Web Application Penetration Testing Report

Report For :

Organization Name : Gujarat Informatics Ltd.

Report Date : August 31st 2024

Report By :

Organization Name : Xiarch Solutions Pvt Ltd

Report Date : August 31st 2024

Address : 403-404, Tower A , Spaze Edge, Sohna Road, Gurugram, Haryana 122018

Tel : +91-9667916333

Email : info@xiarch.com

Web : www.xiarch.com

Disclaimer

All the information contained in this document is confidential to said company, disclosure and use of any information contained in this document by photographic, electronic or any other means, in whole or part, for any reason other than security enhancement is strictly prohibited without written consent of auditee organization.

Whilst all due care and diligence has been taken in the preparation of this document it is not impossible that document of this nature may contain errors or omissions because of a misunderstanding of Clients requirements. Any recommendations are made in good faith as guidelines to assist the client in evaluation and must not be construed as warranties of any kind. Findings in this report are based on various tests conducted using manual techniques and third-party tools and Xiarch Solutions Pvt Ltd has put its best efforts to eliminate all the false positives reported by these tools.

Xiarch Solutions Pvt Ltd shall assume no liability for any changes, omissions, or errors in this document. Xiarch Solutions Pvt Ltd shall not be liable for any damages financial or otherwise arising out of use/misuse of this report by any general member of public

.

Table Of Content

1. Document Control

This document serves as a comprehensive report of the findings and recommendations resulting from the web application security assessment performed on Gujarat Panchayat Service Selection Board(Gujarat Informatics Ltd.) by Xiarch Solutions Private Limited. This report is intended for Management, and is to be used as a guide for improving the security posture of the web application.

1.1 Document Version Control

Version	Report	Person	Action	Date
1.0	Level 01	Manish Gupta	Audit	31/08/2024
1.1	Level 01	Nitin Panwar	Review	31/08/2024
1.2	Level 01	Ronak Bal	Final Approval	31/08/2024

1.2 Document Distribution List

Name	Designation	Organization
Manish Gupta	Associate InfoSec Consultant	Xiarch Solutions Private Limited
Nitin Panwar	InfoSec Consultant	Xiarch Solutions Private Limited
Ronak Bal	Senior InfoSec Consultant	Xiarch Solutions Private Limited

2. Introduction

2.1 Summary

This report outlines the results of a comprehensive web application security assessment conducted for the Gujarat Panchayat Service Selection Board web application. The purpose of this assessment was to identify any potential security vulnerabilities and to provide recommendations for their mitigation.

The assessment was conducted using a combination of manual and automated testing techniques, including vulnerability scanning, penetration testing, and code review. The scope of the assessment covered the application's front-end and back-end components, as well as any supporting infrastructure.

The assessment was conducted in accordance with industry-standard security best practices and guidelines, including the Open Web Application Security Project (OWASP) Top 10 list of web application security risks.

The results of the assessment identified a number of security vulnerabilities, which have been categorized according to their severity and impact on the application's security posture. Recommendations for mitigating these vulnerabilities have been provided, along with a roadmap for addressing any remaining security issues.

The findings and recommendations contained in this report are intended to assist the development team in improving the overall security of the web application, and to help ensure that it meets the necessary security standards and compliance requirements.

2.2 Assessment Objective:

- Identify and assess security flaws in mobile application according to industry principal security standards like OWASP Web Security Project Top 10 and SANS 25 etc.
- Provide recommendations for mitigation of risk(s) emerged during the identified vulnerabilities.

3. Scope

3.1 Scope Details

The security assessment was carried out in the pre-production environment and it included the following scope:

Application Name	**Panchayat Web Application**
URL	https://panchayat.gipl.in/ https://panchayat.gipl.in/workflow
User IDs and Access	Username- DDO_AMR, Panchayat , Password: *********
Testing Environment Details	Testing Environment
Application Description	**Panchayat** Web application

For Employee	
Modules	**Sub module**
Login	N/A
Dashboard	N/A
Edit Details	N/A
Logout	N/A

Out of Scope	
Module	**Sub module**
N/A	N/A

Assumptions:

- No changes were made during assessment by project team, all artefacts/documents shared with us are latest as on shared date and are unaltered.
- This vulnerability assessment exercise did not include the testing of vulnerabilities leading to Denial of service (DoS) and similar attacks.

- Appropriate backup and rollback plan are made prior to implementing the recommendation on the system.
- Vulnerabilities identified were as on the data scan conducted and as per the scan policies (non-intrusive) & plugins selected. There may be vulnerabilities which may exist & not assessed since their exploits may lead to system downtime. Also, the vulnerabilities identified after the scan date may also not form part of this report.

3.2 Post Assessment Clean-up

Any test accounts, which were created for the purpose of this assessment, should be disabled or removed, as appropriate, together with any associated content.

4. Terms , Clarity & Legends

This section describes the format in which the identified vulnerabilities are reported in the later section of the report. "Vulnerability Table" shown below is used to provide the details of the vulnerability, its impact and the recommendations.

4.1 Vulnerability Table

Observation ID & Title :			
CVSS Risk Rating :		**Status :**	
Observation Details :			
Risk/Impact :			
Recommendations :			
Affected URL & Parameter :			
CVSS Vector :			

Title of the Vulnerability	-	A short title that describes the vulnerability
Risk Level	-	It describes the risk level. The title bar of each vulnerability table is colour coded for quick identifications of the severity level of the vulnerabilities
Description	-	It provides a brief description of the vulnerability.
Impact	-	Describes the probable impact if the vulnerability is successfully exploited.
Recommendations	-	Provide the recommendations to fix the vulnerability.
Affects	-	Provide the information where vulnerability is present.
Status	-	Provides the information whether the vulnerability is closed or not.

4.2 Findings Ranking System

In order to prioritize the assessment results, each finding was categorized based on severity classifications. Final analysis of the risk or impact to the application will require an internal evaluation. Xiarch Labs has developed classifications using the severity nomenclature for ranking the issues identified within the various severity categories.

4.2.1 Severity Categories

Based on Xiarch Lab analysis of the particular finding and assets affected, a finding will fall into one of the following severity level categories:

Critical Vulnerabilities require an immediate response through mitigating controls, direct remediation or a combination thereof. Exploitation of critical severity vulnerabilities results in privileged access to the target system, application or sensitive data and enables further access to other hosts or data stores within the environment. In general, a critical severity ranking is warranted when the issue has a direct impact on regulatory or compliance controls imposed on the environment, accesses personally identifiable information (PII) or financial data or could cause significant reputational or financial harm.

High Findings with a high severity ranking require immediate evaluation and subsequent resolution. Exploitation of high severity vulnerabilities leads directly to an attacker gaining privileged, administrative-level access to the system, application or sensitive data. However, it does not enable further access to other hosts or data stores within the environment. If left unmitigated, high severity vulnerabilities can pose an elevated threat that could affect business continuity or cause significant financial loss.

Medium A finding with a medium severity ranking requires review and resolution within a short time. From a technical perspective, vulnerabilities that warrant a medium severity ranking can lead directly to an attacker gaining non-privileged or user-level access to the system, application or sensitive data. Findings that can cause a denial-of-service (DoS) condition on the host, service or application are also classified as medium risk. Alternately, the vulnerability may provide a way for attackers to gain elevated levels of privilege. From a less technical perspective, observations with this ranking are significant, but they do not pose as much of a threat as high or critical severity exposures.

Low Low severity findings should be evaluated for review and resolution once the remediation efforts for critical, high and medium severity issues are complete. From a technical perspective, vulnerabilities that warrant a low severity ranking may leak information to unauthorized or anonymous users used to launch a more targeted attack against the environment.

Informational An informational finding presents no direct threat to the confidentiality, integrity or availability of the data or systems supporting the environment. These issues pose an inherently low threat to the organization and any proposed resolution should be considered as an addition to the information security procedures already in place.

5. Assessment Methodology

A penetration testing methodology for web apps typically includes the following steps:

Probe: Involves gathering information about the target system, such as the target's IP address, server type, operating system, and other details that can help identify vulnerabilities.

Enumeration: In this phase, the tester tries to identify any open ports, services or applications that may be running on the target system. This can be done using various tools such as port, network or web application scanners.

Vulnerability Scanning: Once the tester has identified the services running on the target system, they can use vulnerability scanners to identify any known vulnerabilities or weaknesses that can be exploited.

Exploitation: After identifying vulnerabilities, the tester attempts to exploit them to gain unauthorized access to the target system or its data. This can be done using various tools and techniques such as SQL injection, cross-site scripting or brute force attacks.

After Exploitation: After gaining access to the system, the tester attempts to maintain access and escalate privileges to gain more control over the target system.

Reporting: Finally, the tester documents his findings and provides the client with recommendations for remediation. This report usually contains a detailed description of the vulnerabilities found, their severity, and recommended steps to mitigate them.

