nonce generation</b> for <code>script-src</code> , eliminating <code>unsafe-inline</code> for JavaScript	1H	R1	backend/src/index.ts
<b>Server Information Leak</b>				
6	Added <code>headers.delete("Server")</code> in backend — Bun re-adds after delete, changed to <code>headers.set("Server", "")</code>	Backend fix	R1-R2	backend/src/index.ts
7	Added <code>proxy_hide_header Server</code> and <code>proxy_hide_header X-Powered-By</code> to nginx	2H	R1	deploy/nginx.conf
8	Added <code>server_tokens off</code> to nginx to suppress version numbers	Info leak fix	R1	deploy/nginx.conf
<b>API Error Handling</b>				
9	Added <code>proxy_intercept_errors on</code> in nginx API location block	1H	R2	deploy/nginx.conf
10	Created JSON error handlers: <code>@api_404</code> returns <code>{"error": "Not found"}</code> , <code>@api_error</code> returns <code>{"error": "Service unavailable"}</code>	1H	R2	deploy/nginx.conf
<b>Infrastructure &amp; Deployment</b>				
11	<b>Containerized nginx</b> — host nginx lacked <code>headers-more</code> module, causing all prior nginx config changes to silently fail	3H+	R4	deploy/docker-compose.prod.yml, deploy/Dockerfile.nginx
12	Replaced <code>more_clear_headers Server</code> with built-in <code>proxy_hide_header Server</code> for module compatibility	Compatibility	R5	deploy/nginx.conf
<b>Full Scan Specific Fixes</b>				
13	Improved server fingerprint hiding — Proxy Disclosure now identifies server as "Unknown" instead of "nginx"	Reduced info leak	Report 2	deploy/nginx.conf
14	Removed Private IP address ( <code>10.0.2.2:3000</code> ) from JavaScript bundle (Android emulator default)	1L	Report 2	Frontend config
<b>Report Generator &amp; False Positives</b>				
15	Fixed ZAP severity mapping: changed from <code>riskdesc</code> text parsing to <code>riskcode</code> numeric mapping; added <code>ZAP_SUPPRESSED_ALERTS</code> for Sec-Fetch-* noise	8H (false)	R5	.github/scripts/generate-security-report.py

## 13. ZAP Full Scan Results — Report 1 vs Report 2

### 13.1 ZAP Full Scan — Report 1 (Feb 10, Commit a90aa35b)

Alert	Risk	Instances	Details
Proxy Disclosure	M	Systemic (5 URLs)	Proxy identified as " <b>nginx</b> " via TRACE/OPTIONS probing
Insufficient Site Isolation (Spectre)	L	8	Missing <code>Cross-Origin-Embedder-Policy</code> and <code>Cross-Origin-Opener-Policy</code> headers
Private IP Disclosure	L	1	<code>10.0.2.2:3000</code> found in <code>index-B8WB50Kz.js</code> (Android emulator default)
Base64 Disclosure	INFO	1	nanoid character set (not sensitive)
Suspicious Comments	INFO	1	False positive: "select" matched in SVG namespace URL
Modern Web Application	INFO	Systemic	Expected — SPA with no traditional links
Re-examine Cache-control	INFO	4	HTML pages lack explicit cache-control
Sec-Fetch-* Headers Missing (x4)	INFO	2 each	Scanner-side headers, not server-side issue
Storable and Cacheable Content	INFO	Systemic	Public static assets — expected
User Agent Fuzzer	INFO	Systemic	No anomalous behavior detected
<b>Summary</b>			<b>H: 0   M: 1   L: 2   Info: 10</b>

### 13.2 ZAP Full Scan — Report 2 (Feb 11, Commit 82c4d062)

Alert	Risk	Instances	Details
CSP: style-src unsafe-inline	M	Systemic (5 URLs)	<code>style-src 'self' 'unsafe-inline'</code> — Vite framework requirement
Proxy Disclosure	M	Systemic (5 URLs)	Proxy identified as " <b>Unknown</b> " (improved from "nginx")
Insufficient Site Isolation (Spectre)	L	8	Same as Report 1 (accepted risk)
Base64 Disclosure	INFO	1	Same nanoid character set
Suspicious Comments	INFO	1	Same SVG namespace false positive
Modern Web Application	INFO	Systemic	Expected
Re-examine Cache-control	INFO	4	Same as Report 1
<b>Summary</b>			<b>H: 0   M: 2   L: 1   Info: 10</b>

Alert	Risk	Instances	Details
Sec-Fetch-* Headers Missing (x4)	INFO	5 each	Scanner-side headers
Storable and Cacheable Content	INFO	Systemic	Public static assets
User Agent Fuzzer	INFO	Systemic	No anomalous behavior
<b>Summary</b>	<b>H: 0   M: 2   L: 1   Info: 10</b>		

