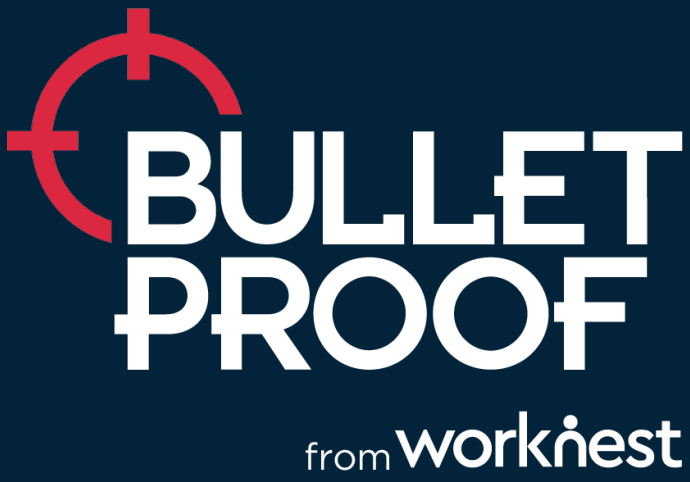


FRAUDFINDER LTD - PENETRATION TESTING (FULL REPORT)

CONFIDENTIAL

DOCUMENT CONTROL

CLASSIFICATION	CONFIDENTIAL
DATE	17/12/2025 - 17/12/2025
APPROVED BY	Jonathan Ross
DOCUMENT REFERENCE	R1-PT85118-3868
LABEL	_[Retest]
DELIVERED	19/12/2025

VERSION	DATE	DESCRIPTION
0.1	07/11/2025	Document Creation
0.2	11/11/2025	Report Completed - Keelan Brady
0.3	11/11/2025	QA Approved - Kunal Kukreja
1.0	11/11/2025	Delivered - Kunal Kukreja
1.1	17/12/2025	Retest Creation
1.2	19/12/2025	Retest Completed - Dimitrios Tsagkarakis
1.3	19/12/2025	QA Approved - Jonathan Ross
2.0	19/12/2025	Delivered - Jonathan Ross

Your penetration test report is delivered through our cyber security SaaS product, Defense.com™ by Bulletproof penetration testers. Bulletproof/Target Defence penetration testers are independently qualified by industry-recognised bodies such as CREST and can be found on the following CREST member companies list https://service-selection-platform.crest-approved.org/member_companies/bulletproof-cyber-serverchoice/.

TABLE OF CONTENTS

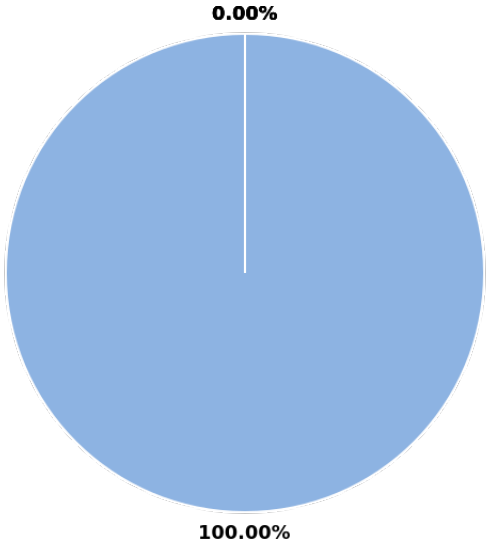
1. EXECUTIVE SUMMARY

1.1 TEST PARAMETERS

DOCUMENT REFERENCE	R1-PT85118-3868
TEST START	17/12/2025
TEST END	17/12/2025
TEST TIME	Office Hours (08:00 - 17:30 UTC)
TEST TYPE	Web App Unauthenticated
TEST TECHNIQUE	Grey box
LIMITATIONS	Assessing and exploiting vulnerabilities that could lead to a denial of service., Social engineering is excluded as way for granting access to an application., There will be no re-test included with this service.
PERSONNEL	Dimitrios Tsagkarakis

