SecureDapp

Your Cybersecurity Edge

VAPT SECURITY AUDIT

of

ARVO

VAPT Security Audit of ARVO

June 13th, 2025 | v. 1.1

TABLE OF CONTENT

AUDIT INTRODUCTION

Auditing Firm	SecureDApp Auditors
Audit Architecture	Secure DApp Auditing Standard
Client Firm	ARVO
Website	https://onearvoventures.com/
Linkedin	https://www.linkedin.com/company/onearvoventures/
Twitter	https://x.com/one_ARVO

About SecureDApp

SecureDApp is a leader in security audits, offering advanced audits and run-time security solutions for projects.

SecureDApp is a Startup recognized by the Department For Promotion Of Industry And Internal Trade, Ministry of Commerce and Industry, Government of India. Powered by Vettedcode Technologies India Pvt. Ltd.

AUDIT DOCUMENT

Name	VAPT Security Audit Report for ARVO
Approved By	Himanshu Gautam| CTO and Co-Founder @SecureDApp
Type	VAPT

AUDIT SCOPE

The Vulnerability Assessment and Penetration Testing (VAPT) for ARVO Services was conducted by SecureDApp with the following defined scope:

In-Scope Components:

The assessment focused on the following areas of ARVO Services:

Web Application Security Testing:

- Authentication & Authorization mechanisms (login, session management, role-based access).
- Input validation and business logic flaws (e.g., IDOR, account bypass).
- Security misconfigurations (headers, CORS, file uploads).
- API security (excessive data exposure, rate limiting).
- Frontend vulnerabilities (React Router issues, clickjacking).

Security Headers & Configurations:

- Missing or misconfigured headers (CSP, X-Frame-Options, HSTS, Referrer-Policy).
- Transport security (HTTPS enforcement, TLS weaknesses).
- Session Management:
- Session fixation, logout functionality, and token handling.

Data Exposure Risks:

- Unintended information leakage (server headers, API responses, error messages).

Limitations:

The VAPT assessment conducted for ARVO Services by SecureDApp was comprehensive, but it had certain limitations that may affect the scope and completeness of the findings:

Scope Restrictions:

- Only predefined systems, applications, and endpoints were tested. Any out-of-scope components (e.g., third-party integrations or internal backend systems) were not assessed.
- Testing was limited to the provided test environment, which may not fully replicate production conditions.

Testing Methodology:

- Automated tools were used for initial scanning, but manual validation was required for critical findings. Some vulnerabilities (e.g., logic flaws or business logic bypasses) may require deeper manual analysis.

Tools:

- **Burp Suite:**
  Burp Suite is a powerful web vulnerability scanner and penetration testing tool used to identify and exploit security flaws in web applications. It includes features like a proxy server, scanner, repeater, and intruder for comprehensive testing.
- **Wappalyzer:**
  Wappalyzer is a browser extension and online tool that identifies the technologies used by websites, such as content management systems, web servers, JavaScript frameworks, and analytics tools.
- **Security Headers:**

Security Headers is a web-based tool that analyzes a website's HTTP response headers and scores them based on their implementation of security-related headers like Content-Security-Policy, Strict-Transport-Security, and X-Frame-Options.

- **Nmap**

  Nmap, short known for Network Mapper, is a free and open-source tool for vulnerability scanning and network discovery. Network administrators use Nmap to identify active devices on their networks, discovering hosts that are available and the services they offer, finding open ports and detecting security risks. 7.94 (Open Source)
- **Metasploit**

  The Metasploit framework is a very powerful tool that can be used by cybercriminals as well as ethical hackers to probe systematic vulnerabilities on networks and servers. pentesting team can use ready-made or custom code and introduce it into a network to probe for weak spots. Version 6.4.x (Open Source)
- **Kali Linux**

  Kali Linux, an Advanced Penetration Testing Linux distribution used for Penetration Testing, Ethical Hacking and network security assessments. It preloaded with lot of web app security tools to do security assessment. 2024.1 (Open Source)

Review Scope

Application URL(s)	https://arvo.services/

Severity Definitions

Risk Level	Description
Critical	Critical vulnerabilities are usually straightforward to exploit and can lead to asset loss or data manipulations.
High	High-level vulnerabilities are difficult to exploit; however, they also have a significant impact on smart contract execution, e.g., public access to crucial functions.
Medium	Medium-level vulnerabilities are important to fix; however, they cannot lead to asset loss or data manipulations.
Low	Low-level vulnerabilities are mostly related to outdated, unused, etc. code snippets that cannot have a significant impact on execution.

All statuses which are identified in the audit report are categorized here for the reader to review:

Status Type	Definition
Open	Risks are open.
Acknowledged	Risks are acknowledged, but not fixed.
Resolved	Risks are acknowledged and fixed.

AUDIT SUMMARY

The SecureDApp team has performed a line-by-line manual analysis and automated review of the ARVO system. The system was analyzed mainly for common vulnerabilities, exploits, and manipulation hacks. According to the audit:

Status	Critical	High	Medium	Low
Open	0	0	4	3
Acknowledged	0	1	0	1
Resolved	2	1	4	1

AUDIT METHODOLOGY

SecureDApp performs Website VAPT to identify security weaknesses, validate exploitability, and recommend remediations. Below are the systematic steps used by SecureDApp to audit websites:

a. Information Gathering (Reconnaissance):

   Understanding the target and collecting as much data as possible.

   i. WHOIS and DNS enumeration
   ii. Identifying technologies in use (CMS, frameworks, servers)
   iii. Subdomain enumeration and URL discovery
   iv. OSINT (Open Source Intelligence) for exposed information

b. Threat Modeling:
   Analyzing gathered information to map the attack surface.
   i. Identifying entry points
   ii. Mapping user roles and access levels
   iii. Assessing business logic and sensitive functions

c. Vulnerability Scanning:
   Using automated tools to detect known vulnerabilities.
   Testing for:
   i. SQL Injection
   ii. Cross-Site Scripting (XSS)
   iii. Command Injection
   iv. Insecure Direct Object References (IDOR)
   v. Security Misconfigurations
   vi. Outdated components

d. Manual Testing:
   Deep-dive testing to uncover logic flaws and vulnerabilities that scanners may miss.
   i. logic bypass
   ii. File upload/test bypass
   iii. Authentication and session management flaws
   iv. Custom script exploitation

FINDINGS SUMMARY:

SI.NO	Finding Title	Severity	Status
01	IDOR	Critical (C)	Closed
02	Account Bypass	Critical (C)	Closed
03	IDOR	High (H)	Closed
04	Excessive Data Exposure	High (H)	Acknowledged
05	Cross-origin resource sharing (CORS)	Medium (M)	Closed
06	Vulnerable React Router version	Medium (M)	Open
07	X-Frame-options header not enforced	Medium (M)	Closed
08	Referrer-policy not enforced	Medium (M)	Open
09	Clickjacking	Medium (M)	Closed
10	X-Content-type-options not enforced	Medium (M)	Open
11	Content security policy not enforced	Medium (M)	Open
12	No Rate Limit	Medium (M)	Closed
13	Strict transport security not enforced	Low (L)	Open

14	Permissions-Policy not enforced	Low (L)	**Open**
15	Unrestricted File Upload	Low (L)	**Open**
16	Improper session management	Low (L)	**Closed**
17	Excessive Server Information Exposure	Low(L)	**Acknowledged**

DETAILED FINDINGS:

Web-01: Insecure Direct Object Reference (IDOR)

Path/pointer/location: Login Page Packet

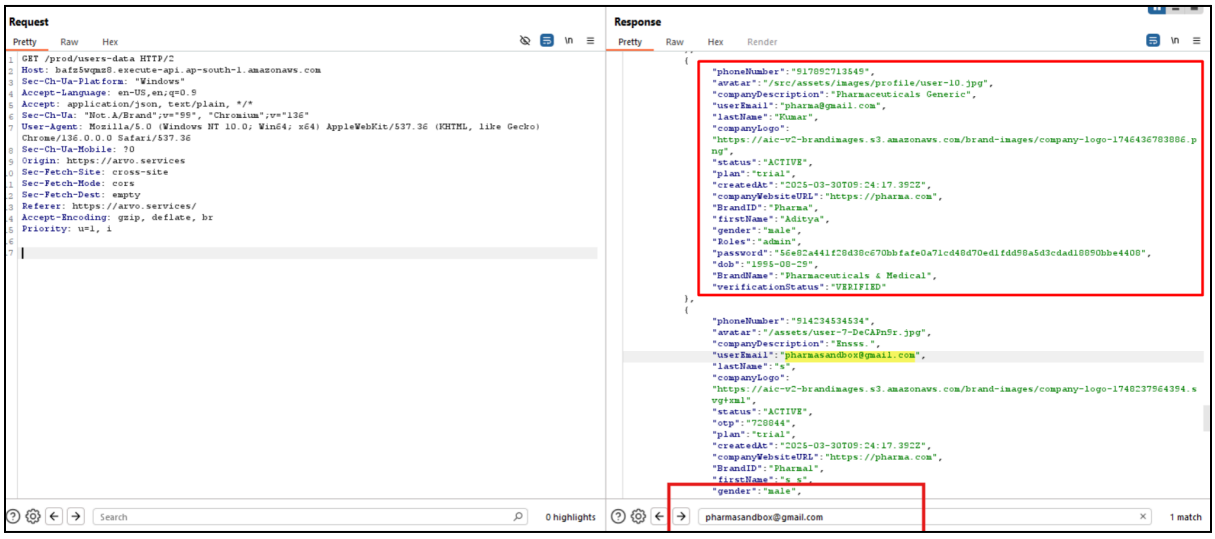
Severity: CRITICAL

CVSS: 9.1

Observation:

- It was observed that the application returns session or user data for all users during the authentication process without requiring any parameter manipulation by the attacker. Simply intercepting the authentication response reveals sensitive information about multiple users.
- This indicates a serious flaw in the application's access control mechanisms, where sensitive data is exposed globally rather than being restricted to the authenticated user only.

