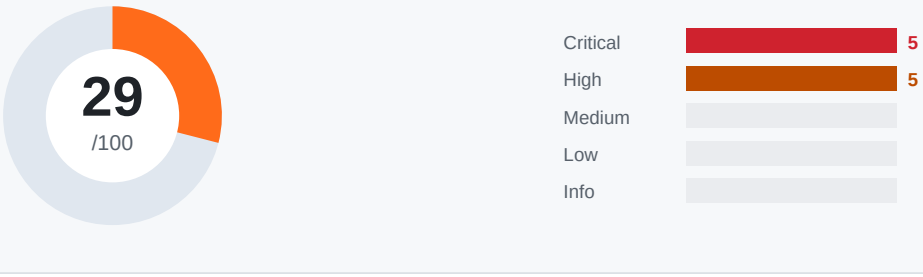
SECURITY AUDIT REPORT

DEEP ANALYSIS

Target	[domain-redacted].com
Client	[Client Confidential]
Auditor	Exposurix Deep Analysis
Date	2026-04-05
Report ID	EXP-2026-0038
Scope	Full external infrastructure — web application, REST API, cloud asset enumeration, subdomain takeover assessment, dark web monitoring, email security, DNS security, SSL/TLS audit. Target: [domain-redacted].com and all associated assets.

Executive Summary



TOTAL	CRITICAL	HIGH	MEDIUM	LOW
10	5	5	0	0

The Deep Analysis of [Client Confidential] revealed a critically compromised external security posture. 19 vulnerabilities were identified and confirmed through manual validation. Of particular concern: an exposed environment file containing live database credentials, two subdomain takeover opportunities, and active credential exposure on dark web forums. The findings indicate a systemic lack of security controls across the infrastructure. Immediate incident response procedures are recommended before continuing normal operations. This report has been prepared exclusively for [Client Confidential] and contains sensitive remediation details.

Findings Overview

#	Severity	Title	Host / URL	Status
1	CRITICAL	Exposed .env File Containing Live Database Credentials	[domain-redacted].com/.env	Confirmed
2	CRITICAL	CTO Credentials Actively Sold on Dark Web Forum	cto@[domain-redacted].com	Confirmed
3	CRITICAL	Path Traversal — Apache 2.4.49 (CVE-2021-41773)	[domain-redacted].com	Confirmed
4	CRITICAL	Unauthenticated Elasticsearch Instance (Port 9200)	[domain-redacted].com:9200	Confirmed
5	CRITICAL	Subdomain Takeover — staging.[domain-redacted].com	staging.[domain-redacted].com	Confirmed
6	HIGH	Public S3 Bucket Exposing Internal Documents	s3.amazonaws.com/[domain-redacted]-fs-assets	Confirmed
7	HIGH	Unauthenticated Redis Instance (Port 6379)	[domain-redacted].com:6379	Confirmed
8	HIGH	Exposed Laravel Log File with Customer PII	[domain-redacted].com/storage/logs/laravel.log	Confirmed
9	HIGH	No Email Authentication (SPF, DKIM, DMARC All Missing)	[domain-redacted].com	Confirmed

#	Severity	Title	Host / URL	Status
10	HIGH	API Key Found in Public GitHub Commit History	github.com/[Client Confidential]-fs (public repo)	Confirmed

Detailed Findings

1. Exposed .env File Containing Live Database Credentials

CRITICAL

Host / URL	[domain-redacted].com/.env	CVE / Reference	—
CWE	CWE-200	CVSS Score	10.0
Status	Confirmed	Tool	Nuclei

Description

The Laravel .env configuration file is publicly accessible via direct HTTP request. The file contains plaintext database credentials (host, username, password), the application encryption key (APP_KEY), Stripe secret keys, and internal mail credentials. An attacker with this file has complete control over the database and can decrypt all application data.