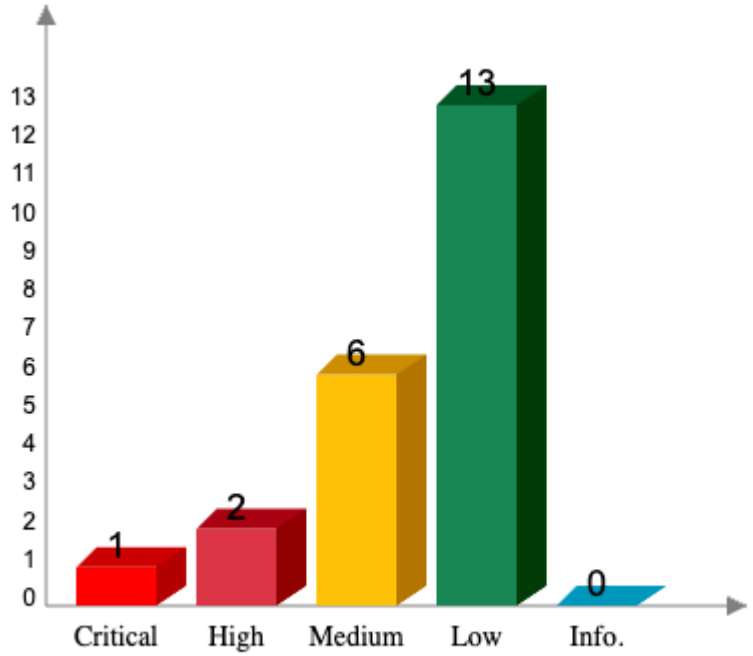
It is important to note that this methodology may vary depending on the specific target being tested and the objectives of the test.

6. Executive Summary

Xiarch conducted a comprehensive security assessment of **Gujarat Informatics Ltd.** in order to determine existing vulnerabilities and establish the current level of security risk associated with the environment and the technologies in use. This assessment harnessed penetration testing and vulnerability assessment techniques to provide **Gujarat Informatics Ltd.** management with an understanding of the risks and security posture of their corporate environment.

6.1 Application Health

Severity Stats



Bar Chart - Severity Stats

7. Assessment Findings Overview

The table below provides a summary of the assessment findings categorized by group and ranked by severity. The table provides an overview of all of the findings from the assessment and allows the remediation team to focus efforts on the areas of highest severity as determined by Xiarch Labs. Click the individual link below to go directly to that finding.

S. No.	Vulnerability Title	Severity	Status
1	Improper Access Control	Critical	Open
2	No Rate Limit - Account Bruteforce	High	Open
3	Broken Authentication	High	Open
4	Improper Cache Control	Medium	Open
5	Improper Error handling	Medium	Open
6	Vulnerable JavaScript Library	Medium	Open
7	Missing Security Header	Medium	Open
8	SQL Possible	Medium	Open
9	TLS Fallback SCSV Support	Medium	Open
10	Banner Grabbing	Low	Open
11	No Session Timeout	Low	Open
12	Dangerous Method Enabled	Low	Open
13	HTTP Strict Transport Security	Low	Open
14	Weak lock out mechanism	Low	Open
15	AutoComplete Enabled	Low	Open
16	Weak password policy	Low	Open
17	Missing secure Flag Not Set	Low	Open
18	Improper Session Expiry	Low	Open
19	Clickjacking	Low	Open
20	TLS 1.0 And TLS 1.1 Enabled	Low	Open
21	Sweet32 Attack	Low	Open
22	Improper Input Validation	Low	Open

8. Assessment Findings Details

Observation ID & Title	8.1 Improper Access Control		
CVSS Risk Rating	Critical	Status	Open
Observation Details			
It is been observed that the account can be open directly without providing the credentials in the website. As per observation, user logged in first in the account and logout the account. Now attacker just opened the account with the dashboard URL of a account which comes after logging in the account.			
Risk			
It is possible to takeover the account without providing the credentials. This can be used to access the account number of times and attacker can change or edit the details of the user.			
Recommendations			
The Following are the recommendations that should be implemented. 1. Continuous Inspection and Testing Access Control 2. Limiting CORS Usage 3. Proper authentication should be done in the application.			
Affected URL & Parameter			
Throughout the Application			
CVSS Vector			
9.0 CVSS:3.0/AV:L/AC:L/PR:N/UI:N/S:C/C:L/I:H/A:L			

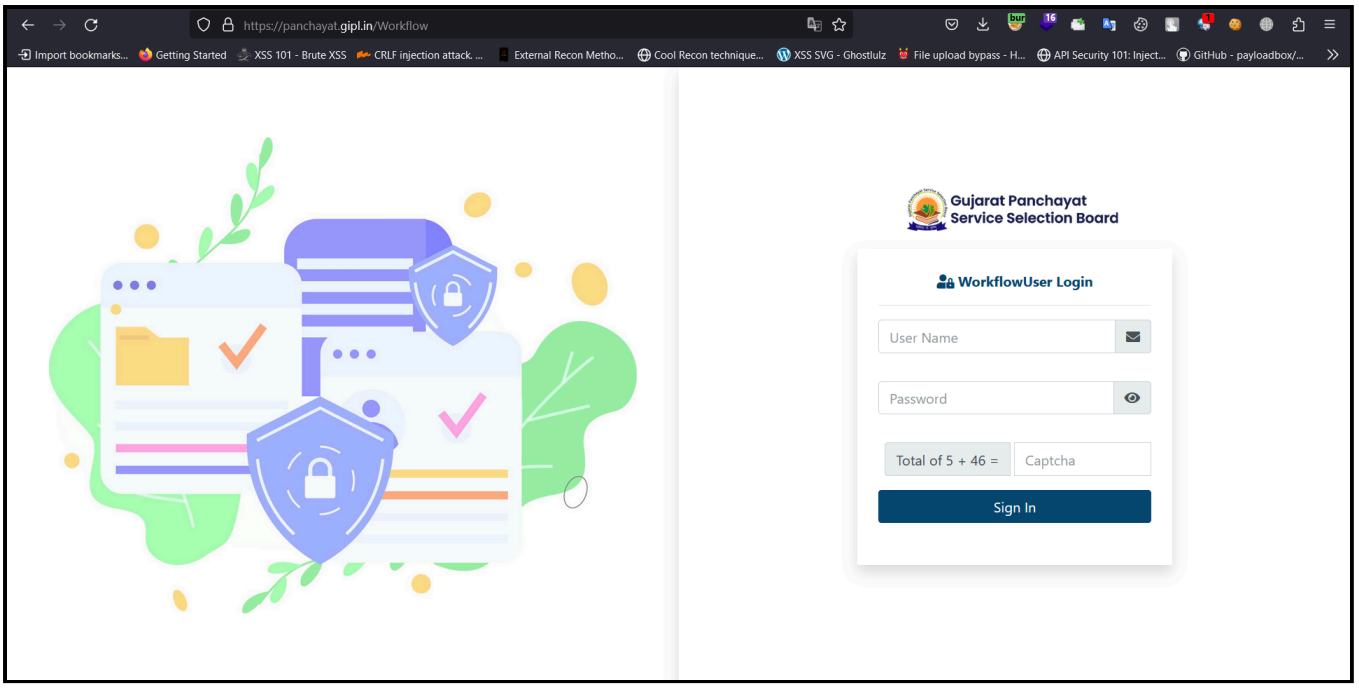


Figure 1 - Go to the mentioned URL

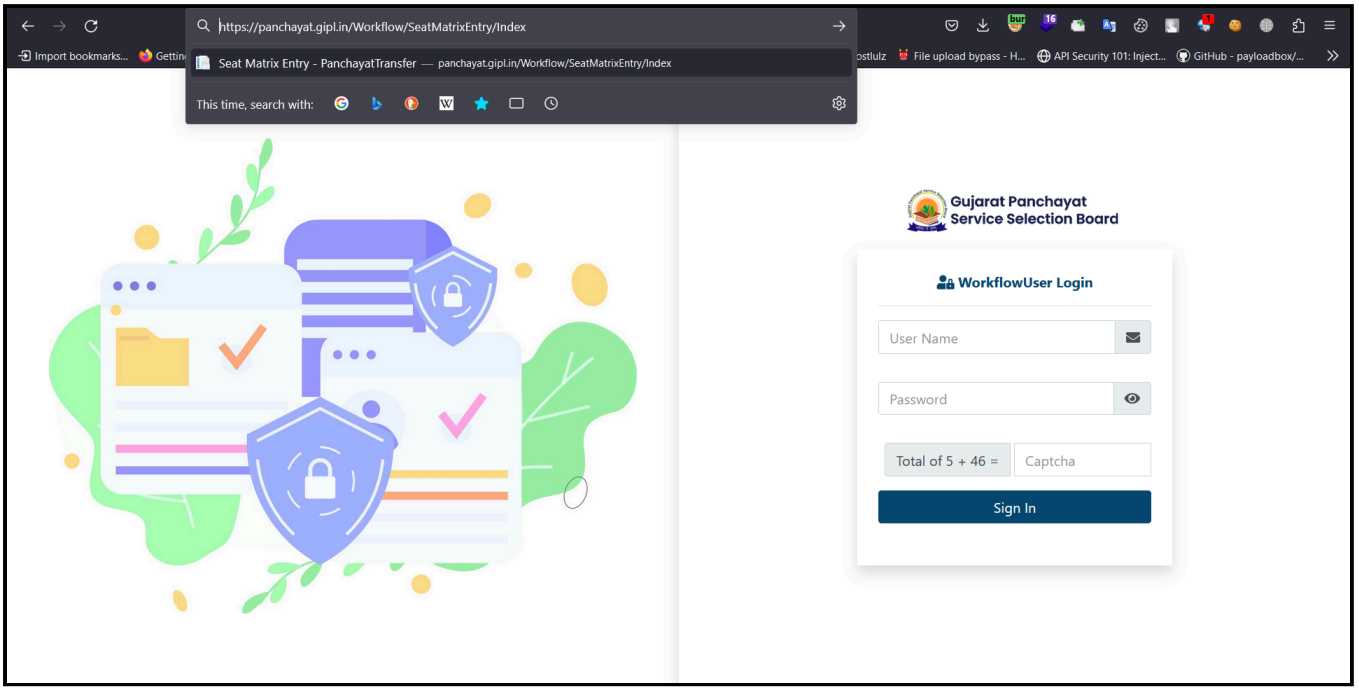


Figure 2 - Paste the URL that comes after Logging in the account.