1.2 RESULTS SUMMARY

RISK LEVEL	RISKS FOUND
Critical	0
High	0
Medium	0
Low	0
Recommendations	2
TOTAL	2



1.3 RISK RATING TABLE

AREA	DESCRIPTION	RATING
Configuration	Overall security level of service and device configurations.	Strong
Authentication	User authentication methods used.	Strong
Patching	Vulnerable software versions.	Strong
Encryption	Encryption methods and protocols used.	Strong

1.4 TEST TARGETS

TYPE	TARGET
App	https://rc.fraudfinder.app/

2. ASSESSMENT OVERVIEW

2.1 ENVIRONMENT OVERVIEW

Scope

An application penetration test was performed against the FRAUDFINDER LTD web application.

The aim of the assessment was to gather the necessary information concerning the target in scope from the previous report, verifying any changes and fixes that might have occurred, and reflecting this with evidence within this report. Such re-assessment aimed to reflect the new business risk and impact. The assessment was carried out from an unauthenticated perspective and was conducted in-line with security best practises. The methodology that was used was comprised of all applicable OWASP practises, an industry-approved framework.

The security assessment started by performing automated scans and specific manual checks required to assess all previously flagged weaknesses, whereas any attempts to discover new weaknesses have been excluded.

A retest was conducted which covered all previously identified vulnerabilities within the original assessment.

Overview

It was evident that remediation efforts had been performed as 2 issues had been fully resolved. However, the remaining 2 issues were either unresolved or had only been partially resolved.

A full breakdown has been included below for clarity:

- `Outdated Website Library - PDF.js: Resolved`
- `HTTP Header Disclosure: Partially Resolved (Severity changed to Recommendation - no severity - listed for informational purposes)`
- `Missing HTTP Security Headers: Resolved`
- `Content Spoofing - Text Injection: Unresolved`

A Response header was still found to disclose information regarding the back-end software being utilised by the server. This information included the name of the service, "Netlify". It is recommended that this be remediated by changing server configurations to not disclose such information in response headers, as they are not needed and only serve as information that an attacker can consider when planning an attack.

The magic email link functionality was still found to result in a templated response, which used the user's email within a request parameter, which was then reflected on the template. In an attack scenario this could be used to spoof content, allowing for user-based attacks such as phishing to gain more credibility as the spoofed content would be hosted on a trusted source. It's recommended that email reflections be handled server-side.

Caveats

It is important that FRAUDFINDER LTD are aware that the vulnerabilities found resolved or unresolved in this document are dependent on the time and test limitations given for this penetration test. Other issues may come to light after this test, which is why it is recommended to carry out regular penetration tests.

2.2 BUSINESS RISK SUMMARY

The recommendations set forth in this report are the result of the re-assessment of the shortcomings identified in the original application penetration testing activities that were carried out against the original pre-defined scope.

Throughout the first round of tests, a number of weaknesses were identified, ranging from High risk to Recommendation which posed an immediate threat to key business operations.

Following the re-test activities, it was identified that two issues had been fully resolved.

Taking this into consideration, the overall 'Configuration' and 'Patching' security levels of the environment was upgraded to Strong on the basis that the remaining issues were isolated in nature or are part of a hardening strategy.

The below business risk is still present, and likely affects the wider organisation.

A number of hardening suggestions are present within this report; software and various other products are rarely considered to be secure 'out of the box' or are not set to the highest security level. A system hardening process involves actions taken to reconfigure systems and applications to adhere to best practices by reducing the attack surface and consequently leading to an improvement of the security posture. The hardening suggestions identified predominantly concerned the web application and as such, it is recommended that a number of hardening activities are performed across the board to ensure that other applications/systems within the estate are not affected with the same or similar issues.