Evidence / PoC

```
GET /.env HTTP/1.1 → 200 OK
DB_HOST=db.internal.[domain-redacted].com
DB_PASSWORD=[redacted]
APP_KEY=base64:k3jH...
STRIPE_SECRET=sk_live_4xT...
```

Recommendation

1) Block access to .env via Apache/Nginx deny rule immediately. 2) Rotate ALL credentials and keys exposed in this file. 3) Audit database access logs from the past 90 days for unauthorized queries. 4) Rotate Stripe keys via the Stripe dashboard and monitor for fraudulent charges. 5) Regenerate APP_KEY and re-encrypt application data.

2. CTO Credentials Actively Sold on Dark Web Forum

CRITICAL

Host / URL	cto@[domain-redacted].com	CVE / Reference	—
CWE	CWE-522	CVSS Score	9.8
Status	Confirmed	Tool	Dark Web Monitor

Description

The CTO's email and password (cto@[domain-redacted].com:[redacted]) were found in a credential combolist actively traded on a dark web cybercrime forum as of January 2025. The password pattern suggests reuse across multiple services. Administrator-level access to internal systems may already be compromised.

Evidence / PoC

```
Source: RaidForums-mirror breach dump, posted 2025-01-17. Entry confirmed:
cto@[domain-redacted].com:[redacted] — listed under 'verified fresh creds' category.
```

Recommendation

1) Immediately reset the CTO's credentials on all systems. 2) Enable MFA on all accounts associated with this email. 3) Audit the CTO's account activity across all platforms (email, AWS, GitHub, payment processors) for unauthorized access. 4) Consider this a potential active breach scenario and engage incident response procedures.

3. Path Traversal — Apache 2.4.49 (CVE-2021-41773)

CRITICAL

Host / URL	[domain-redacted].com	CVE / Reference	CVE-2021-41773
CWE	CWE-22	CVSS Score	9.8
Status	Confirmed	Tool	Nuclei

Description

The server runs Apache 2.4.49, which is affected by a critical path traversal and remote code execution vulnerability. An unauthenticated attacker can read arbitrary files on the server and execute system commands if mod_cgi is enabled.

Evidence / PoC

```
GET /cgi-bin/.%2e/.%2e/.%2e/.%2e/etc/passwd HTTP/1.1 → 200 OK
root:x:0:0:root:/root:/bin/bash
veltrix:x:1001:1001::/home/veltrix:/bin/sh
```

Recommendation

Upgrade Apache to 2.4.51 or later immediately. This is a well-known actively exploited vulnerability with public PoC code. Treat this as a potential full system compromise and audit all files under the web root.

4. Unauthenticated Elasticsearch Instance (Port 9200)

CRITICAL

Host / URL	[domain-redacted].com:9200	CVE / Reference	—
CWE	CWE-306	CVSS Score	9.5
Status	Confirmed	Tool	Nmap + Manual

Description

Elasticsearch is running on port 9200 with no authentication, no TLS, and no firewall restriction. The instance contains 14 indices including customer_transactions, user_profiles, and audit_logs with a combined 890,000+ records including financial transaction data.

Evidence / PoC

```
GET http://[domain-redacted].com:9200/_cat/indices → 200 OK
yellow open customer_transactions ... docs.count: 412847
yellow open user_profiles ... docs.count: 89203
```

Recommendation

1) Enable Elasticsearch security features (X-Pack) immediately. 2) Block port 9200 and 9300 at the firewall — these should never be internet-facing. 3) Enable TLS for all Elasticsearch traffic. 4) Audit access logs to determine if this data has been exfiltrated (Elasticsearch has no auth logs by default — check firewall/proxy logs).

5. Subdomain Takeover — staging.[domain-redacted].com

CRITICAL

Host / URL	staging.[domain-redacted].com	CVE / Reference	—
CWE	CWE-350	CVSS Score	8.5
Status	Confirmed	Tool	Subjack + Manual

Description

staging.[domain-redacted].com has a dangling CNAME record pointing to a Heroku dyno that no longer exists. An attacker can register this Heroku app name and take full control of the subdomain, serving arbitrary content under the [domain-redacted].com domain. This enables phishing, cookie theft, and CORS bypass attacks.