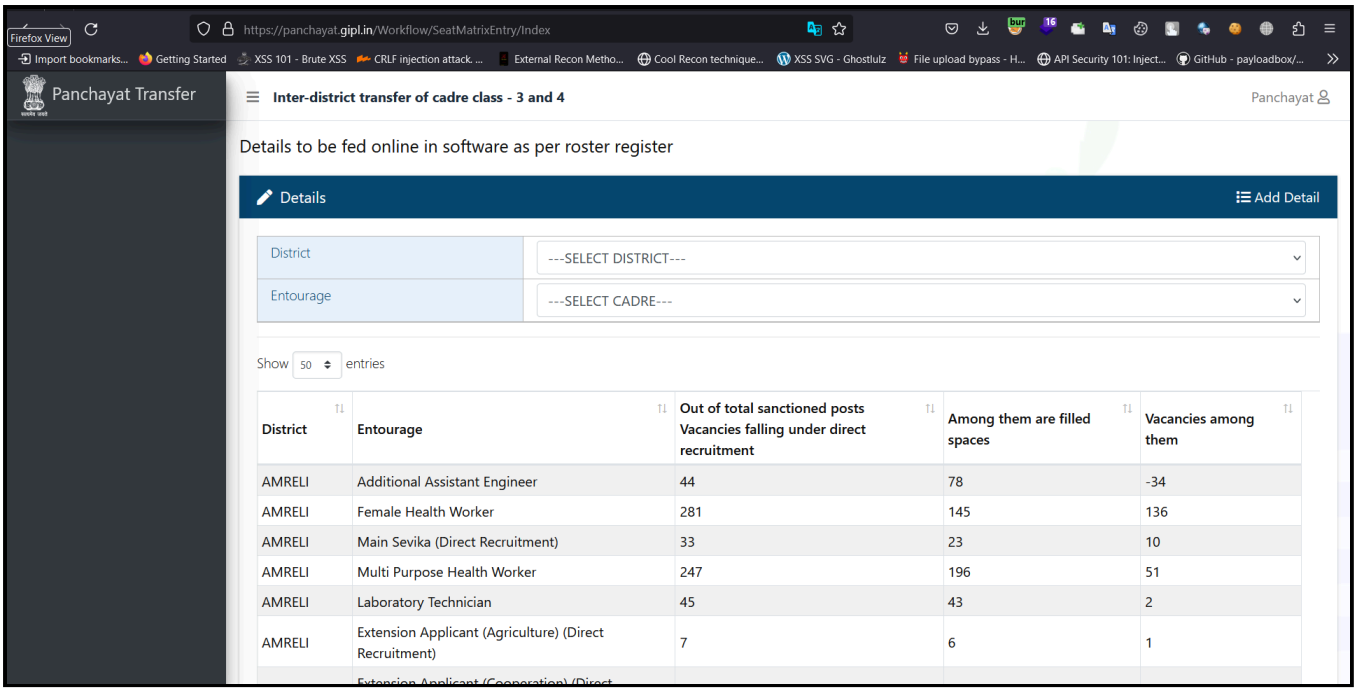


Figure 3 - As you can see that the attacker get the access of the user.

Observation ID & Title	8.2 No Rate Limit - Account Bruteforce		
CVSS Risk Rating	High	Status	Open

Observation Details

Rate limiting is a process to limit requests possible. It is used in web applications in order to prevent brute force attacks. This is necessary to prevent the attackers from logging in accounts by brute forcing credentials or even sending excessive requests to the server.

No rate limit is a flaw that doesn't limit the no. of attempts one makes on a website server to extract data. It is a vulnerability which can prove to be critical when misused by attackers.

Risk

• Identity theft – stealing someone's identity to access their accounts, such as bank accounts or credit cards. This enables the attacker to purchase goods using these details. In addition, information such as social security numbers can be sold for use in other cyber-attacks.
• Loss of data – due to loss of confidentiality if data is stolen which could destroy company reputation. Additionally, there may be reputational damage caused by a leak of sensitive customer information that leads to public distrust and dissatisfaction with the business.
• Downtime – this refers to system outages where websites or computer networks cannot be accessed due to a cyber-attack. This is costly to the business in terms of lost revenue, customer satisfaction as well as loss of image.

Recommendations

The Following are the recommendations that should be implemented.
• Never use information that can be found online (like names of family members).
• Have as many characters as possible.
• Combine letters, numbers, and symbols.
• To protect your organization from brute force password hacking, enforce the use of strong passwords.
• Be different for each user account.
• Avoid common patterns.
• Lockout policy
• Progressive delays
• Authorization should be properly validated from the server side.

Affected URL & Parameter

Throughout the Application

CVSS Vector

8.5 CVSS:3.0/AV:L/AC:L/PR:N/UI:N/S:C/C:L/I:H/A:L

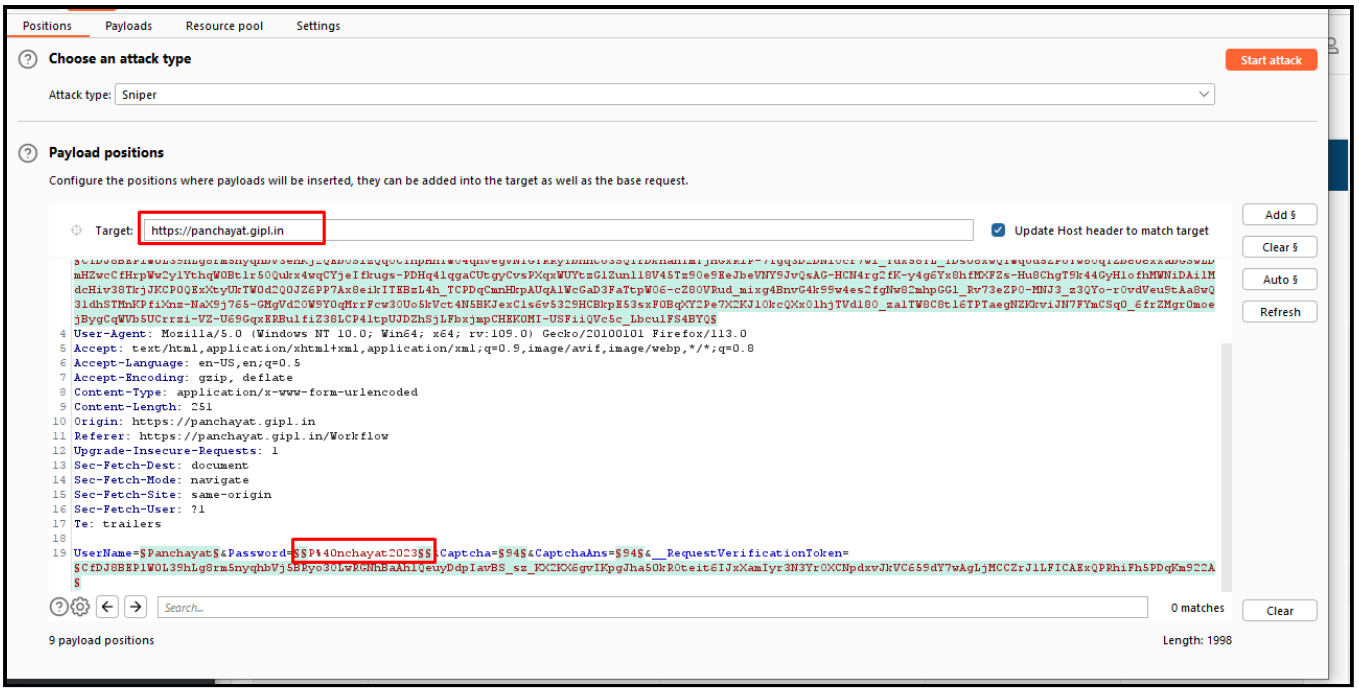


Figure 4 - Intercept the login request > Send the request to Intruder. Select the password as a attack vector.