To conclude, it is strongly advised that a re-test is planned once remediation of the final remaining issues is applied. It is also strongly suggested to consider planning a dedicated penetration test that will be carried out from an authenticated perspective at least on annual basis. This will aim to identify any new risks within the environment and improve the security posture overall.

2.3 RISK RESULTS

DESCRIPTION	RISK RATING	REFERENCE
HTTP Header Disclosure	RECOMMENDATION	R1-PT85118-3868-R0570
Content Spoofing - Text Injection	RECOMMENDATION	R1-PT85118-3868-R8273

2.4 CRITICALITY INDEX

Findings have been measured in-line with the CVSS scoring system.

RISK LEVEL	DESCRIPTION	RECOMMENDATION
CRITICAL SCORE: 9-10	A critical risk indicates serious and immediate risk to systems and data being compromised.	Critical rated issues need to be addressed and resolved immediately.
HIGH SCORE: 7-9	High risk indicates that a serious weakness or exposure exists.	High rated issues need to be addressed and resolved immediately.
MEDIUM SCORE: 4-7	Medium risk indicates that a significant issue needs to be addressed.	Actions need to be taken once high risks have been addressed.
LOW SCORE: 1-4	Low-risk indicates minor issues that generally are harmless but can be used when profiling an organisation.	No immediate action is required but should be addressed through the remediation phase.
RECOMMENDATION SCORE: N/A	Recommendations are included for improvements purposes only as they pose an indirect risk to current environment.	N/A

3. ASSESSMENT RESULTS

3.1 WEB APP UNAUTHENTICATED

3.1.1 OUTDATED WEBSITE LIBRARY - PDF.JS

REFERENCE	R1-PT85118-3868-R7969
AFFECTED TARGETS	https://rc.fraudfinder.app/
PASS / FAIL	PASS
DESCRIPTION	

Retest

The initially identified issue was found to be resolved during the retesting. Specifically, v3.11.174 was not identified. No other instances of PDF.js were identified.

Initial Test

An outdated library with one vulnerability was identified within the target environment.

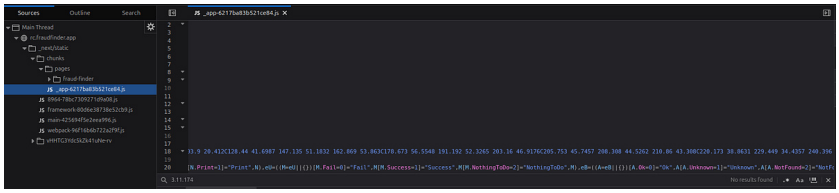
Libraries used by web services are often overlooked by an organisation's patching policy. As such, outdated web service software has been a feature in the OWASP top 10 for many years. Known vulnerabilities can often be trivially exploited by an attacker, especially if exploit code is publicly available.

In this case, PDF.js v3.11.174 was found to be installed, which is vulnerable to Arbitrary Code Injection (CVE-2024-4367) Attacks.

EVIDENCE

FINDING 1 - RETEST - PDF.JS (RESOLVED)

The following figure is presented as a proof of concept that the vulnerable version was not found.