Proof of Concept:



Impact:

Attackers can intercept the authentication request and exploit an IDOR vulnerability to access session data of other users. This results in unauthorized disclosure of sensitive information.

Remediation:

Implement strict session isolation to ensure each user session is uniquely and securely scoped. Avoid exposing user identifiers or session metadata in client-side responses. Additionally, validate and sanitize all session-related data server-side, and use secure, short-lived session tokens with proper access controls to prevent unauthorized access.

Web-02: Account Bypass

Path/pointer/location: My Profile ⟶ Edit Page Packet

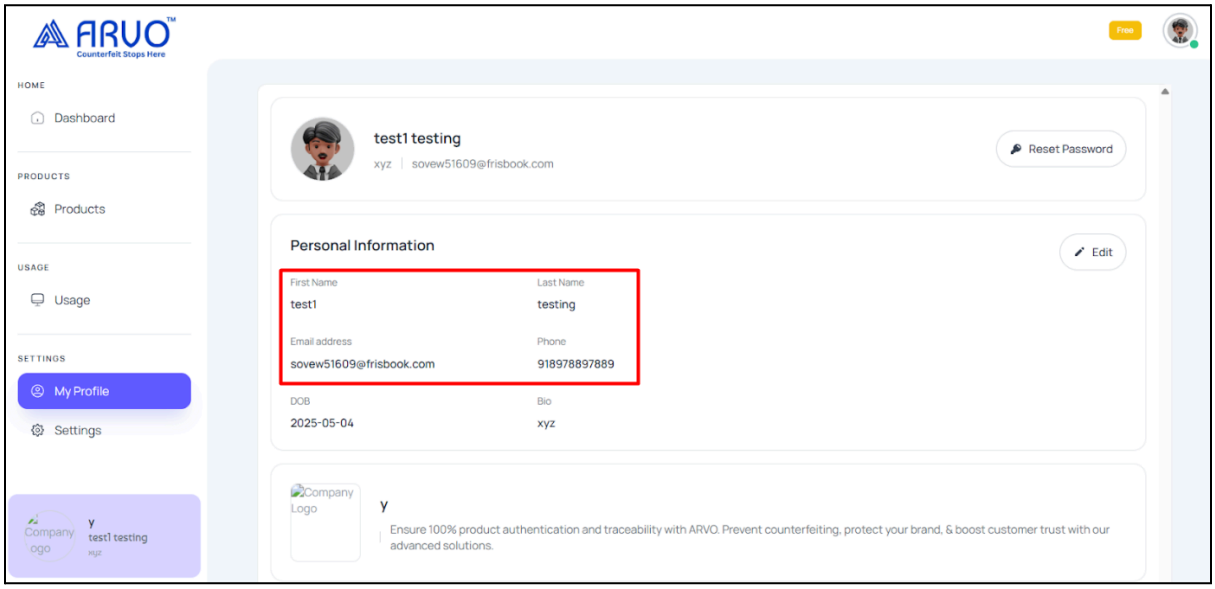
Severity: CRITICAL

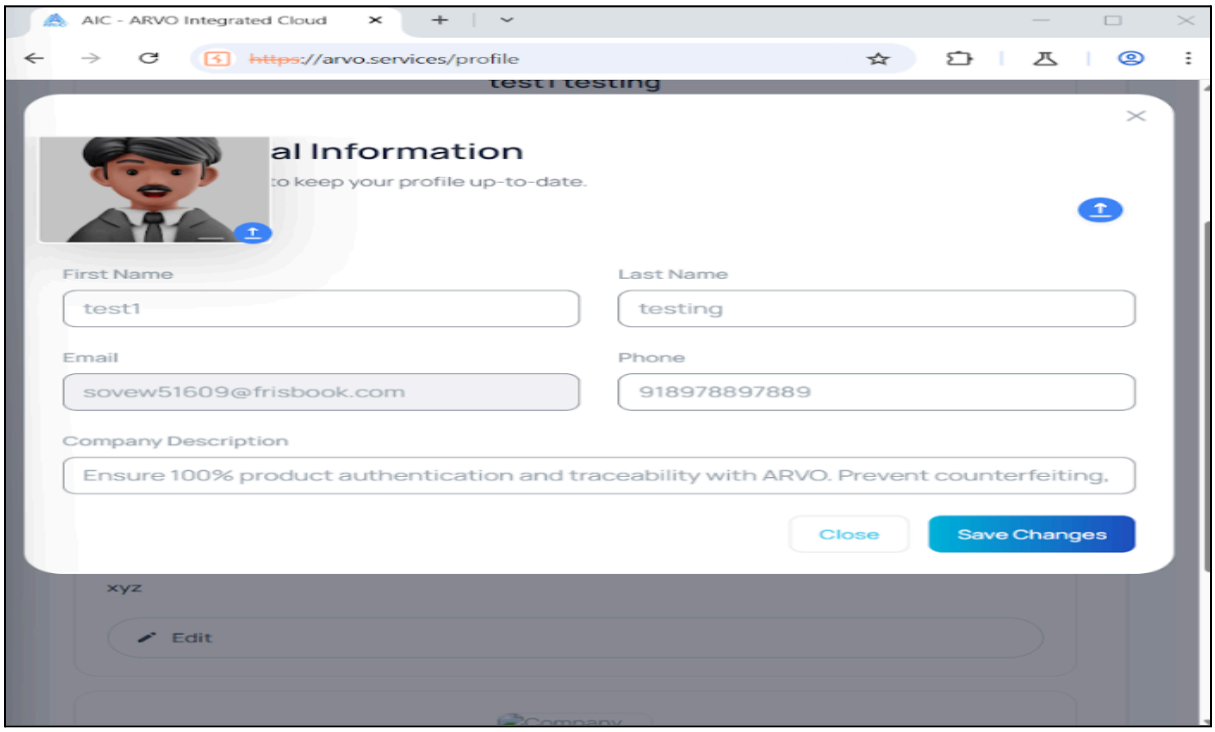
Observation:

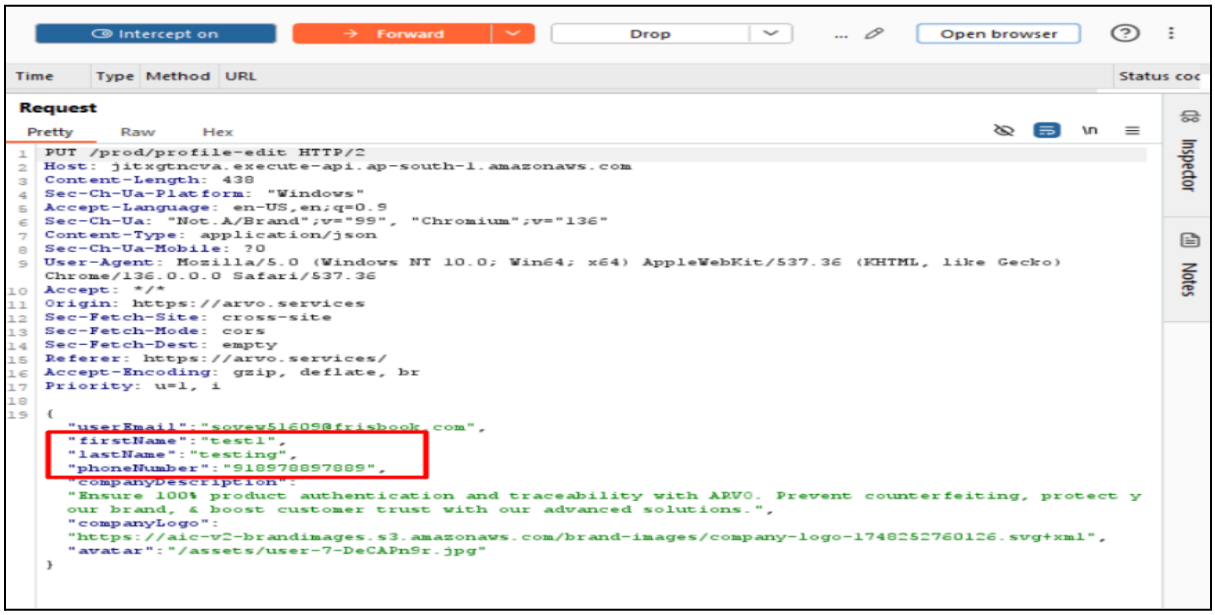
- It was observed that the application does not properly validate user authorization when updating profile information. By intercepting the profile update request and changing the user identifier (e.g., user ID), an attacker can successfully modify another user's profile details.
- This indicates a lack of proper access control on sensitive update operations.