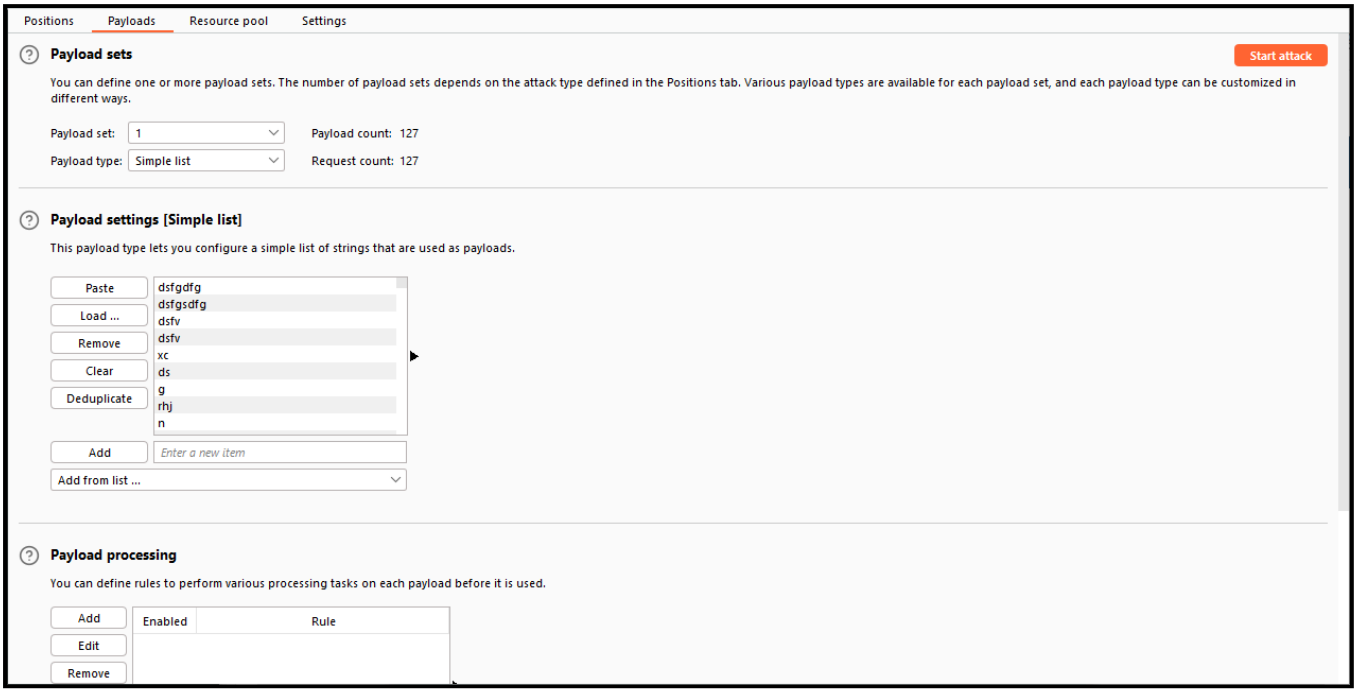


Figure 5 - Select simple list > Upload the wordlist for the attack.

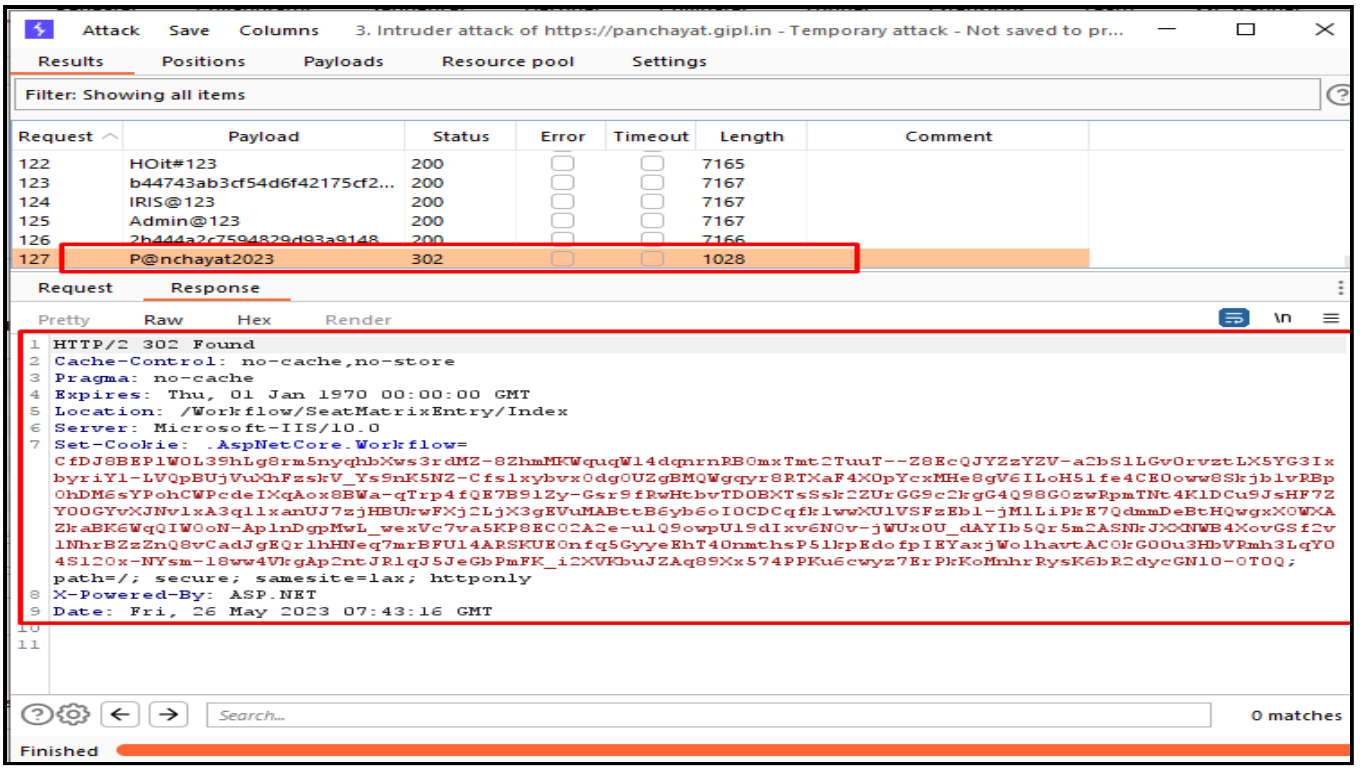


Figure 6 - Start the attack > you will get the password as response of Status 302 Found.

Observation ID & Title	8.3 Broken Authentication		
CVSS Risk Rating	High	Status	Open

Observation Details

The application does not properly invalidate a user's session on the server after the user initiates logout. User sessions remain active on the server, and any requests submitted including the user's session identifier will execute successfully, as though the user had made those requests.

Risk

An attacker can use previous used or available session token to change anything in the user account without logging in the user account.

Recommendations

The Following are the recommendations that should be implemented.

The user's HTTP session should be terminated on the server immediately after a logout action is performed. It is important to note that simply deleting the cookie from the browser will not terminate the server session. The session must be invalidated at the server, using the HTTP container's intrinsic session abandonment mechanism.