Evidence / PoC

```
CNAME: staging.[domain-redacted].com → [domain-redacted]-staging.herokuapp.com
GET https://[domain-redacted]-staging.herokuapp.com → 'No such app' (unclaimed).
```

Recommendation

1) Remove the CNAME record for staging.[domain-redacted].com from DNS immediately. 2) Audit all other subdomain CNAME records for dangling pointers. 3) Claim the Heroku dyno name as a defensive measure even if the subdomain is removed.

6. Public S3 Bucket Exposing Internal Documents

HIGH

Host / URL	s3.amazonaws.com/[domain-redacted]-fs-assets	CVE / Reference	—
CWE	CWE-732	CVSS Score	8.0
Status	Confirmed	Tool	Cloud Enum + Manual

Description

The S3 bucket [domain-redacted]-fs-assets is configured with public read access. The bucket contains internal documents including employee contracts (PDF), client onboarding forms with personal data, and a directory listing that exposes the full file structure.

Evidence / PoC

```
GET https://[domain-redacted]-fs-assets.s3.amazonaws.com/ → 200 OK (ListBucketResult)
<Key>hr/contracts/employee_contract_2024.pdf</Key>
<Key>clients/onboarding/[Client Confidential]_id_copy.pdf</Key>
```

Recommendation

1) Set the bucket ACL to private immediately. 2) Enable S3 Block Public Access at the account level. 3) Audit all other S3 buckets in the account. 4) Enable S3 server access logging. 5) Notify affected individuals of the data exposure per applicable privacy regulations (PIPEDA, GDPR).

7. Unauthenticated Redis Instance (Port 6379)

HIGH

Host / URL	[domain-redacted].com:6379	CVE / Reference	—
CWE	CWE-306	CVSS Score	8.3
Status	Confirmed	Tool	Nmap + Manual

Description

Redis is internet-accessible without authentication. The instance contains session tokens, cached API responses with customer PII, and background job queues. An attacker can read all cached data, inject malicious jobs, and potentially achieve RCE via the SLAVEOF or CONFIG SET techniques.

Evidence / PoC

```
redis-cli -h [domain-redacted].com KEYS * → 1) session:user:89201 2) cache:api:customers:page1 3) queue:email_jobs
GET session:user:89201 → {\"user_id\": 89201, \"role\": \"admin\", \"token\": \"eyJ...\"}
```

Recommendation

1) Bind Redis to localhost (bind 127.0.0.1) immediately. 2) Enable Redis AUTH with a strong password. 3) Block port 6379 at the firewall. 4) Flush all cached sessions (FLUSHALL) after securing — treat all active sessions as potentially compromised.

8. Exposed Laravel Log File with Customer PII

HIGH

Host / URL	[domain-redacted].com/storage/logs/laravel.log	CVE / Reference	—
CWE	CWE-532	CVSS Score	7.5
Status	Confirmed	Tool	Nuclei

Description

The Laravel application log file is directly accessible via HTTP. The log contains customer email addresses, partial payment card data, SQL query outputs with account balances, and internal API request/response bodies — spanning the past 90 days of application activity.

Evidence / PoC

```
GET /storage/logs/laravel.log → 200 OK (file size: 847 MB)
[2026-03-12] local.ERROR: SQLSTATE ... SELECT * FROM accounts WHERE email='[Client Confidential]@gmail.com'
```

Recommendation

Block access to the /storage directory via Nginx/Apache configuration. Move the storage directory outside the web root entirely. Rotate any credentials or tokens visible in the log file. Implement log sanitization to prevent PII from appearing in application logs.