Proof of Concept:

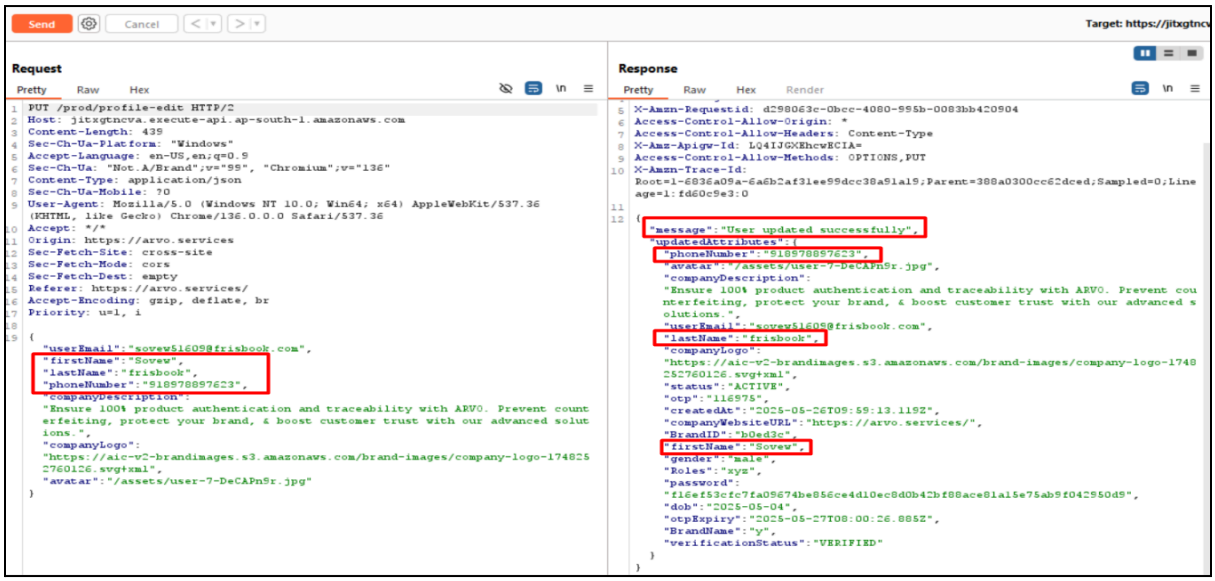
1) Before Modifying

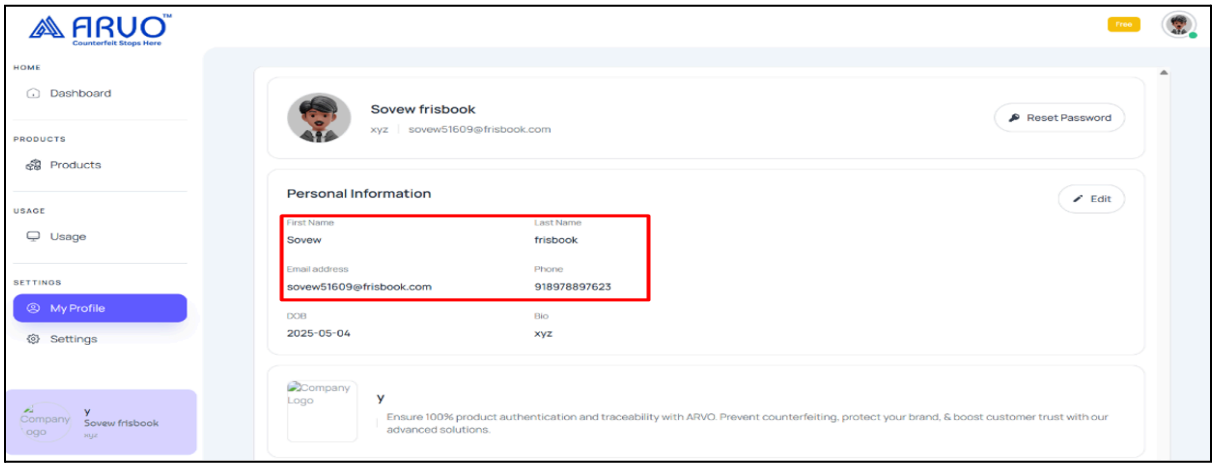



AIC - ARVO Integrated Cloud
https://arvo.services/profile
al Information
to keep your profile up-to-date.
First Name
test1
Last Name
testing
Email
sovew51609@frisbook.com
Phone
918978897889
Company Description
Ensure 100% product authentication and traceability with ARVO. Prevent counterfeiting.
Close
Save Changes
xyz
Edit


Intercept on
Forward
Drop
Open browser
Time
Type
Method
URL
Status coc
Request
Pretty
Raw
Hex
PUT /prod/profile-edit HTTP/2
Host: jitxgtncva.execute-api.ap-south-1.amazonaws.com
Content-Length: 438
Sec-Ch-Ua-Platform: "Windows"
Accept-Language: en-US,en;q=0.9
Sec-Ch-Ua: "Not.A/Brand";v="99", "Chromium";v="136"
Content-Type: application/json
Sec-Ch-Ua-Mobile: ?0
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/136.0.0.0 Safari/537.36
Accept: */*
Origin: https://arvo.services
Sec-Fetch-Site: cross-site
Sec-Fetch-Mode: cors
Sec-Fetch-Dest: empty
Referer: https://arvo.services/
Accept-Encoding: gzip, deflate, br
Priority: u=1, i
{
"userEmail":"sovew51609@frisbook.com",
"firstName":"test1",
"lastName":"testing",
"phoneNumber":"918978897889",
"companyDescription":
"Ensure 100% product authentication and traceability with ARVO. Prevent counterfeiting, protect your brand, & boost customer trust with our advanced solutions.",
"companyLogo":
"https://aic-v2-brandimages.s3.amazonaws.com/brand-images/company-logo-1748252760126.svg+xml",
"avatar":"/assets/user-7-DeCAPn9r.jpg"
}
Inspector
Notes

2) After Modifying





Impact:

An attacker can modify another user's profile details by intercepting and altering the request, leading to unauthorized account manipulation, data integrity issues, and potential account takeover.

Remediation:

Enforce strong server-side authorization checks to ensure users can only update their own profiles, validating the authenticated user's identity against the profile being modified before applying changes.

Web-03: Insecure Direct Object Reference (IDOR)

Path/pointer/location: Login Page Packet

Severity: HIGH

CVSS: 7.1

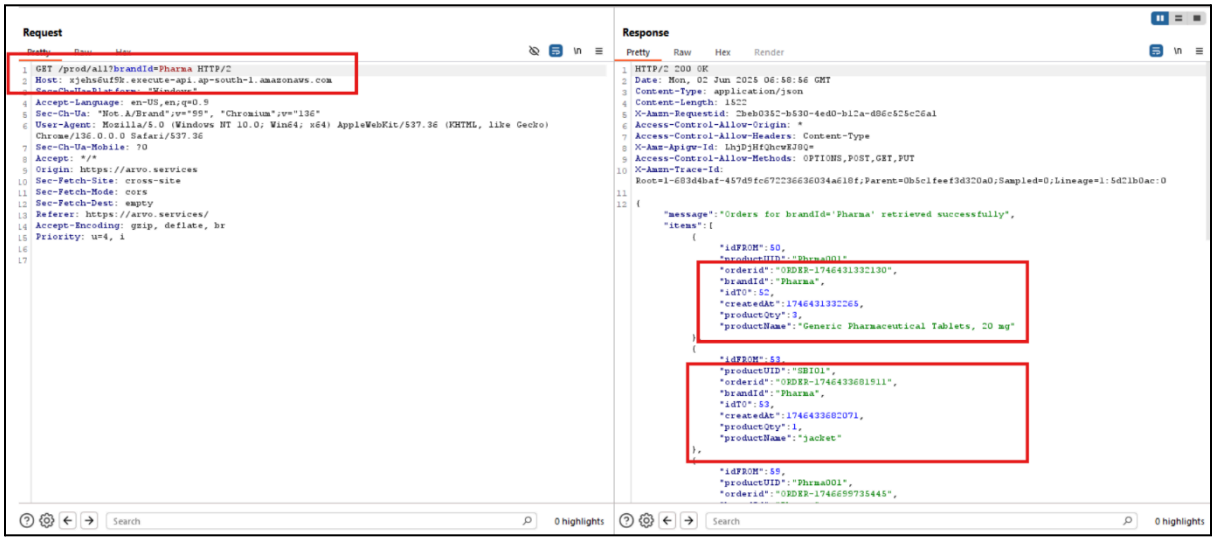
Observation:

- It was observed that the application does not enforce proper access controls on product data linked to brand identifiers. By manipulating the brand_id parameter in the request, an attacker can enumerate and access product information belonging to other brands, including unpublished or internal data.
- This indicates an Insecure Direct Object Reference (IDOR) vulnerability in the product lookup functionality.

Proof of Concept:

1) Before Modifying

2) After Modifying



Impact:

An attacker can enumerate and access product information by manipulating the brand_id, potentially exposing unpublished or internal data not intended for public view, leading to information leakage and unauthorized access.

Remediation:

Implement strict server-side access controls to validate user permissions and ensure only authorized, published product data is returned based on user roles and brand ownership.