Affected URL & Parameter

Throughout the Application

CVSS Vector

8.0 CVSS:3.0/AV:L/AC:L/PR:N/UI:N/S:C/C:L/I:H/A:L

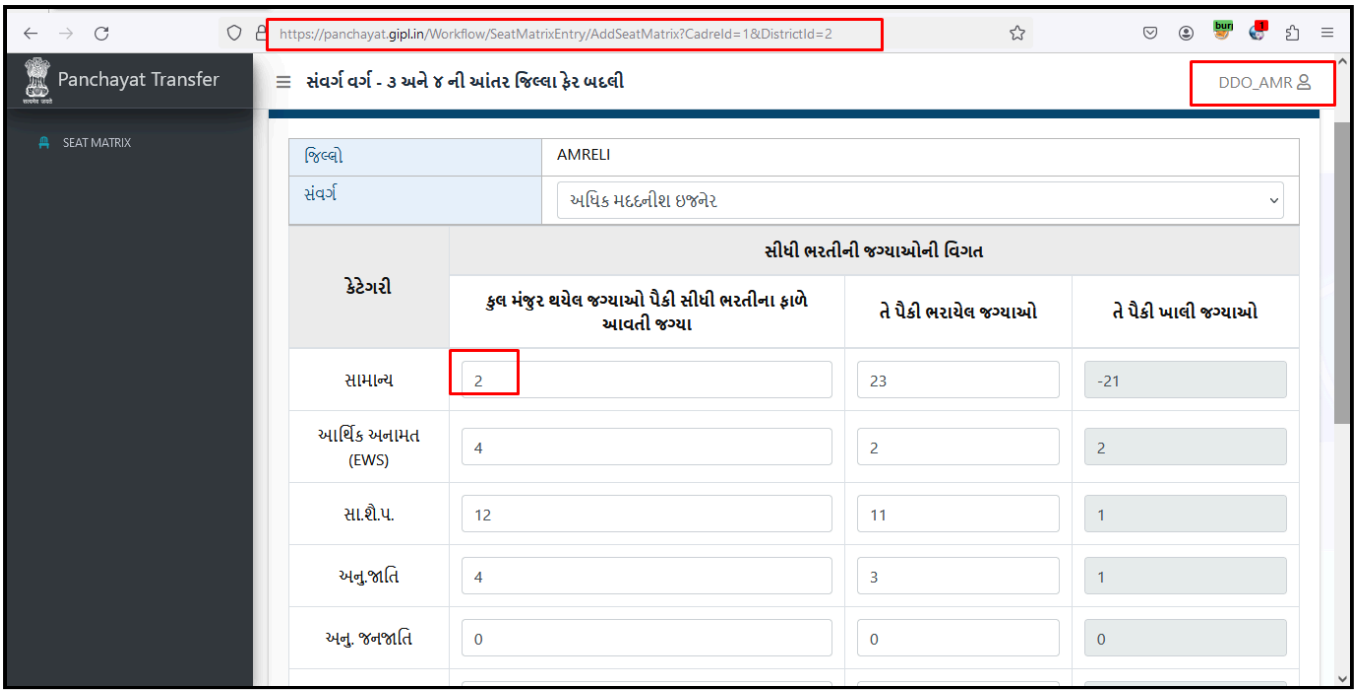


Figure 7 - Intercept the request by clicking on save in the Edit Details.

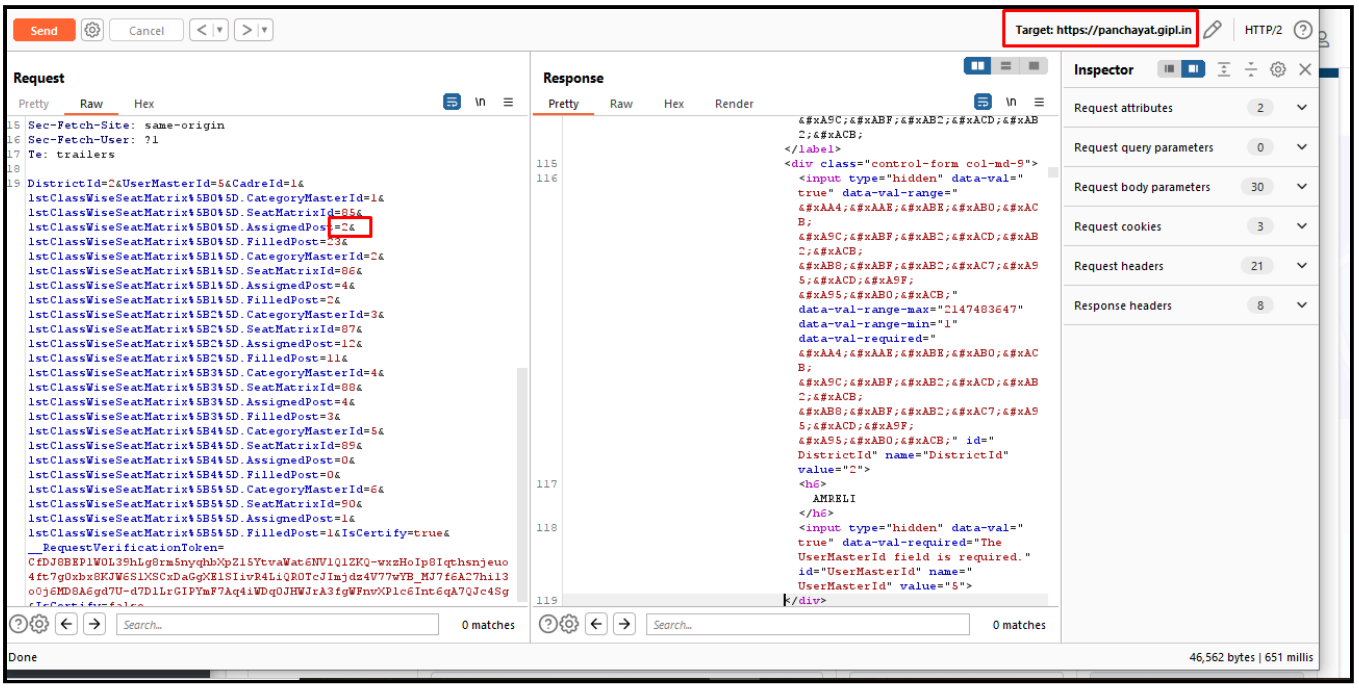


Figure 8 - Send the request to Repeater > Logout the account.

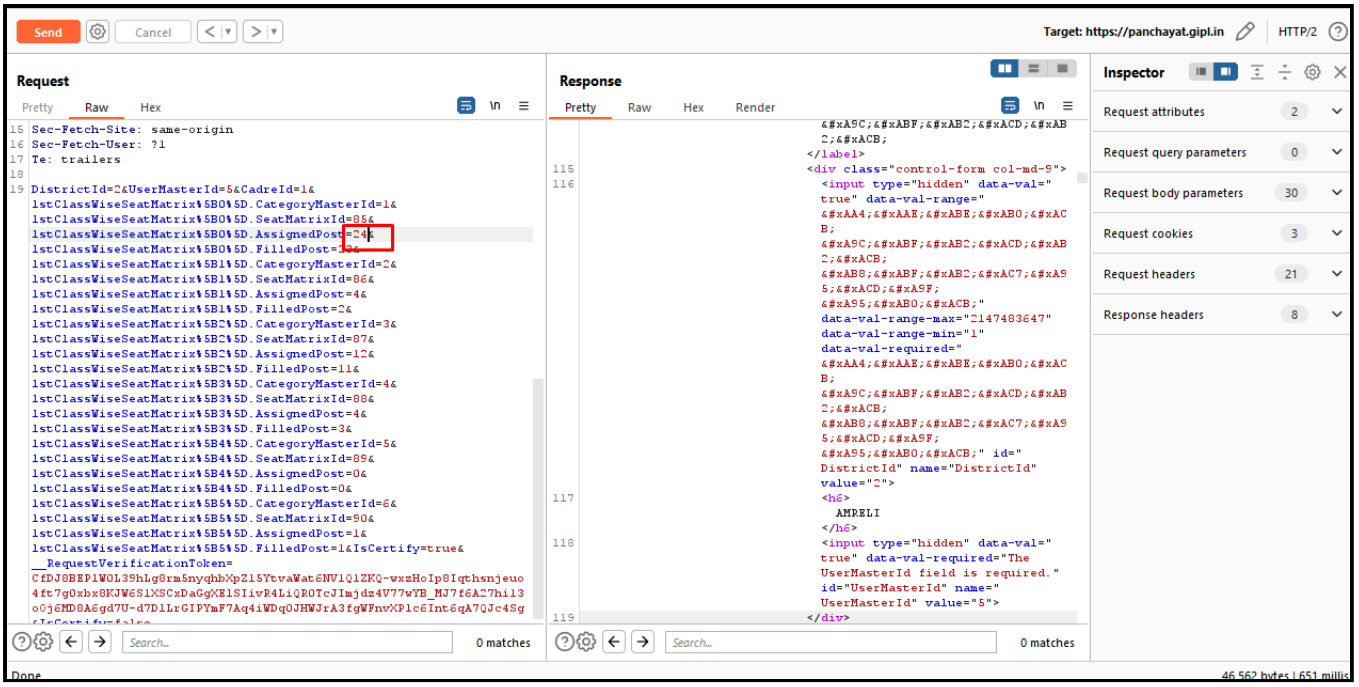


Figure 9 - Now in the repeater, change the value from 2 to 24. Click on Send.