Retest - PDF.js

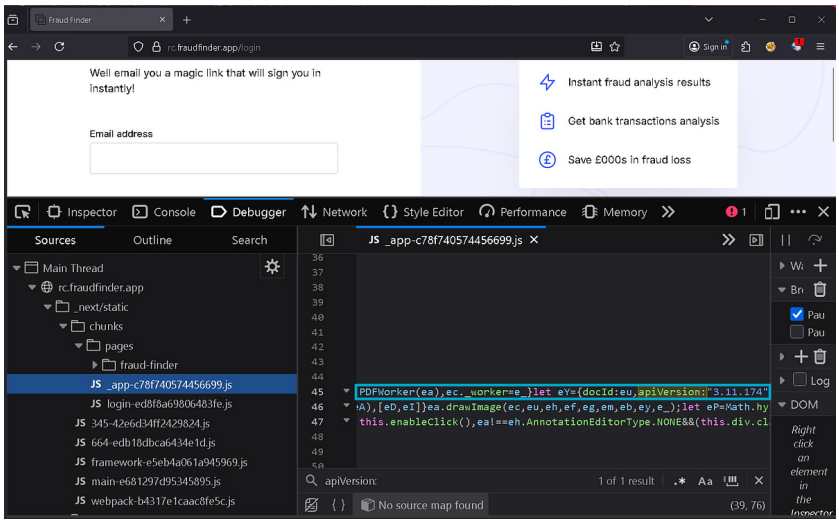
FINDING 2 - INITIAL TEST - PDF.JS

Usage of the web application utilises the app JavaScript file to import libraries. A direct link cannot be given as a random string of letters and numbers appended to the _app- filename; however, checking the Sources tab of the Chrome Developer Tools reveals the exact file.

URL: https://rc.fraudfinder.app/_next/static/chunks/pages/_app-RAND.js

Snipped Response:

em=er.port?PDFWorker.fromPort(er):new PDFWorker(er),ea._worker=em}let eG={docId:eo,apiVersion:"3.11.174"



PDF.js version seen from login page

REFERENCES

https://nvd.nist.gov/vuln/detail/cve-2024-4367

3.1.2 HTTP HEADER DISCLOSURE

REFERENCE	R1-PT85118-3868-R0570
AFFECTED TARGETS	https://rc.fraudfinder.app/
PASS / FAIL	FAILED
SEVERITY	RECOMMENDATION
LIKELIHOOD	NOT_APPLICABLE
EFFORT TO FIX	NOT_APPLICABLE
DESCRIPTION	

Retest

The initially identified issue was found to be partially resolved during the retesting. Specifically, only the Netlify header was found. The severity of the finding was changed to a 'Recommendation' and was reported for informational purposes.

Initial Test

Multiple HTTP Headers were observed within the environment that were found to disclose sensitive information; an attacker may be able to leverage this information in order to gain further insight into the environment or simply to tailor attacks to a specific product or server. Therefore, it is considered good practice to exclude information such as this from HTTP responses.

The HTTP server within the tested scope provided information about the underlying infrastructure of the environment; some examples of the type of data contained within the HTTP responses are shown in the evidence below.

EVIDENCE

FINDING 1 - RETEST - HTTP HEADER DISCLOSURE (PARTIALLY RESOLVED)

The following snippet is presented as a proof of concept that the Next.js header was not found in the retest.

Request

```
GET /assets/branding/fraudfinder HTTP/2
Host: rc.fraudfinder.app
... snip ...
```

Response

```
HTTP/2 404 Not Found
Age: 0
Cache-Control: private,no-cache,no-store,max-age=0,must-revalidate
Cache-Status: "Netlify Durable"; fwd=bypass
Cache-Status: "Netlify Edge"; fwd=miss
Content-Security-Policy: frame-ancestors 'none'
Content-Type: text/html; charset=utf-8
Date: Wed, 17 Dec 2025 09:55:43 GMT
Etag: "14nefkbnvch23a-df"
Netlify-Vary: query=__nextDataReq|_rsc,header=x-nextjs-data|x-next-debug-logging|next-router-prefetch|next-router-segment-prefetch|next-router-state-tree|next-url|rsc|accept-encoding,cookie=__prerender_bypass|__next_preview_data
Permissions-Policy: accelerometer=(), ambient-light-sensor=(), autoplay=(), battery=(), camera=(), cross-origin-isolated=(), display-capture=(), document-domain=(), encrypted-media=(), fullscreen=(), geolocation=(), gyroscope=(), magnetometer=(), microphone=(), midi=(), payment=(), picture-in-picture=(), publickey-credentials-get=(), screen-wake-lock=(), sync-xhr=(), usb=(), web-share=(), xr-spatial-tracking=()
Referrer-Policy: no-referrer
Server: Netlify
Strict-Transport-Security: max-age=31536000
Vary: Accept-Encoding
X-Content-Type-Options: nosniff
X-Frame-Options: DENY
X-Nf-Request-Id: 01KCNVSABDB6BP150NX8PV80JS
... snip ...
```