Web-04: Excessive Data Exposure

Path/pointer/location: Login Page Packet

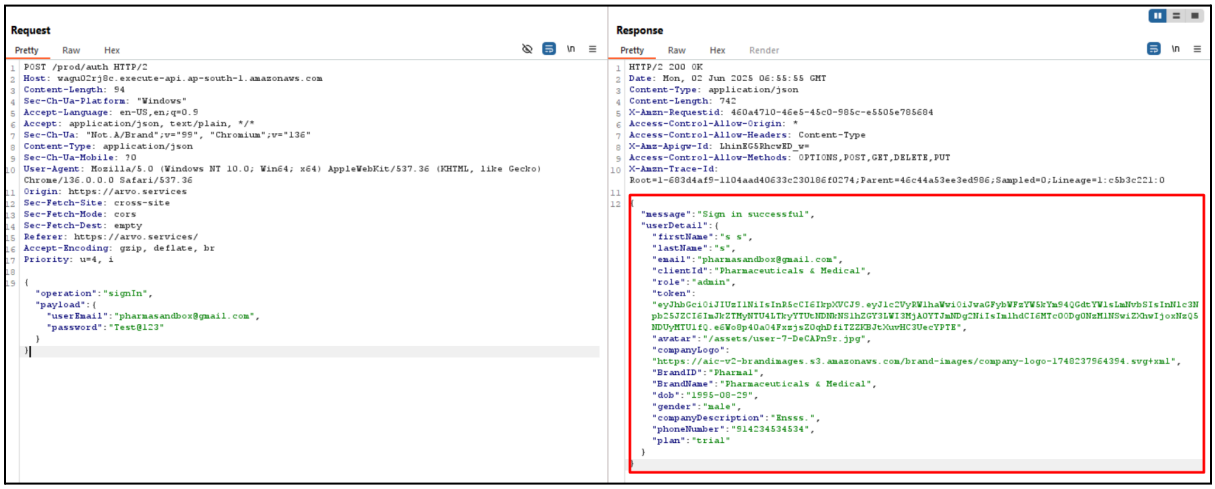
Severity: HIGH

CVSS: 7.5

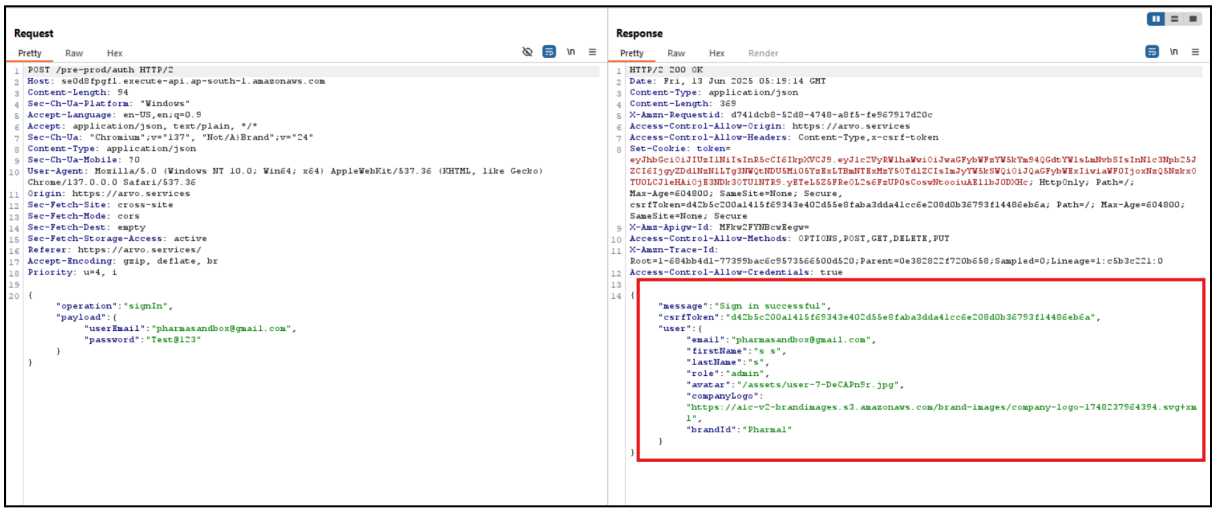
Observation:

- It was observed that the login API response includes full user details such as name, email, phone number, and date of birth in plaintext.
- This sensitive information is exposed immediately upon successful authentication, enabling attackers to harvest user data through automated or replayed login attempts.

Proof of Concept:



After Patch (13/06/2025)



Impact:

The login API response exposes full user details (name, email, phone, DOB, etc.) in plaintext, allowing attackers to harvest sensitive data by replaying or automating login attempts.

- **(CVE-2023-40662)**

Remediation:

Limit API responses to minimal authentication tokens only; never expose user profile data during login. Implement strict access control and sanitize responses to avoid unnecessary data leakage.

Web-05: Cross-origin resource sharing (CORS)

Path/pointer/location: Login Page Packet

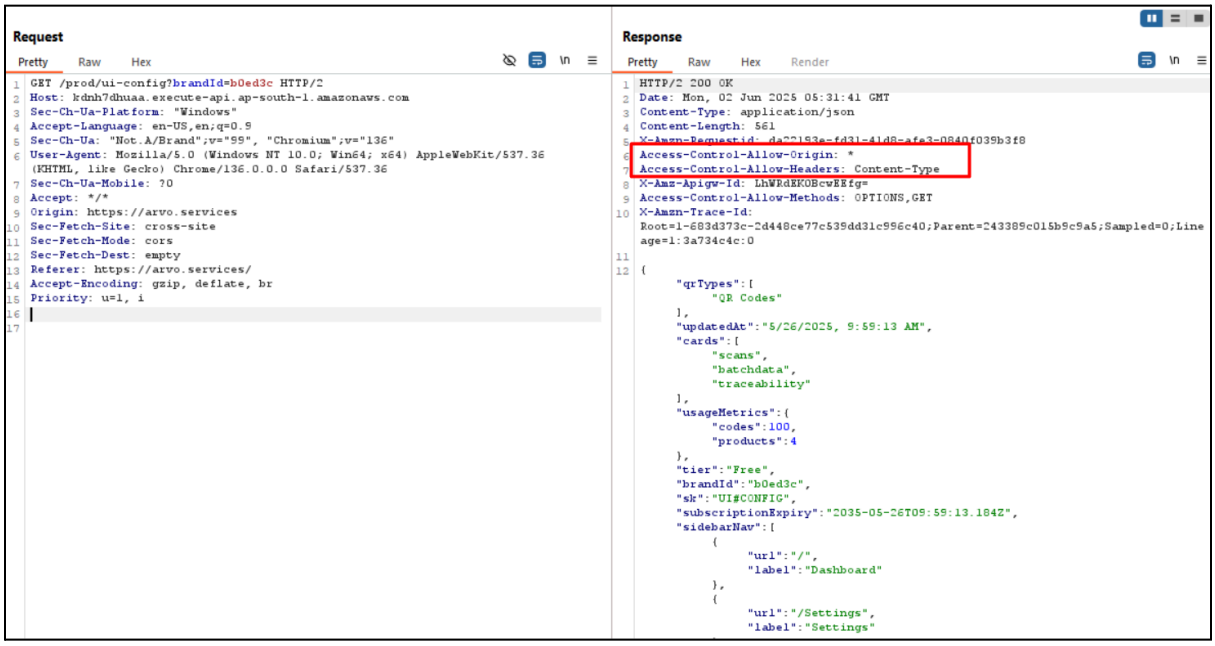
Severity: MEDIUM

CVSS: 6.8

Observation:

- It was observed that the server is configured to allow cross-origin requests from arbitrary or untrusted origins while Access-Control-Allow-Credentials is set to true.
- This insecure CORS configuration can allow attackers to perform unauthorized actions or access sensitive user data from another origin, potentially leading to privilege escalation and data leakage.

Proof of Concept:



Impact:

Cross-origin resource sharing (CORS) enables browsers to perform cross domain requests in a controlled manner. This request has an Origin header that identifies the domain that is making the initial request and defines the protocol between a browser and server to see if the request is allowed.

An attacker can take advantage of this and possibly carry out privileged actions and access sensitive information when the Access-Control-Allow Credentials is enabled.

- **(CVE-2021-27786)**

Remediation:

Disable Access-Control-Allow-Credentials unless necessary, validate Origin headers and limit cross-origin sharing to trusted domains to mitigate risks.

Web-06: Vulnerable React Router version

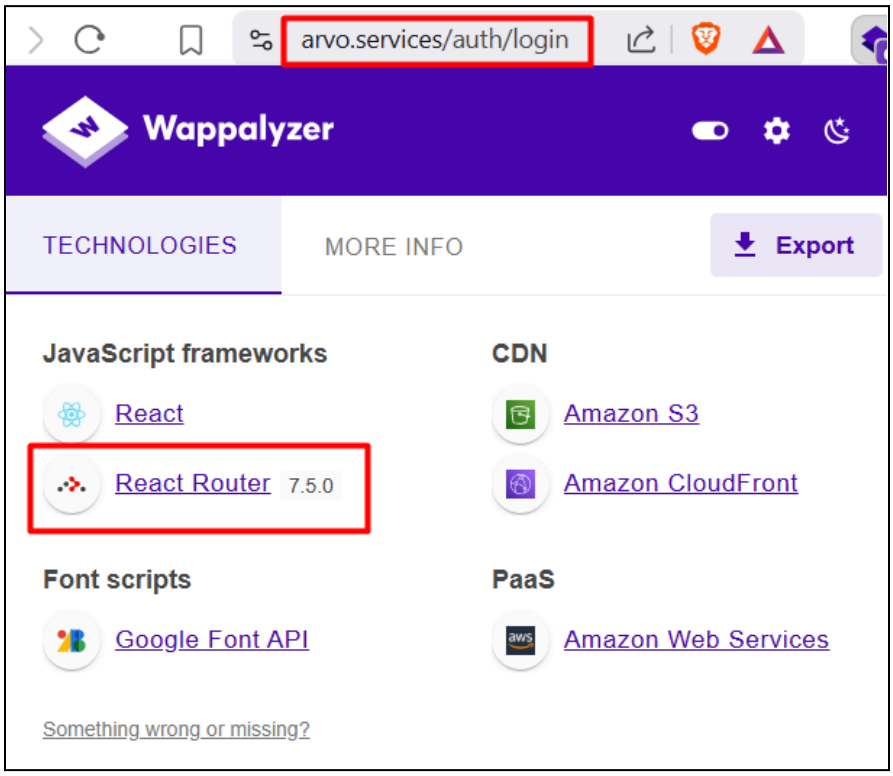
Path/pointer/location: Login Page

Severity: MEDIUM

Observation:

- The application is using React Router version 7.5.0, which is a relatively recent release (as of early 2025), but may still contain unpatched or undiscovered security issues.
- Lack of updates increases the attack surface and risk of compromise.

Proof of Concept:



Impact:

React-Router (Framework mode) processes the `X-React-Router-SPA-Mode` header, forcing Server-Side Rendering (SSR) pages into Single-Page Application (SPA) mode. This triggers an unhandled error, corrupting the page response. If cached, the poisoned response serves the error to all users, causing a denial-of-service (DoS) via cache poisoning. The attack requires a page using a `loader` function and a misconfigured cache layer storing error responses.

Remediation:

Upgrade to React-Router's latest version and make sure that all the headers must be sanitized.