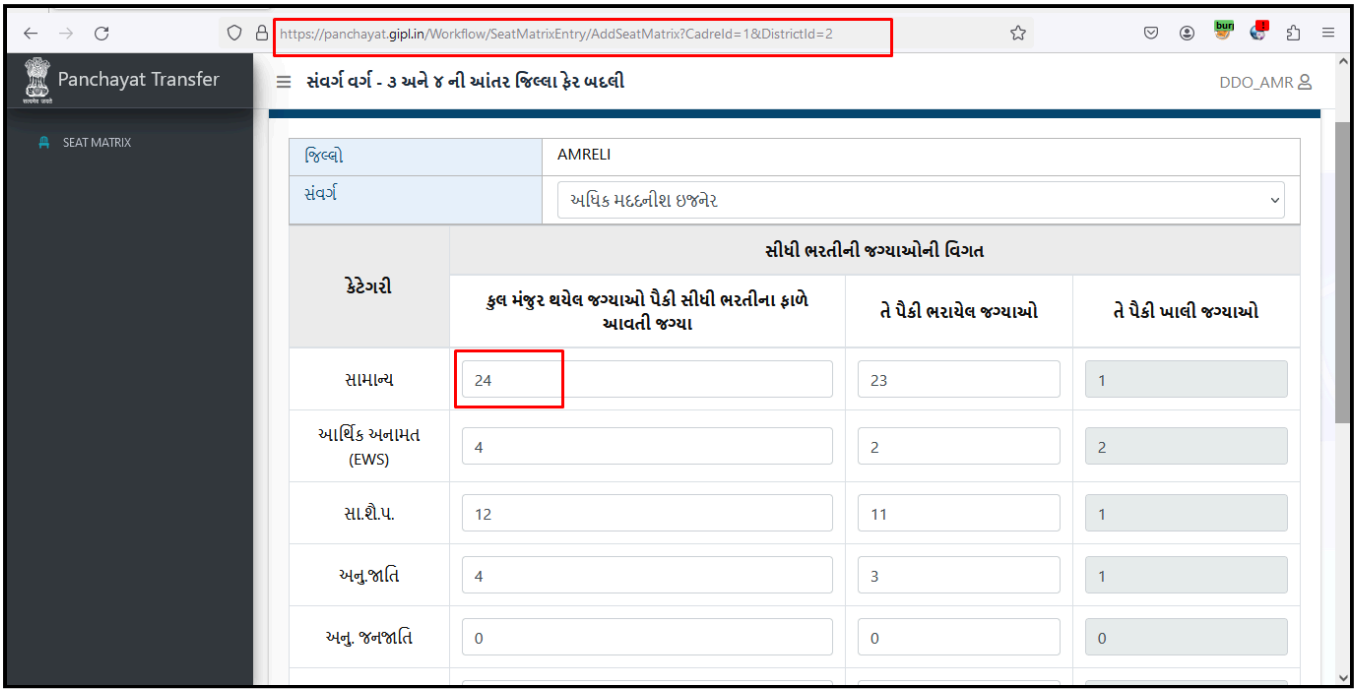


Figure 10 - Login again the account and you will see that the value has been change to 24.

Observation ID & Title	8.4 Improper Cache Control		
CVSS Risk Rating	Medium	Status	Open

Observation Details

While testing web application, it came to our notice that, this application is vulnerable to Improper cache control. Cache-control is an HTTP header that dictates browser caching behavior. when someone visits a website, their browser will save certain resources, such as images and website data, in a store called the cache. When that user revisits the same website, cache-control sets the rules which determine whether that user will have those resources loaded from their local cache or not.

Risk

Using this vulnerabilities, an attacker can view sensitive information even if he is not logged into the account

Recommendations

The Following are the recommendations that should be implemented.

The web server should return the following HTTP headers in all responses containing sensitive content:

Cache-control: no-store
Pragma: no-cache

Affected URL & Parameter

Throughout the Application

CVSS Vector

5.3 CVSS:3.0/AV:L/AC:L/PR:N/UI:N/S:C/C:L/I:H/A:L

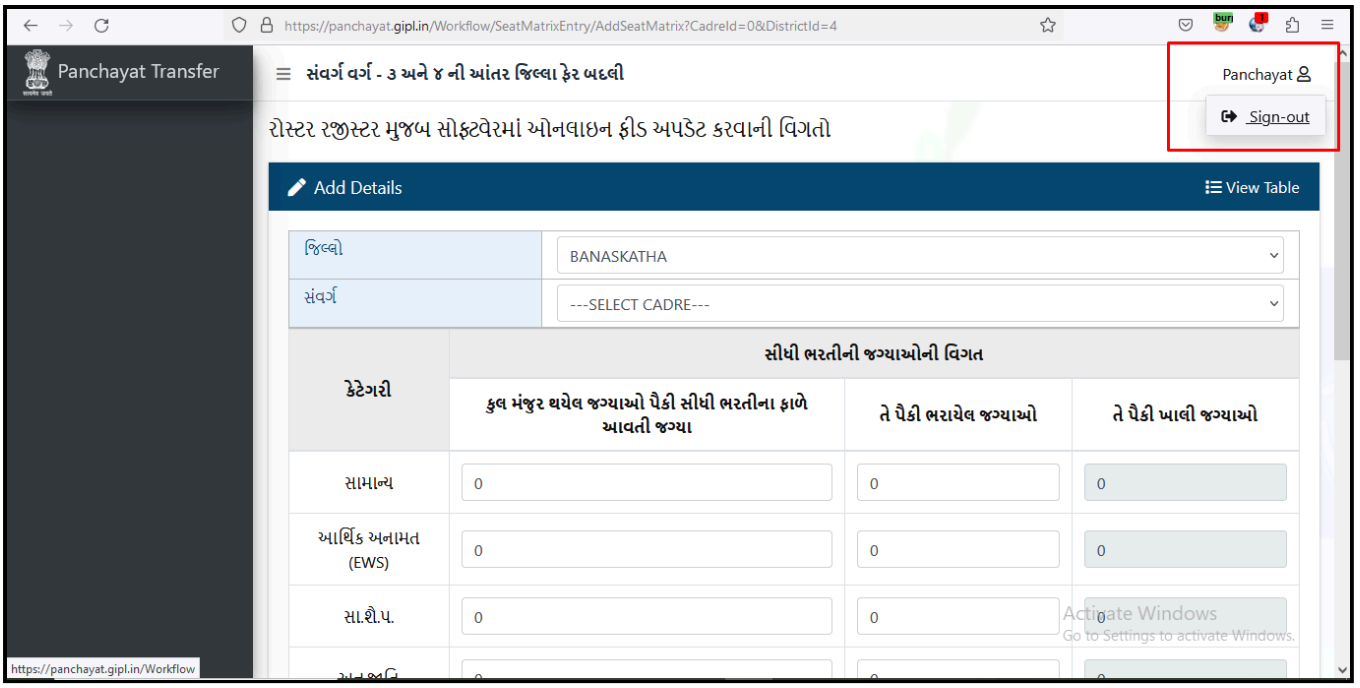


Figure 11 - Login in the account and access any page.

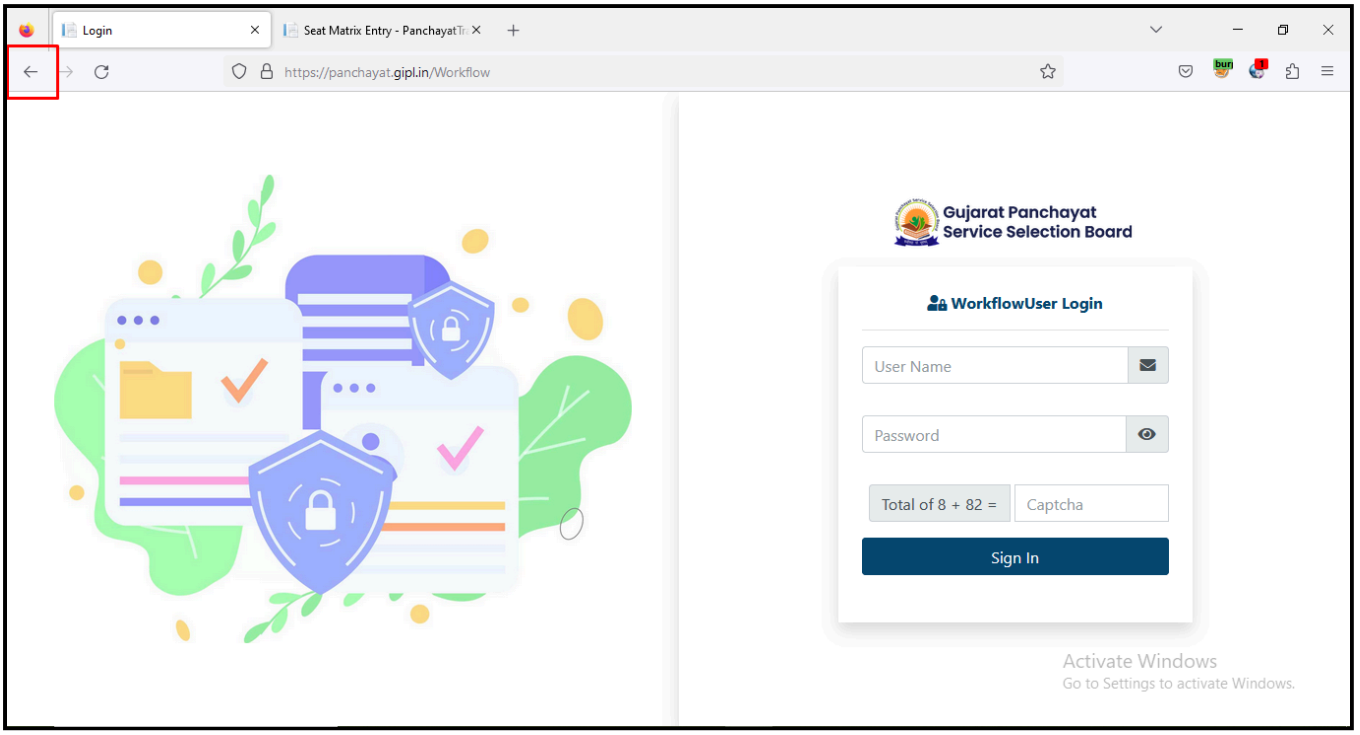


Figure 12 - Logout the account and click on back button.