FINDING 2 - INITIAL TEST - HTTP HEADER DISCLOSURE

Requests to the asset were found to respond with HTTP headers that revealed the underlying service of the application. The `Next.JS` header was specifically found when on 404 pages:

URL: `https://rc.fraudfinder.app/assets/branding/fraudfinder/`

Response: (trimmed to only show relevant data)

```
HTTP/2 404 Not Found
Server: Netlify
X-Powered-By: Next.js
```

REMEDIATION

The web server should be reconfigured to ensure that software version information, which could be used by an attacker, is not included in HTTP responses.

For Netlify

Service documentation should be reviewed to remove or overwrite the "Server" headers from the current response. If unable to be removed it could potentially be overwritten to something not representative of the software such as Server: FraudFinder

REFERENCES

https://cheatsheetseries.owasp.org/cheatsheets/HTTP_Headers_Cheat_Sheet.html

3.1.3 MISSING HTTP SECURITY HEADERS

REFERENCE	R1-PT85118-3868-R5349
AFFECTED TARGETS	https://rc.fraudfinder.app/
PASS / FAIL	PASS
DESCRIPTION	

Retest

The initially identified issue was found to be resolved during the retesting. Specifically, the initially suggested headers were implemented.

Initial Test

Several HTTP headers that can be used to provide additional security were not used; these security headers can help defend against a number of common attack vectors.

It is advised to include these headers whenever applicable, due to their following functions:

- `Content-Security-Policy` - Aids in preventing code injection, such as XSS, by defining what content sources are allowed to be loaded by the browser.
- `X-Content-Type-Options` - Protects against MIME-type sniffing.
- `X-Frame-Options` - Prevents the website from being placed into an iframe, depending on the options set. This is often used to prevent clickjacking.
- `Referrer-Policy` – Allows the ability to censor the URL that would be provided to another website when a user is directed there (e.g. via a hyperlink). This can allow for data leakage if the URL contains sensitive information.
- `Permissions-Policy` - This optional policy allows the web application developers to specify which peripheral devices can be requested for use. Limiting the access of these peripherals helps protect the privacy of users.

EVIDENCE

FINDING 1 - RETEST - MISSING HTTP SECURITY HEADERS (RESOLVED)

The following snippet is presented as a proof of concept that the initially suggested HTTP response headers were implemented.

Request

```
GET /login HTTP/2
Host: rc.fraudfinder.app
... snip ...
```

Response

```
HTTP/2 200 OK
Age: 0
Cache-Control: public,max-age=0,must-revalidate
Cache-Status: "Netlify Durable"; fwd=stale; ttl=29298522; stored
Cache-Status: "Netlify Edge"; fwd=miss
Content-Security-Policy: frame-ancestors 'none'
Content-Type: text/html; charset=utf-8
Date: Wed, 17 Dec 2025 09:58:00 GMT
Etag: "nf9rz4ugc925k-df"
Netlify-Vary: query=__nextDataReq|_rsc,header=x-nextjs-data|x-next-debug-logging|next-router-prefetch|next-router-segment-prefetch|next-router-state-tree|next-url|rsc|accept-encoding,cookie=__prerender_bypass|__next_preview_data
Permissions-Policy: accelerometer=(), ambient-light-sensor=(), autoplay=(), battery=(), camera=(), cross-origin-isolated=(), display-capture=(), document-domain=(), encrypted-media=(), fullscreen=(), geolocation=(), gyroscope=(), magnetometer=(), microphone=(), midi=(), payment=(), picture-in-picture=(), publickey-credentials-get=(), screen-wake-lock=(), sync-xhr=(), usb=(), web-share=(), xr-spatial-tracking=()
Referrer-Policy: no-referrer
Server: Netlify
Strict-Transport-Security: max-age=31536000
Vary: Accept-Encoding
X-Content-Type-Options: nosniff
X-Frame-Options: DENY
X-Nf-Request-Id: 01KCNVXG3FHH4MNK25Z1J4H81J
... snip ...
```