Web-07: X-Frame-options header not enforced

Path/pointer/location: Security Header ⟶ X-Frame-options header not enforced

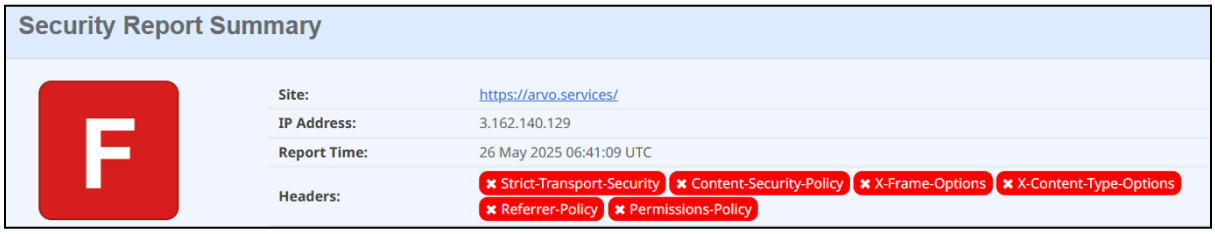
Severity: MEDIUM

CVSS: 6.3

Observation:

- The application does not enforce the X-Frame-Options header, which leaves it vulnerable to clickjacking attacks.
- In the absence of this header, the site can be embedded within a <frame> or <iframe> on a malicious site, potentially tricking users into performing unintended actions (e.g., clicking buttons or links without realizing it).
- This can lead to unauthorized actions, data leakage, or session hijacking under certain conditions.

Proof of Concept:



Impact:

Clickjacking, session hijacking, data exposure, and trust issues pose security and compliance threats. Implement header for protection.

- **(CVE-2022-3260)**

Remediation:

Implement X-Frame-Options header, use frame-busting scripts, and ensure strict iframe management to prevent these attacks effectively.

Web-08: Referrer-policy not enforced

Path/pointer/location: Security Header ⟶ Referrer-policy not enforced

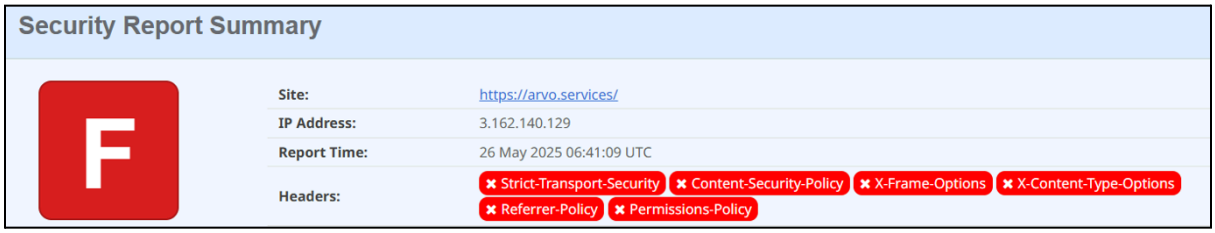
Severity: MEDIUM

CVSS: 5.3

Observation:

- The application does not enforce the Referrer-Policy HTTP header, which may lead to unintended leakage of sensitive information (such as full URLs, query parameters, or paths) to third-party websites via the Referrer header.
- Absence of this header weakens user privacy and increases the risk of information disclosure in cross-origin requests.

Proof of Concept:



Impact:

If a document's Referrer Policy attribute is set to "no-referrer" sometimes two network requests are made for elements instead of one. One of these requests includes the referrer instead of respecting the set policy to not include a referrer on requests.

- **(CVE-2017-7842)**

Remediation:

Implement a Referrer-Policy by using the Referrer-Policy response header or by declaring it in the meta tags. It's also possible to control referrer information over an HTML-element by using the rel attribute.

Web-09: Clickjacking

Path/pointer/location: Login Page

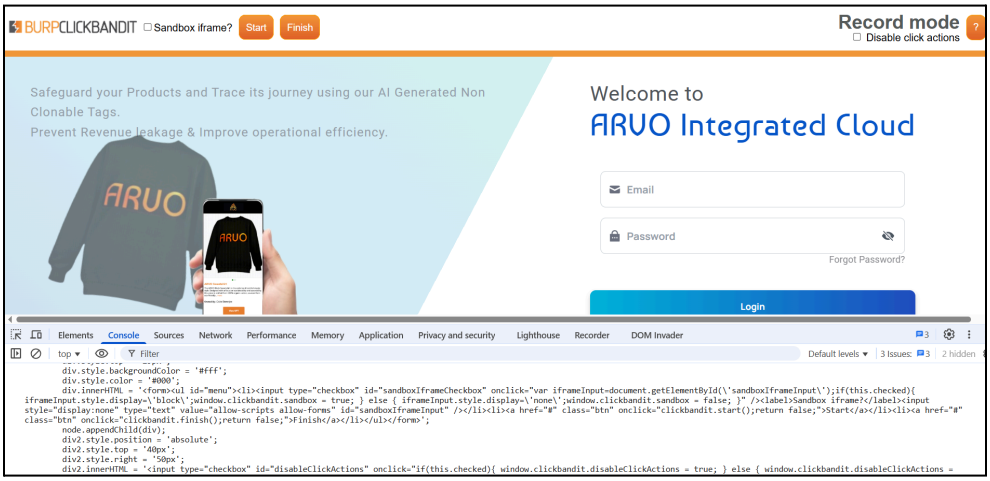
Severity: MEDIUM

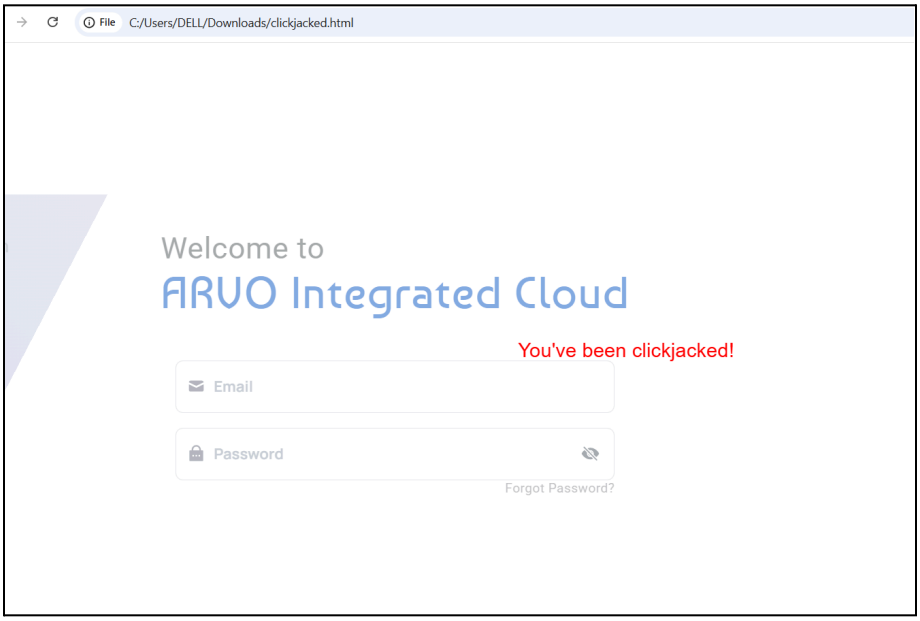
CVSS: 5.0

Observation:

- The application is vulnerable to clickjacking if it does not implement proper framing protections using security headers like X-Frame-Options or Content-Security-Policy (frame-ancestors).
- This attack can lead to unauthorized actions, such as changing settings, making transactions, or revealing sensitive information.

Proof of Concept:





Impact:

Clickjacking is an attack that occurs when an attacker uses a transparent iframe in a window to trick a user into clicking on an actionable item, such as a button or link, to another server in which they have an identical webpage.

- **(CVE-2021-35237)**

Remediation:

Sending the proper X-Frame-Options in HTTP response headers that instruct the browser to not allow framing from other domains.

Web-10: X-Content-type-options not enforced

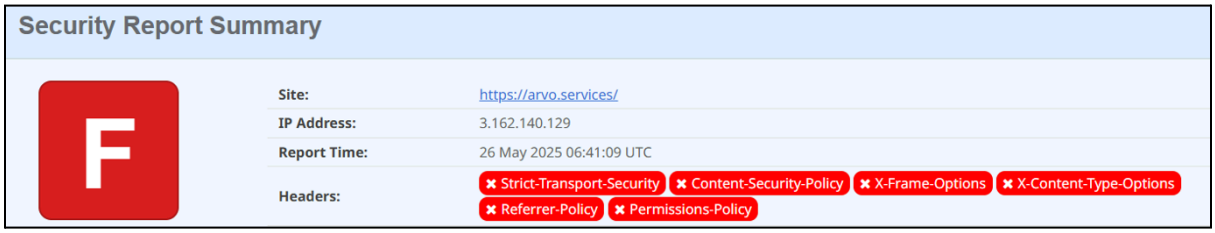
Path/pointer/location: Security Header ⟶ X-Content-type-options not enforced

Severity: MEDIUM

Observation:

- The application does not enforce the X-Content-Type-Options header, which can allow browsers to perform MIME type sniffing on responses.
- Without this header, a browser may attempt to guess the content type of a resource, even if the Content-Type is incorrectly declared.

Proof of Concept:



Impact:

Risks content misinterpretation, increasing vulnerabilities like content spoofing and potential security breaches.

Remediation:

Set 'X-Content-Type-Options' to 'nosniff' in the HTTP response headers to prevent MIME-type sniffing vulnerabilities. X-Content-Type-Options.

Web-11: Content security policy not enforced

Path/pointer/location: Security Header ⟶ Content security policy not enforced

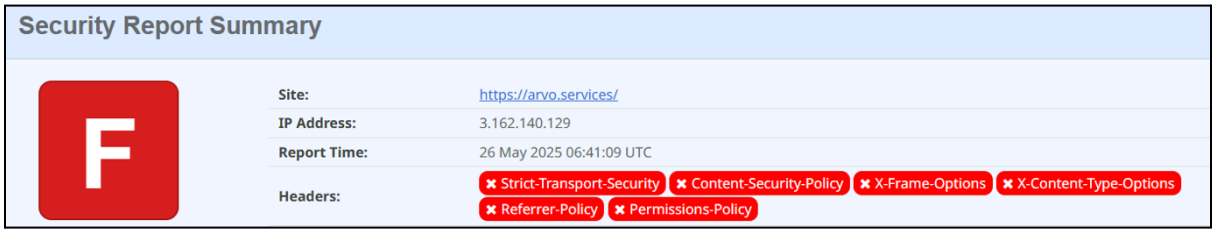
Severity: MEDIUM

CVSS: 6.3

Observation:

- The application does not enforce a Content-Security-Policy (CSP) header, which significantly weakens its defence against cross-site scripting (XSS), data injection, and content hijacking attacks.
- Without a CSP, browsers will accept and execute content from any source by default, which could lead to the execution of malicious code if an attacker injects a payload.