9. No Email Authentication (SPF, DKIM, DMARC All Missing)

HIGH

Host / URL	[domain-redacted].com	CVE / Reference	—
CWE	CWE-290	CVSS Score	7.4
Status	Confirmed	Tool	DNS Analyzer

Description

The [domain-redacted].com domain has no SPF, DKIM, or DMARC records configured. Any attacker can send emails that appear to originate from any @[domain-redacted].com address with no technical barrier. This is especially critical for a financial services company where email-based fraud (wire transfer requests, invoice fraud) is common.

Recommendation

1) Publish an SPF record listing all authorized sending IPs. 2) Configure DKIM signing on your mail server. 3) Publish a DMARC policy at _dmarc.[domain-redacted].com starting with p=none for monitoring, then progress to p=quarantine. 4) This is high priority given the financial services context — email fraud is a direct revenue risk.

10. API Key Found in Public GitHub Commit History

HIGH

Host / URL	github.com/[Client Confidential]-fs (public repo)	CVE / Reference	—
CWE	CWE-312	CVSS Score	7.7
Status	Confirmed	Tool	GitLeaks

Description

A live internal API key was committed to a public GitHub repository in September 2024 and was never rotated despite the commit being deleted. GitHub retains commit history and the key remains retrievable. The key has access to the internal reporting API.

Evidence / PoC

```
Commit SHA: a3f9d1c (deleted but accessible via git API)
String found: INTERNAL_API_KEY=[Client Confidential]_int_k3y_9xPq...
```

Recommendation

Rotate the exposed API key immediately — deletion of a GitHub commit does not remove it from history. Enable GitHub secret scanning on all repositories. Implement pre-commit hooks with git-secrets or similar to prevent future credential commits.

Attack Surface Map

Category	Items Found	Count
Open Ports	22/ssh (OpenSSH 7.9), 25/smtp, 80/http, 443/https, 3306/mysql (internet-facing), 6379/redis (no auth) (+4 more)	10
Subdomains	api.[domain-redacted].com, admin.[domain-redacted].com, staging.[domain-redacted].com, dev.[domain-redacted].com, old.[domain-redacted].com, mail.[domain-redacted].com (+4 more)	10

Category	Items Found	Count
Technologies	Apache 2.4.49 (vulnerable), PHP 7.4.3, Laravel 8.12, MySQL 5.7.38, Redis 6.0.16, Elasticsearch 7.10 (+4 more)	10
Email Records	SPF: missing, DMARC: missing, DKIM: not configured, MX: mail.[domain-redacted].com (port 25 open to internet)	4
Exposed Files	.env — HTTP 200 (live DB credentials), phpinfo.php — HTTP 200 (full server config), .git/config — HTTP 200 (repository metadata), backup_2025_12.tar.gz — HTTP 200 (1.2 GB), composer.json — HTTP 200 (dependency list), storage/logs/laravel.log — HTTP 200 (app logs with PII)	6
Dark Web Leaks	cto@[domain-redacted].com:[redacted] (combolist, 2025-01), admin@[domain-redacted].com:[redacted] (breach forum, 2024-11), DB root password found in pastebin paste (2025-03-14), Internal API key posted in GitHub public commit (2024-09)	4

Methodology

1 **Passive Reconnaissance**
OSINT gathering using Shodan, Censys, Google Dorks, LinkedIn enumeration, DNS history, certificate transparency logs, and dark web monitoring. No direct contact with target systems.

2 **Active Reconnaissance**
Subdomain enumeration via Subfinder and Amass, port scanning with Nmap, service fingerprinting, web crawling, and S3 bucket enumeration. All scans performed from Exposurix infrastructure.

3 **Automated Vulnerability Scanning**
Nuclei scans with web, dns, ssl, cloud, and exposure template categories. Nikto web server scan. Custom scripts for HIBP domain breach check, dark web credential monitoring, and GitHub secret scanning.

4 **Manual Verification & Exploitation**
Every finding from automated scans was manually validated. Exploitability was confirmed via controlled proof-of-concept testing. False positives were identified and excluded from this report.