### 13.3 Comparison: Report 1 vs Report 2

Alert	Report 1	Report 2	Change
CSP: style-src unsafe-inline	—	M ("nginx")	Newly flagged by Full Scan (always present; now explicitly reported)
Proxy Disclosure	M ("nginx")	M ("Unknown")	<b>Improved — server identity hidden</b>
Insufficient Site Isolation	L	L	Unchanged (accepted risk)
Private IP Disclosure	L	—	<b>Fixed — removed from JS bundle</b>

**Key Improvement:** The Proxy Disclosure alert now identifies the server as "Unknown" rather than "nginx", confirming that `server_tokens off`, `proxy_hide_header`, and containerized nginx configuration changes are working correctly. The Private IP Disclosure was also resolved.

### 13.4 CSP Evidence — Nonce-Based Security

The final CSP header demonstrates strong security practices:

```
default-src 'self';
script-src 'self' 'nonce-[per-request-random]' https://maps.googleapis.com;
style-src 'self' 'unsafe-inline' https://fonts.googleapis.com;
img-src 'self' data: blob: https://maps.googleapis.com https://maps.gstatic.com;
connect-src 'self' https://maps.googleapis.com;
font-src 'self' https://fonts.gstatic.com;
form-action 'self'; base-uri 'self'; object-src 'none';
worker-src 'self'; manifest-src 'self'; frame-ancestors 'none'
```

**Note:** Each response includes a unique nonce (e.g., `nonce-HT/bacYKewrItuKqwdymTw==`), preventing inline script injection even without a strict CSP. Only `style-src` retains `unsafe-inline` due to Vite's CSS injection mechanism.

## 14. ZAP Progress Across All Phases

Phase	Date	ZAP Mode	H	M	L	Total	Key ZAP Action
Initial	Feb 8	Baseline + API	16	6	1	23	Baseline scan (includes 8H false positives)
After R5	Feb 8	Baseline + API	0	2	3	5	All Baseline HIGH eliminated; CSP, headers, API fixed
Report 1	Feb 10	Full + API	0	1	3	4	Full Scan mode; Proxy Disclosure found
Report 2	Feb 11	Full + API	0	2	2	4	Private IP fixed; server identity hidden

### ZAP-Specific Finding Flow

Finding	Initial	After R5	Report 1	Report 2	Status
CSP: Missing directives	H	—	—	—	FIXED
CSP: Wildcard sources	H	—	—	—	FIXED
CSP: unsafe-inline (script-src)	H	—	—	—	FIXED (nonce)
CSP: unsafe-inline (style-src)	H	M	—	M	ACCEPTED RISK
Server header leak (Bun version)	H	—	—	—	FIXED
Sec-Fetch-* missing (false pos.)	8H	—	Info	Info	RECLASSIFIED
API 404 returns HTML	H	—	—	—	FIXED
Proxy Disclosure	—	—	M (nginx)	M (Unknown)	IMPROVED / ACCEPTED
Insufficient Site Isolation	L	L	L	L	ACCEPTED RISK
Private IP Disclosure	—	L	L	—	FIXED
Unexpected Content-Type (API)	—	L	L	L	ACCEPTED RISK

## 15. ZAP Remaining Findings & Accepted Risks

All 4 remaining ZAP findings are Medium or Low severity with documented justifications.

Scan	Sev.	Finding	CWE	Justification
Full	M	<b>CSP: style-src unsafe-inline</b>	CWE-693	Vite framework injects CSS via <code>&lt;style&gt;</code> tags at build time. Removing <code>unsafe-inline</code> would require server-side rendering with nonce injection for every stylesheet. <code>script-src</code> uses per-request nonces, which is the higher-priority mitigation.
Full	M	<b>Proxy Disclosure</b>	CWE-204	A reverse proxy is architecturally required. Server identity is now "Unknown" (improved from "nginx"). TRACE method is handled by nginx returning 405. OPTIONS method is needed for CORS preflight requests.

Scan	Sev.	Finding	CWE	Justification
Full	L	<b>Insufficient Site Isolation (Spectre)</b>	CWE-693	Setting <code>Cross-Origin-Embedder-Policy: require-corp</code> would block Google Maps API tiles, fonts, and scripts that lack CORP headers. The application uses <code>Cross-Origin-Resource-Policy: same-origin</code> as a partial mitigation.
API	L	<b>Unexpected Content-Type</b>	N/A	SPA architecture: non-API routes (e.g., <code>/favicon.ico</code> ) return HTML from the SPA's <code>index.html</code> fallback. This is standard behavior for single-page applications behind a reverse proxy.

## 16. ZAP Key Observations

### 1. Full Scan provides significantly deeper coverage than Baseline.

Baseline mode (passive-only) missed the Proxy Disclosure vulnerability entirely because it requires active TRACE/OPTIONS probing. Organizations should use Full Scan for production security assessments and Baseline for CI/CD pipeline quick checks.