Proof of Concept:



Impact:

Content Security Policy (CSP) is not applied correctly to all parts of multipart content sent with the "multipart/x-mixed-replace" MIME type. This could allow for script to run where CSP should block it, allowing for cross-site scripting (XSS) and other attacks. This vulnerability affects Firefox < 60.

- **(CVE-2018-5164)**

Remediation:

By whitelisting sources of approved content, you can prevent the browser from loading malicious assets.

Web-12: No Rate Limit

Path/pointer/location: My Profile page ⟶ Reset password

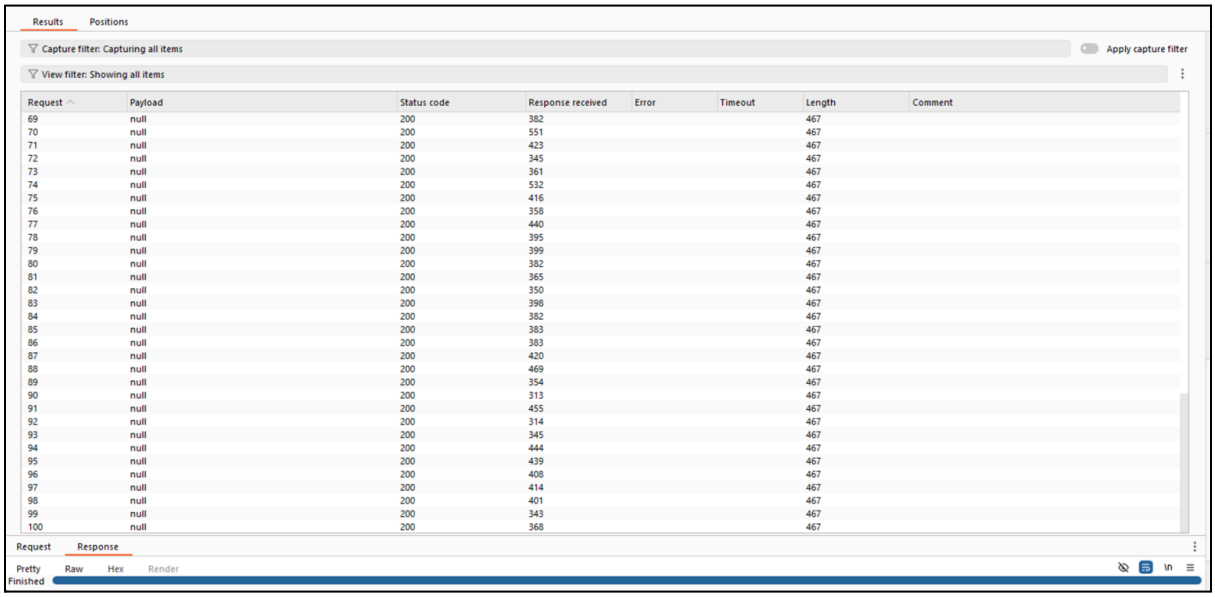
Severity: MEDIUM

Observation:

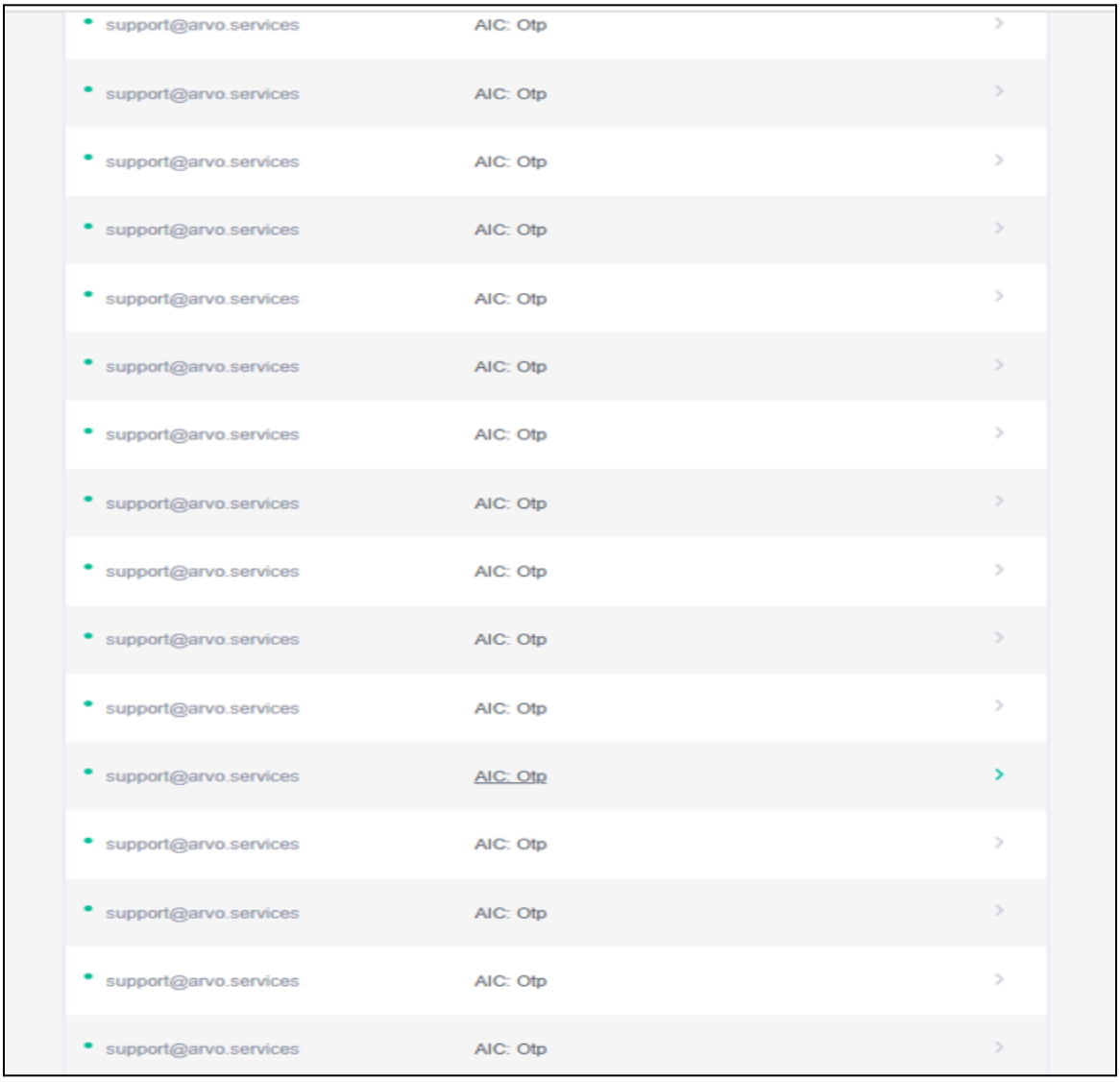
- It was observed that the application does not implement rate limiting on critical endpoints such as login or form submissions.
- As a result, attackers can send an unlimited number of requests without being blocked or throttled, enabling brute force attacks, spam, and abuse of functionality.

Proof of Concept:

1) Sending Request for 100 OTP's



2) Receiving 100 OTP's



Impact:

No rate limit vulnerabilities allow attackers to perform brute force, spam, or abuse actions without restriction, leading to account compromise and service disruption.

Remediation:

Implement rate limiting, CAPTCHAs, and monitoring to restrict excessive requests and detect abusive behaviour.

Web-13: Strict transport security not enforced

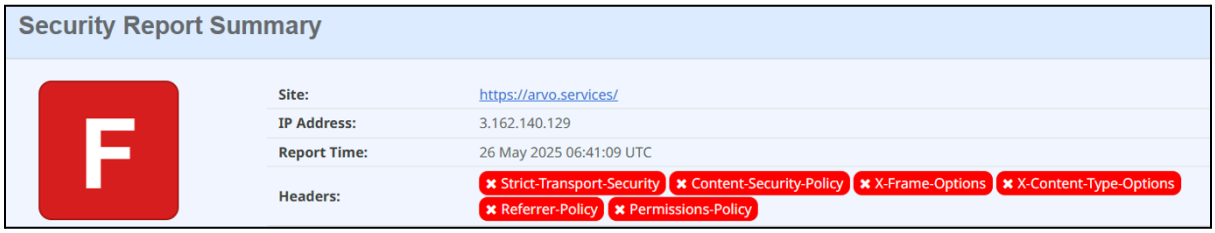
Path/pointer/location: Security Header ⟶ Strict transport security not enforced

Severity: Low

Observation:

- The lack of this header weakens transport layer security and user trust, especially for login forms, sensitive data, and authenticated sessions.

Proof of Concept:



Impact:

The application fails to prevent users from connecting to it over unencrypted connections. An attacker able to modify a legitimate user's network traffic could bypass the application's use of SSL/TLS encryption, and use the application as a platform for attacks against its users. This attack is performed by rewriting HTTPS links as HTTP, so that if a targeted user follows a link to the site from an HTTP page, their browser never attempts to use an encrypted connection.

Remediation:

The application should instruct web browsers to only access the application using HTTPS. To do this, enable HTTP Strict Transport Security (HSTS) by adding a response header with the name 'Strict Transport-Security' and the value 'max-age=expireTime', where expireTime is the time in seconds that browsers should remember that the site should only be accessed using HTTPS. Consider adding the 'includeSubDomains' flag if appropriate.