FINDING 2 - INITIAL TEST - MISSING HTTP SECURITY HEADERS

Requests to the asset were found to respond with a HTTP headers where a lack of the aforementioned security headers was observed:

URL: https://rc.fraudfinder.app/login

Response:

```
HTTP/2 200 OK
Accept-Ranges: bytes
Age: 1
Cache-Control: public,max-age=0,must-revalidate
Content-Type: text/html; charset=UTF-8
Date: Fri, 07 Nov 2025 09:50:18 GMT
Etag: "830356a4739e44ab5c058e3db7d2ac44-ssl-df"
Server: Netlify
Strict-Transport-Security: max-age=31536000
Vary: Accept-Encoding
Content-Length: 2711
[...]
```

REFERENCES

https://cheatsheetseries.owasp.org/cheatsheets/HTTP_Headers_Cheat_Sheet.html

3.1.4 CONTENT SPOOFING - TEXT INJECTION

REFERENCE	R1-PT85118-3868-R8273
AFFECTED TARGETS	https://rc.fraudfinder.app/
PASS / FAIL	FAILED
SEVERITY	RECOMMENDATION
LIKELIHOOD	NOT_APPLICABLE
EFFORT TO FIX	NOT_APPLICABLE
DESCRIPTION	

Retest

The initially identified issue was found to be unresolved during the retesting. Specifically, the text injection was still possible.

Initial Test

The systems in scope did not possess a robust input validation strategy and as such, malicious payloads or characters used to conduct further attacks could be supplied for a variety of parameters within the available input fields and parameters. This opens the systems in scope to the risk of injection vulnerabilities such as content spoofing.

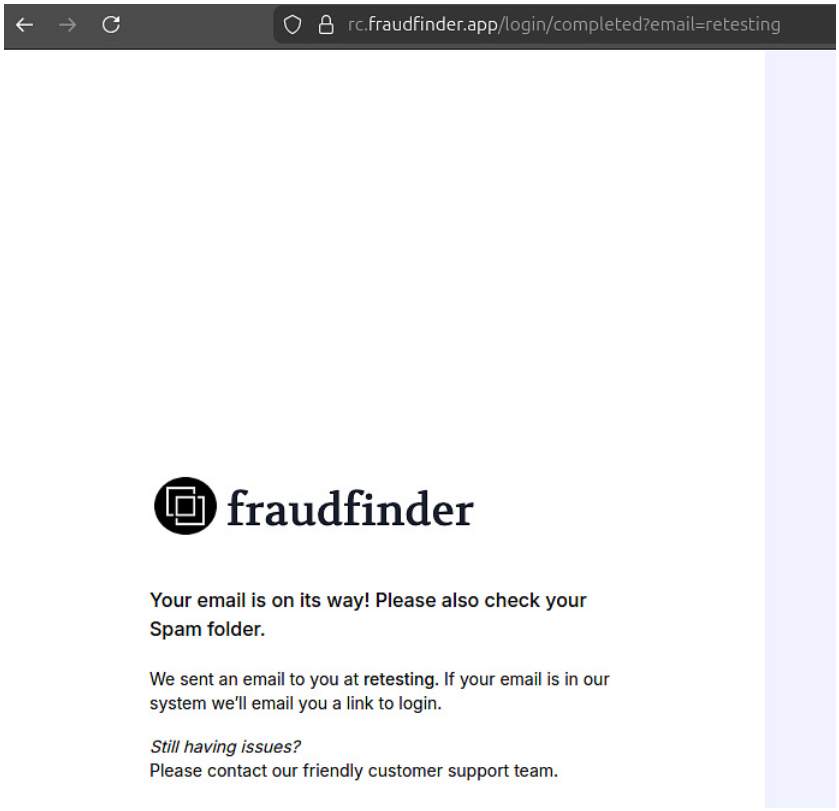
Content spoofing, also referred to as content injection, arbitrary text injection or virtual defacement, is an attack targeting a user made possible by an injection vulnerability in a web application. When an application does not properly handle user-supplied data, an attacker can supply content to a web application, typically via a parameter value, that is reflected back to the user. This presents the user with a modified page under the context of the trusted domain.