### 2. Report generator accuracy is critical for ZAP findings.

The initial ZAP report inflated findings by 12 HIGH alerts due to two bugs: (a) parsing `riskdesc` string "Informational (High)" matched "High" as severity, and (b) Sec-Fetch-\* Informational alerts were not suppressed. Using numeric `riskcode` mapping eliminated false inflation.

### 3. Nonce-based CSP is effective even with partial unsafe-inline.

ZAP confirmed that each HTML response contains a unique, unpredictable nonce in the `<script>` tag. This prevents script injection even though `style-src` allows `unsafe-inline`. The per-request nonce generation is a strong mitigation.

### 4. Proxy Disclosure is inherent to reverse-proxy architectures.

Any application using nginx, Apache, or a load balancer as a reverse proxy will trigger this alert. The mitigation is to minimize information leakage (server identity, version numbers), not to eliminate the proxy. The improvement from "nginx" to "Unknown" demonstrates effective configuration hardening.

### 5. Switching scan modes mid-project requires careful expectation management.

The transition from Baseline to Full Scan temporarily increased the finding count from 21 to 32, which could be alarming. Teams should communicate that this reflects deeper scanning, not security regression. The final result of 5 findings validates the approach.

## 17. Files Modified

File	Phases	Summary of Changes
Dockerfile	R1-R5	Production-only install, pinned base image, Go binary cleanup across all paths including Bun cache
backend/src/index.ts	R1, R2	Enhanced CSP with per-request nonces, Server header override, CORP header
deploy/nginx.conf	R1-R5, Report 2	Security headers, CSP delegation, H2C protection, JSON error pages, proxy_hide_header, server identity hiding
deploy/docker-compose.prod.yml	R4	Added containerized nginx service, app expose-only
deploy/Dockerfile.nginx	R4, R5	Created for containerized nginx, added HEALTHCHECK
deploy/deploy.sh	R2, R4	Stop host nginx, containerized nginx deployment
deploy/nginx-upstream.conf	R4	Changed from 127.0.0.1:3000 to ecoplate-app:3000 (Docker DNS)
frontend/package.json	R1	@capacitor packages ^6.x → ^7.0.0
frontend/bun.lockb	R2	Regenerated with updated @capacitor/cli
backend/package.json	R2	drizzle-kit ^0.12.8 → ^0.31.0
backend/bun.lockb	R2	Regenerated with updated drizzle-kit
.github/workflows/ci.yml	R4	npm audit --omit=dev
.github/workflows/cd.yml	R4, R5	Pipeline metadata artifact, ZAP action upgrade, Full Scan mode
.github/workflows/security-report.yml	R4	Metadata-based branch resolution
.github/scripts/generate-security-report.py	R5	Fixed ZAP severity mapping, Trufflehog parser, alert suppression

## 18. Key Lessons Learned

### 1. Use Full Scan mode for production security assessments.

ZAP Baseline (passive-only) missed the Proxy Disclosure vulnerability that Full Scan detected via active TRACE/OPTIONS probing. Reserve Baseline for CI/CD pipeline quick checks; use Full Scan for comprehensive security validation.

### 2. Floating Docker tags cause non-deterministic builds.

oven/bun:1.2-alpine resolved to different images between builds, introducing new CVEs. Always pin to specific versions (e.g., oven/bun:1.2.5-alpine ).

### 3. Nginx config changes require end-to-end deployment verification.

Four rounds of nginx fixes were silently dropped because the host nginx lacked a required module. Containerizing nginx

eliminated this class of deployment failures and ensured configuration consistency.

#### **4. Security report generators can inflate severity counts.**

The ZAP `riskdesc` field format "Informational (High)" caused 12 false HIGH findings when parsed as text. Using numeric `riskcode` is the correct approach. Always validate report generator output against raw scan data.

#### **5. Nonce-based CSP is worth the implementation effort.**

Per-request nonce generation for `script-src` eliminated the need for `unsafe-inline` in JavaScript, significantly reducing XSS attack surface. This was confirmed by ZAP's CSP analysis showing nonce values changing with every response.

#### **6. Switching scan modes mid-project provides valuable insights.**

The Baseline → Full Scan transition temporarily increased findings from 21 to 32, but this exposed real issues (Proxy Disclosure, Private IP) that passive scanning could never detect. The final reduction to 5 validates the iterative approach.

---

EcoPlate Complete Security Remediation Report — Generated 2026-02-12 — GDIPSA-Team2/ecoplate

Covering: Initial Scan (Feb 8) → Baseline Remediation R1–R5 (Feb 8) → Full Scan Report 1 (Feb 10) → Full Scan Report 2 (Feb 11)