Web-14: Permissions-Policy not enforced

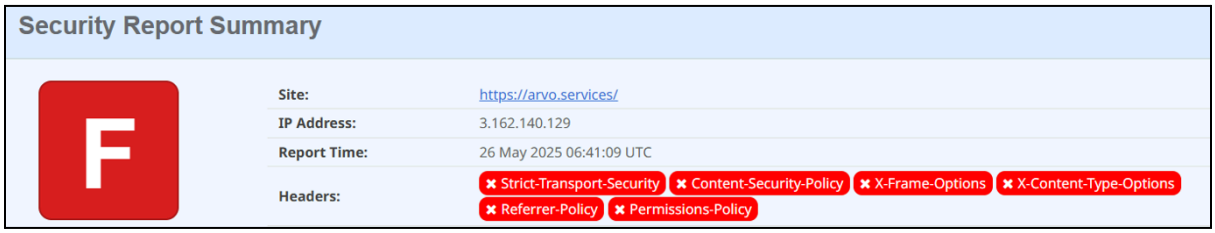
Path/pointer/location: Security Header ⟶ Permissions-Policy not enforced

Severity: Low

Observation:

- Without this policy, all supported features may be available by default, even if they are not required, which increases the attack surface and risks unauthorized access to sensitive device capabilities.
- The absence of this header may lead to privacy issues and unintended behaviours, especially in cross-origin iframes or embedded content.

Proof of Concept:



Impact:

The Permissions-Policy header allows developers to selectively enable and disable use of various browser features and APIs.

Remediation:

Enforce Permissions-Policy headers, limit web features, and enhance website security

Web-15: Unrestricted File Upload

Path/pointer/location: Sign in Page ⟶ Register

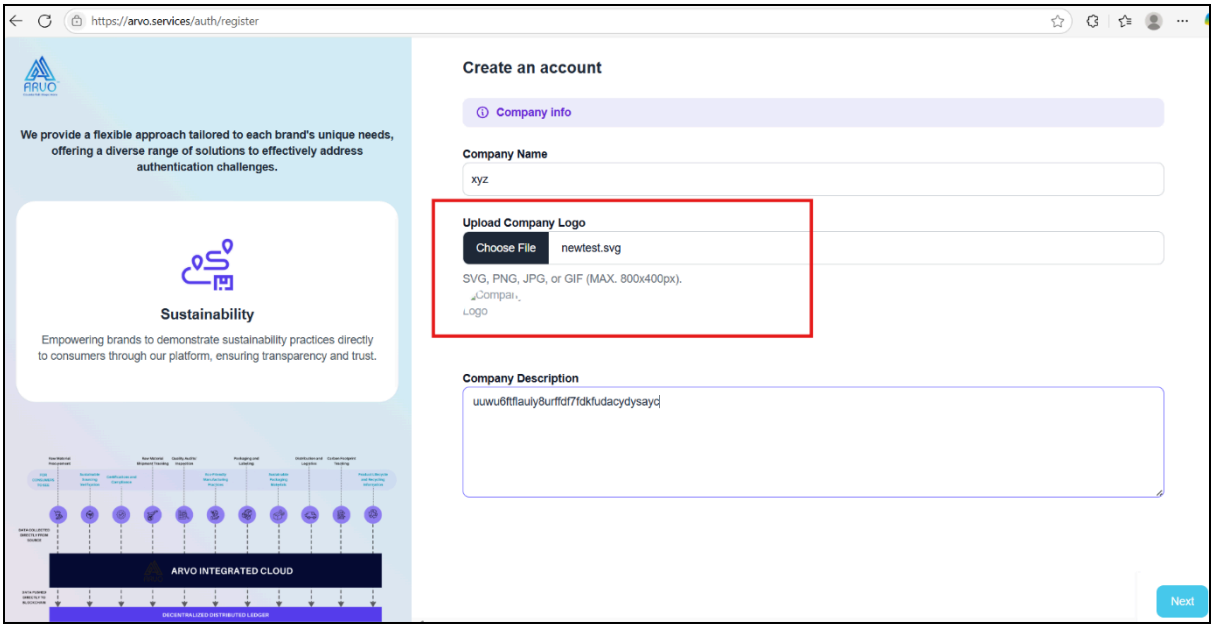
Severity: Low

Observation:

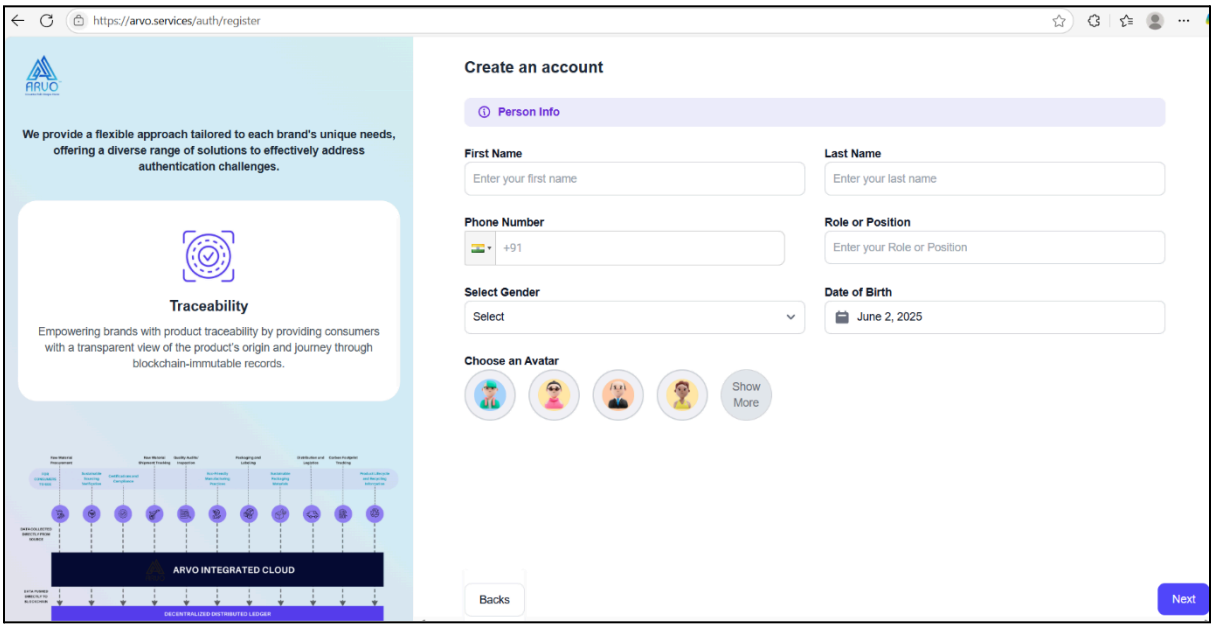
- It was observed that the application allows file uploads without properly validating file type, content, or extension.
- This lack of restriction enables attackers to upload malicious files, which can be executed on the server, potentially leading to remote code execution and full system compromise.

Proof of Concept:

1) File upload



2) File upload successful and moved to the next page



Impact:

Unrestricted file uploads can lead to severe security breaches, enabling attackers to upload and execute malicious files on the server. This can result in data breaches, remote code execution, and potentially complete compromise of the web application and server.

Remediation:

To mitigate unrestricted file upload risks:

● Implement File Type Validation: Restrict allowed file types to a specific whitelist and reject any other file types.

● Implement File Size Limitations: Set size limits for uploaded files to prevent denial of service attacks and conserve server resources.

● Store Uploaded Files Securely: Save uploaded files outside the web root directory, ensuring they can't be executed.

Web-16: Improper session management

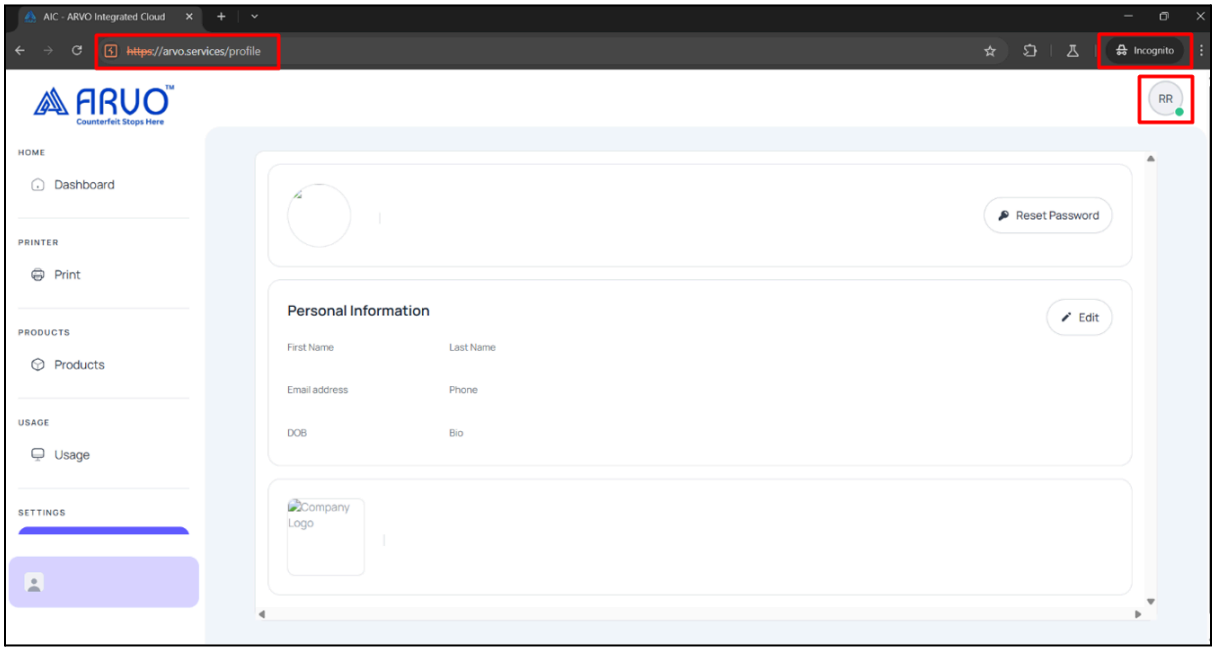
Path/pointer/location: arvo.services/profile

Severity: Low

Observation:

- It was observed that the /profile endpoint is accessible without authentication. Even after logging out, unauthenticated users can still view random user profiles, resulting in unauthorized access to sensitive personal information and serious privacy violations.