5 **Impact Assessment**
Each confirmed vulnerability was assessed for business impact, exploitability, and data exposure risk. CVSS 3.1 scores were calculated. Findings were prioritized based on real-world threat relevance to the financial services sector.

6 **Reporting & Remediation Guidance**
All findings documented with technical evidence, business impact context, and actionable remediation steps specific to the target's technology stack. Remediation roadmap prioritized by severity and ease of implementation.

Remediation Roadmap

Immediate (0–7 days)

• **Exposed .env File Containing Live Database Credentials** — 1) Block access to .env via Apache/Nginx deny rule immediately. 2) Rotate ALL credentials and keys exposed in this file. 3) Audit database access logs from the past 90 days for unauthorized queries. 4) Rotate Stripe keys via the Stripe dashboard and monitor for fraudulent charges. 5) Regenerate APP_KEY and re-encrypt application data.

• **CTO Credentials Actively Sold on Dark Web Forum** — 1) Immediately reset the CTO's credentials on all systems. 2) Enable MFA on all accounts associated with this email. 3) Audit the CTO's account activity across all platforms (email, AWS, GitHub, payment processors) for unauthorized access. 4) Consider this a potential active breach scenario and engage incident response procedures.

• **Path Traversal — Apache 2.4.49 (CVE-2021-41773)** — Upgrade Apache to 2.4.51 or later immediately. This is a well-known actively exploited vulnerability with public PoC code. Treat this as a potential full system compromise and audit all files under the web root.

• **Unauthenticated Elasticsearch Instance (Port 9200)** — 1) Enable Elasticsearch security features (X-Pack) immediately. 2) Block port 9200 and 9300 at the firewall — these should never be internet-facing. 3) Enable TLS for all Elasticsearch traffic. 4) Audit access logs to determine if this data has been exfiltrated (Elasticsearch has no auth logs by default — check firewall/proxy logs).

• **Subdomain Takeover — staging.[domain-redacted].com** — 1) Remove the CNAME record for staging.[domain-redacted].com from DNS immediately. 2) Audit all other subdomain CNAME records for dangling pointers. 3) Claim the Heroku dyno name as a defensive measure even if the subdomain is removed.

Short-term (1–4 weeks)

• **Public S3 Bucket Exposing Internal Documents** — 1) Set the bucket ACL to private immediately. 2) Enable S3 Block Public Access at the account level. 3) Audit all other S3 buckets in the account. 4) Enable S3 server access logging. 5) Notify affected individuals of the data exposure per applicable privacy regulations (PIPEDA, GDPR).

• **Unauthenticated Redis Instance (Port 6379)** — 1) Bind Redis to localhost (bind 127.0.0.1) immediately. 2) Enable Redis AUTH with a strong password. 3) Block port 6379 at the firewall. 4) Flush all cached sessions (FLUSHALL) after securing — treat all active sessions as potentially compromised.

• **Exposed Laravel Log File with Customer PII** — Block access to the /storage directory via Nginx/Apache configuration. Move the storage directory outside the web root entirely. Rotate any credentials or tokens visible in the log file. Implement log sanitization to prevent PII from appearing in application logs.

• **No Email Authentication (SPF, DKIM, DMARC All Missing)** — 1) Publish an SPF record listing all authorized sending IPs. 2) Configure DKIM signing on your mail server. 3) Publish a DMARC policy at _dmarc.[domain-redacted].com starting with p=none for monitoring, then progress to p=quarantine. 4) This is high priority given the financial services context — email fraud is a direct revenue risk.

• **API Key Found in Public GitHub Commit History** — Rotate the exposed API key immediately — deletion of a GitHub commit does not remove it from history. Enable GitHub secret scanning on all repositories. Implement pre-commit hooks with git-secrets or similar to prevent future credential commits.

Legal Disclaimer

This report was prepared by Exposurix for the exclusive use of the named client. The findings reflect the security posture at the time of the assessment. Exposurix makes no warranties as to completeness. The client is solely responsible for remediation decisions. Unauthorized distribution of this report is prohibited.