This attack is typically used as, or in conjunction with, social engineering because the attack is exploiting a code-based vulnerability and a user's trust.

EVIDENCE

FINDING 1 - RETEST - TEXT INJECTION (UNRESOLVED)

The following figure is presented as a proof of concept that the text injection was still possible during the retest.

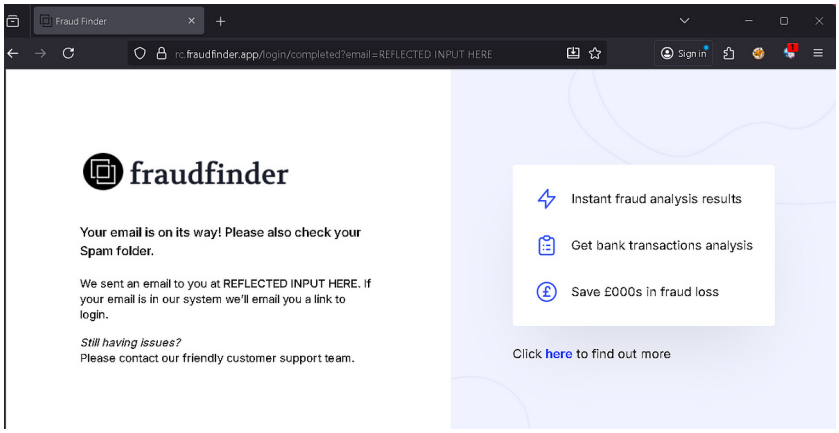


Retest - Text Injection

FINDING 2 - INITIAL TEST - TEXT INJECTION

A content spoofing attack could be used to present false information to application users via text manipulation. The magic link functionality results in a templated response, which takes in the user's email as a URL parameter, which is then reflected on the template.

URL: `https://rc.fraudfinder.app/login/completed?email=REFLECTED INPUT HERE`



Input Reflection

REMEDIATION

Use ephemeral sessions instead of presenting messages via request parameters and validate the data returned by the request parameter.

4. APPENDIX

4.1 TESTING METHODOLOGY

This Bulletproof Cyber Limited penetration test used the CREST framework as an overarching methodology, into which the required frameworks are embedded, such as the Penetration Testing Execution Standard (PTES) and Open Web Application Security Project (OWASP).

The below "test highlights" listed for each assessment category describe some of the most important areas that will be covered as part of the engagement. These objectives are primarily achieved by assessing aspects such as the design, configuration, deployment, operational security and direct/indirect risks of all assets in scope. The assessments carried out, checks performed and security best practice recommendations are all in line with industry approved standards and methodologies. Furthermore, our bespoke engagements often include additional custom checks and attack scenarios that are tailored against the target environment and customer.

WEB APPLICATION ASSESSMENT

- Perform application mapping to identify dynamic functionality in use according to the intended design/purpose and any hidden content.
- Assess the application(s) and any associated components/dependencies for their patch levels.
- Tamper with available functionalities and parameters in order to manipulate/bypass authentication, identify lack of input sanitisation, exploit any injection-based vulnerabilities, leverage improper session management, etc.
- Assess the implemented access control enforcement; and as applicable based on the agreed roles in a vertical and horizontal manner; including any data exposure from unintended functionalities.
- Assess the cryptographic protocols and ciphers that are used by the application(s) and their dependencies that provide secure communications.
- Assess the deployment of the application(s) to their respective service(s).