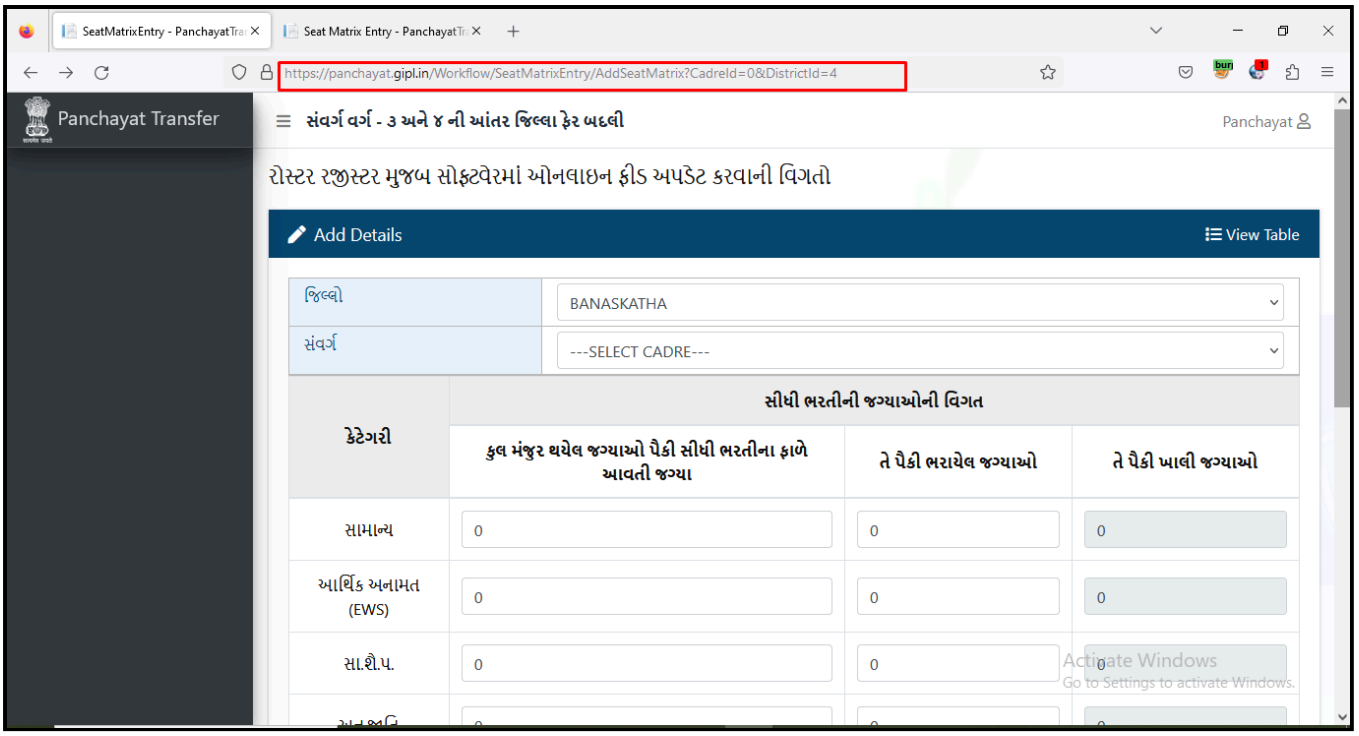


Figure 13 - As you can see that the application is returned to the previous page without any authentication.

Observation ID & Title	8.5 Improper Error handling		
CVSS Risk Rating	Medium	Status	Open

Observation Details

Improper error handling refers to the failure to properly manage and handle errors that occur during the execution of a program. This can include not properly logging or reporting errors, not providing adequate feedback to the user, or not properly securing error messages.

Risk

If errors are not properly handled, they can potentially be exploited by attackers to gain unauthorized access to sensitive information or to perform other malicious actions.

Some potential impacts of improper error handling include:

Unauthorized access to sensitive information: If an attacker is able to view or exploit error messages, they may be able to gain access to sensitive information, such as passwords or financial data, that is stored in the program's database.

Reputation damage: If an attacker is able to successfully exploit improper error handling, it could damage the reputation of the organization that operates the program, potentially leading to loss of trust from customers and other stakeholders.

Legal and regulatory consequences: Depending on the nature and severity of the attack, the organization may be subject to penalties or fines from regulatory authorities for failing to properly handle errors in their program.

Recommendations

It is recommended that it is important to properly handle errors that occur during its execution. Some recommended steps for implementing effective error handling include:

Properly log and report errors: It is important to log and report errors in a way that allows developers to quickly identify and fix the problem. This can include providing detailed information about the error, such as the location and cause of the error, as well as any relevant context.

Provide adequate feedback to the user: When an error occurs, it is important to provide clear and concise feedback to the user about what has happened and what steps they can take to resolve the issue. This can help prevent confusion and frustration on the part of the user.

Properly secure error messages: It is important to properly secure error messages to prevent attackers from being able to view or exploit them. This can include not displaying sensitive information in error messages, and properly configuring error handling to prevent stack traces from being visible to unauthorized users.

Regularly review and test your program: It is important to regularly review and test your program to identify and prevent errors, as well as vulnerabilities that may be related to error handling. This can include conducting regular code reviews and performing penetration testing to identify potential weaknesses.

Affected URL & Parameter

Throughout the Application

CVSS Vector

5.7 CVSS:3.0/AV:L/AC:L/PR:N/UI:N/S:C/C:L/I:H/A:L

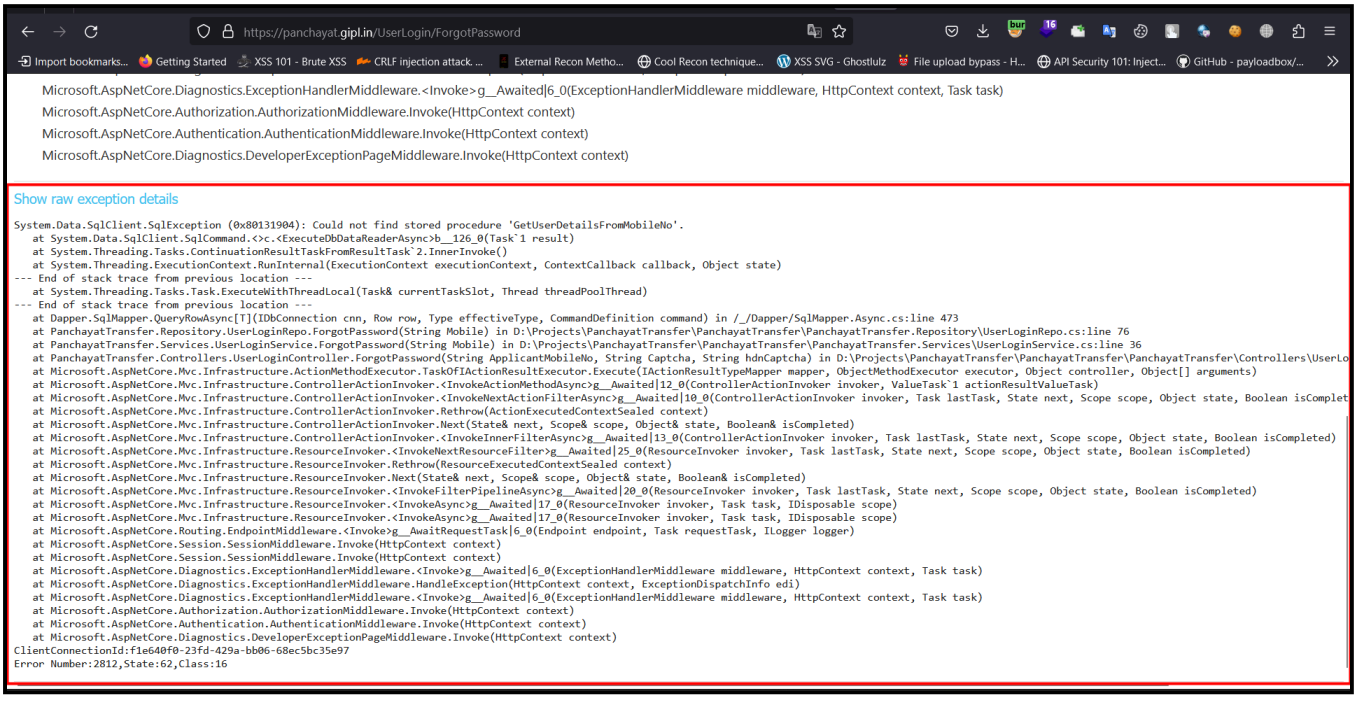


Figure 14 - As you can see that after submitting the mobile number in forgot password, application is giving code errors.