Proof of Concept:



Impact:

Unauthenticated users can access random user profiles via the /profile endpoint after logout, leading to serious data leakage and privacy violations.

Remediation:

Enforce strict authentication checks on protected routes and ensure server-side session tokens are properly invalidated upon logout.

Web-17: Excessive Server Information Exposure

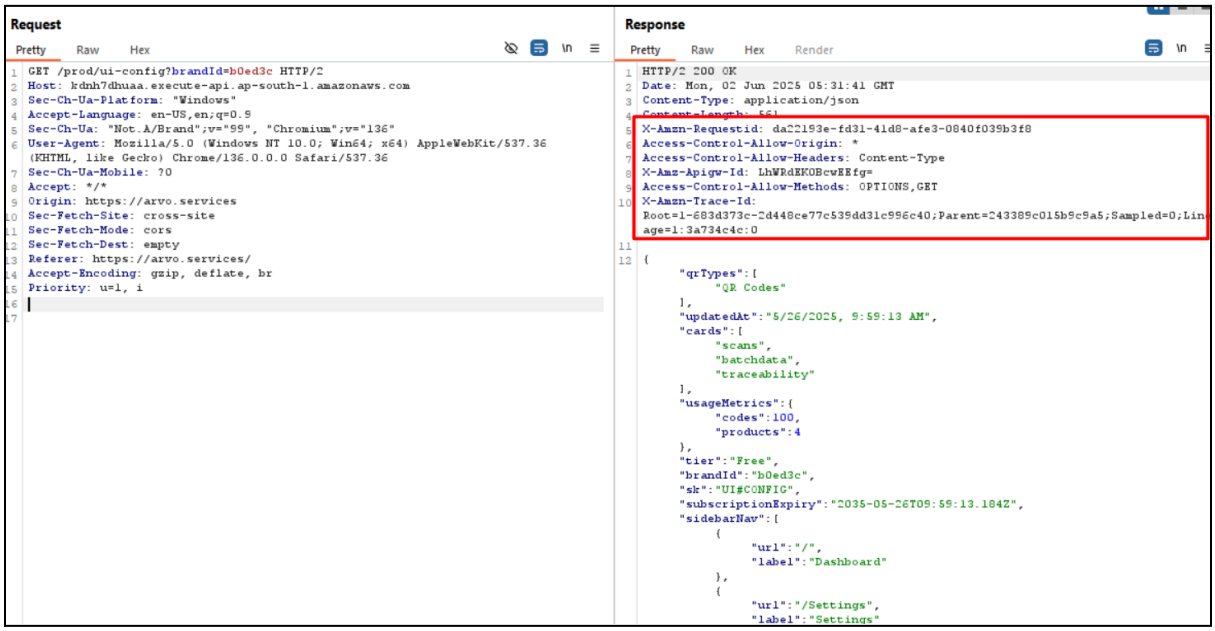
Path/pointer/location: Login Page Packet

Severity: Low

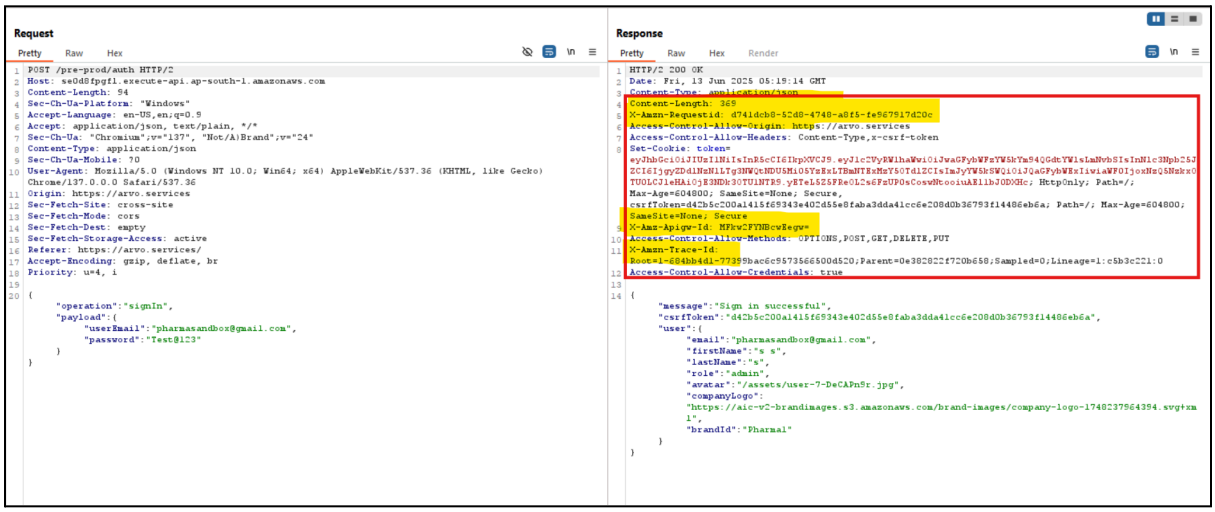
Observation:

- In almost all the packages of ARVO website the sensitive API ID's are Displaying.
- It was observed that the application exposes internal headers such as X-Amzn-RequestId, X-Amz-Apigw-Id, and X-Amzn-Trace-Id in responses. These headers reveal internal system identifiers that could assist attackers in reconnaissance and potentially facilitate targeted attacks.

Proof of Concept:



After Patch (13/06/2025)



Impact:

Exposing headers like X-Amzn-RequestId, X-Amz-Apigw-Id, and X-Amzn-Trace-Id can provide attackers with internal identifiers that aid in reconnaissance, debugging, or chaining with other vulnerabilities for targeted attacks.

Remediation:

Configure the server or API gateway to suppress unnecessary internal headers in production responses, and ensure only essential information is exposed to end users.

CONCLUSION:

The Vulnerability Assessment and Penetration Testing (VAPT) of the ARVO web application identified a few areas of improvement, mainly concerning configuration settings, component updates, and authentication mechanisms. We are pleased to report that all critical vulnerabilities have been successfully remediated, along with the majority of high, medium, and low-severity findings—demonstrating the team's strong commitment to security. These proactive measures have notably enhanced the overall security posture of the application. To ensure continued protection against evolving threats, we recommend ongoing improvements and regular security assessments.

DISCLAIMER

SecureDApp provides clear and actionable cybersecurity assessments of web applications, APIs, cloud platforms, mobile apps, and enterprise systems. The findings in this report are based on industry-standard vulnerability assessment and penetration testing methodologies. While the report identifies known security risks and configuration issues, it makes no claim of guaranteeing absolute security or the complete absence of vulnerabilities.

This report is intended solely for the client's internal use. SecureDApp does not make any warranties or representations—express or implied—regarding the accuracy, completeness, reliability, or suitability of the information contained herein. The audit is limited to the scope defined at the time of engagement and does not extend to software dependencies, third-party services, user devices, or components outside of the testing environment.

Cybersecurity is inherently complex, and all assessments carry a degree of uncertainty. As such, this report may contain false positives or false negatives and should not be considered exhaustive. The client acknowledges and accepts that use of this report, and any associated materials or services, is at their own risk and on an "as-is," "where-is," and "as-available" basis.

CONFIDENTIALITY

This report is confidential and subject to the terms outlined in the service agreement between SecureDApp and the client. Distribution, disclosure, or reliance by any third party is strictly prohibited without prior written authorization from SecureDApp. All content within this report is the intellectual property of SecureDApp and is intended solely for the use of the intended recipient.

NO FINANCIAL OR LEGAL ADVICE

This cybersecurity assessment report is not an endorsement of any product, service, organization, or technology. It is provided purely for informational and technical purposes. It should not be interpreted as financial, legal, investment, or regulatory advice. SecureDApp does not accept liability for decisions made by third parties based on this report.

TECHNICAL DISCLAIMER

SecureDApp disclaims all warranties, express or implied, including but not limited to warranties of merchantability, fitness for a particular purpose, and non-infringement. We do not guarantee that our findings will meet the client's specific goals, be error-free, or detect every potential threat or vulnerability. Compatibility, performance, and behavior of systems outside the testing scope are not covered.

TIMELINESS OF CONTENT

This report reflects the state of the application and infrastructure at the time of testing. SecureDApp does not commit to continuous updates or tracking of changes after delivery of the report. Security is a constantly evolving field, and new threats may arise at any time.

THIRD-PARTY LINKS

This report may reference third-party websites or tools. These links are provided for informational purposes only. SecureDApp is not responsible for the accuracy, functionality, or content of any external site and disclaims all liability for their use.

ABOUT SECURE DAPP

SecureDApp is a cybersecurity solutions provider focused on helping businesses safeguard their digital infrastructure through comprehensive security assessments and proactive threat mitigation. We specialize in identifying vulnerabilities across web applications, mobile apps, APIs, cloud platforms, and enterprise systems.

Our services include Vulnerability Assessments, Penetration Testing (VAPT), Secure Code Reviews, Security Architecture Consulting, DevSecOps integration, and threat modeling. Whether you are a startup or an enterprise, SecureDApp helps you build and maintain a resilient security posture.

Our team consists of seasoned security professionals, ethical hackers, engineers, and compliance experts distributed globally. We combine manual and automated testing methodologies to uncover potential weaknesses, misconfigurations, and security gaps before they can be exploited.

At SecureDApp, our mission is to deliver scalable, high-quality, and easy-to-integrate security services that empower organizations to operate safely and confidently in today's complex digital landscape.

To learn more, visit : https://securedapp.in/

To view our audit portfolio, visit : github.securedapp.in

To book an audit, message : securedapp.telegram

Securedapp.in

Securedapp_Linkedin

Securedapp_Telegram