Observation ID & Title	8.6 Vulnerable JavaScript Library		
CVSS Risk Rating	Medium	Status	Open

Observation Details

jQuery, Angular, Vue, and React are a few well-known Javascript libraries. An unpatched JavaScript library can leave a website significantly open to numerous types of attacks. A number of DOM-based vulnerabilities, such as DOM-XSS, can be drawn by third-party JavaScript modules and used to steal user accounts. The given web application was found to be using a vulnerable JavaScript library as well, which immediately should be remediated.

Risk

JavaScript library's security vulnerabilities can be exploited to perform cross-site scripting, cross-site request forgery, and buffer overflow attack.

Recommendations

It is highly recommended that:-

As part of patch management, implement version management for JavaScript libraries.

Remove libraries that are no longer in use to reduce your attack surface.

Frequently check for patches and upgrade JavaScript libraries to the latest version.

Affected URL & Parameter

Throughout the Application

CVSS Vector

4.7 CVSS:3.0/AV:L/AC:L/PR:N/UI:N/S:C/C:L/I:H/A:L

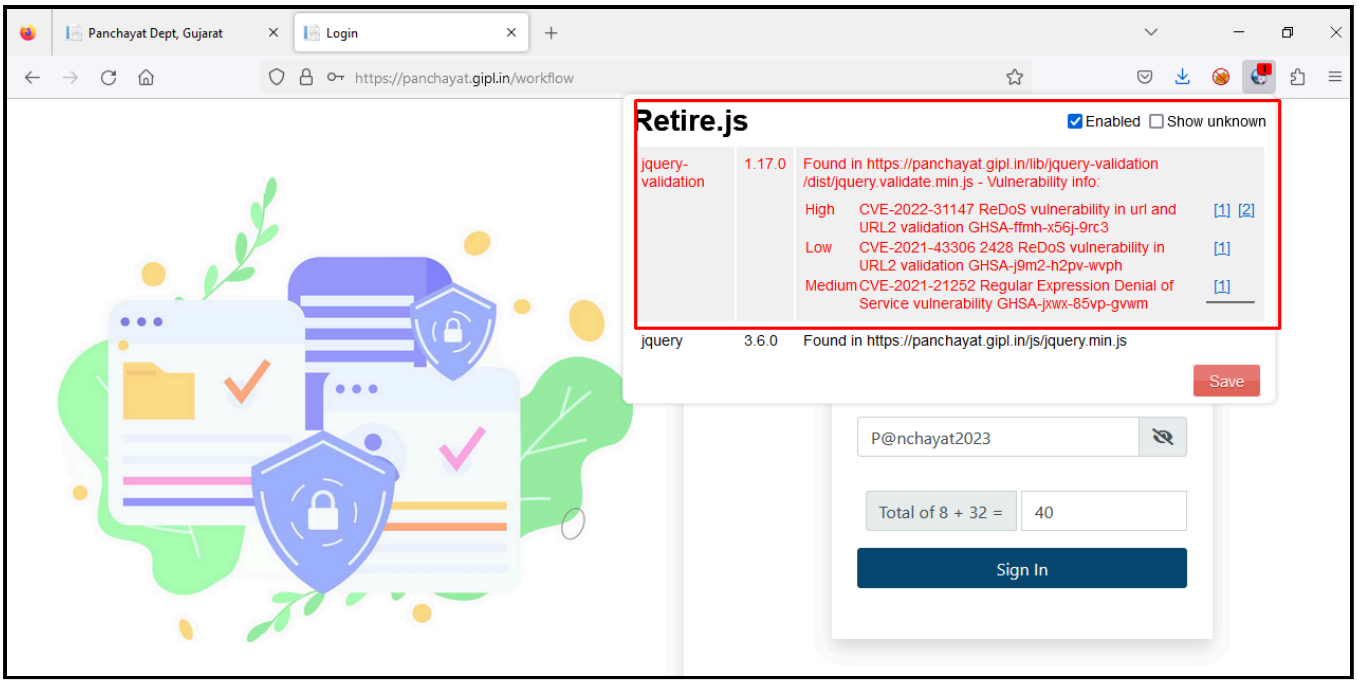


Figure 15 - As you can see that the application is vulnerable to old Jquery version.

Observation ID & Title	8.7 Missing Security Header		
CVSS Risk Rating	Medium	Status	Open

Observation Details

Upon testing the web application, we found that there were important security headers missing from the web application. Whenever a browser requests a page from any web server, the server responds with the content along with HTTP response headers. These HTTP security headers tell the browser how to behave while handling the website content.

Risk

If security headers are missing in a web application or have not been implemented properly, attackers can conduct various attacks such as cross site scripting, Spectre, data injection, HTTP redirection, downgrade attacks, etc.

Recommendations

The Following are the recommendations that should be implemented.

Apply Webserver Configuration (Apache, Nginx, and HSTS).

Apply X-Frame-Options.

Apply X-XSS-Protection.

Apply X-Content-Type-Options

Apply HTTP Strict-Transport-Security

Apply Same-Site Cookie

Apply Content-Security-Policy

Apply Referrer-Policy

Apply Cache-Control

Apply Access-Control-Allow-Origin

Affected URL & Parameter

Throughout the Application

CVSS Vector

4.5 CVSS:3.0/AV:L/AC:L/PR:N/UI:N/S:C/C:L/I:H/A:L

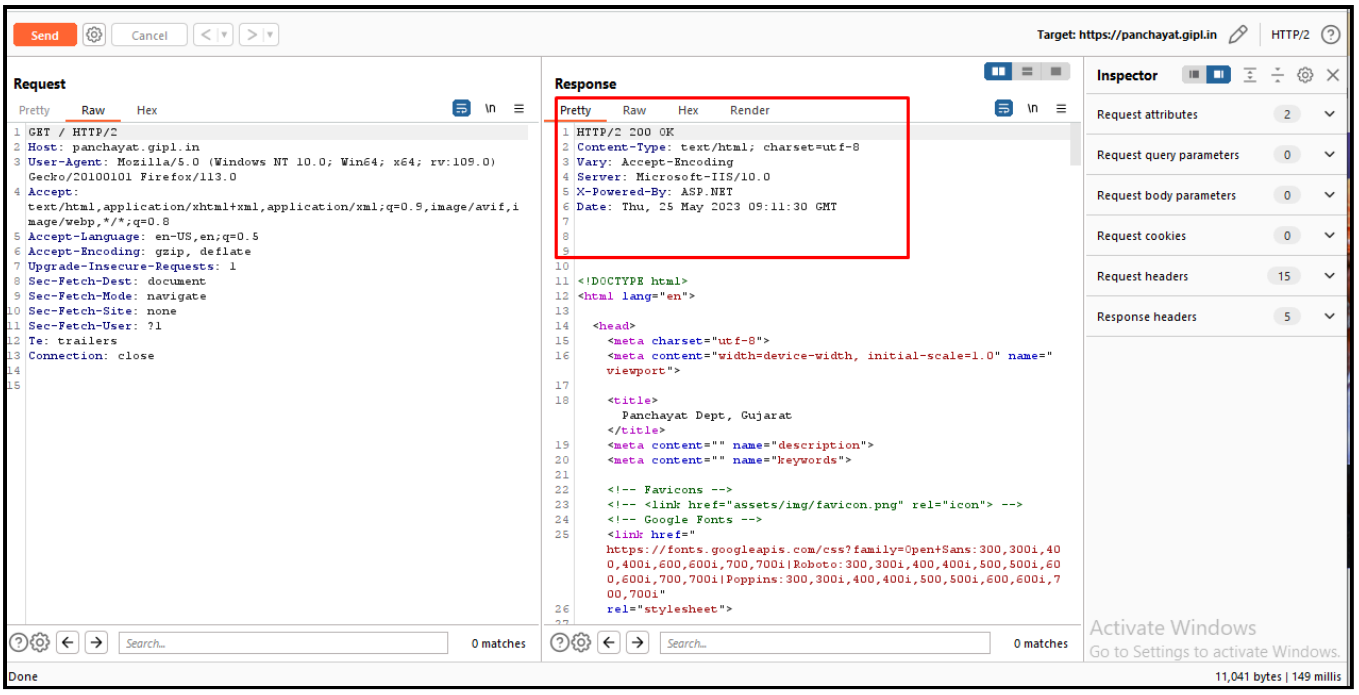


Figure 16 - As you can see that the headers are missing in the application.

Observation ID & Title	8.8 SQL Possible		
CVSS Risk Rating	Medium	Status	Open
Observation Details			
As per observation, while resetting the password, application is giving the sql error which is further possible to extract the data if the attacker wants.			
Risk			
Attacker can extract the data of the server files with the help of sql error.			
Recommendations			
The Following are the recommendations that should be implemented. Application should not give such SQL error which can result to the high impact to the organization.			
Affected URL & Parameter			
Throughout the Application			
CVSS Vector			
4.9 CVSS:3.0/AV:L/AC:L/PR:N/UI:N/S:C/C:L/I:H/A:L			

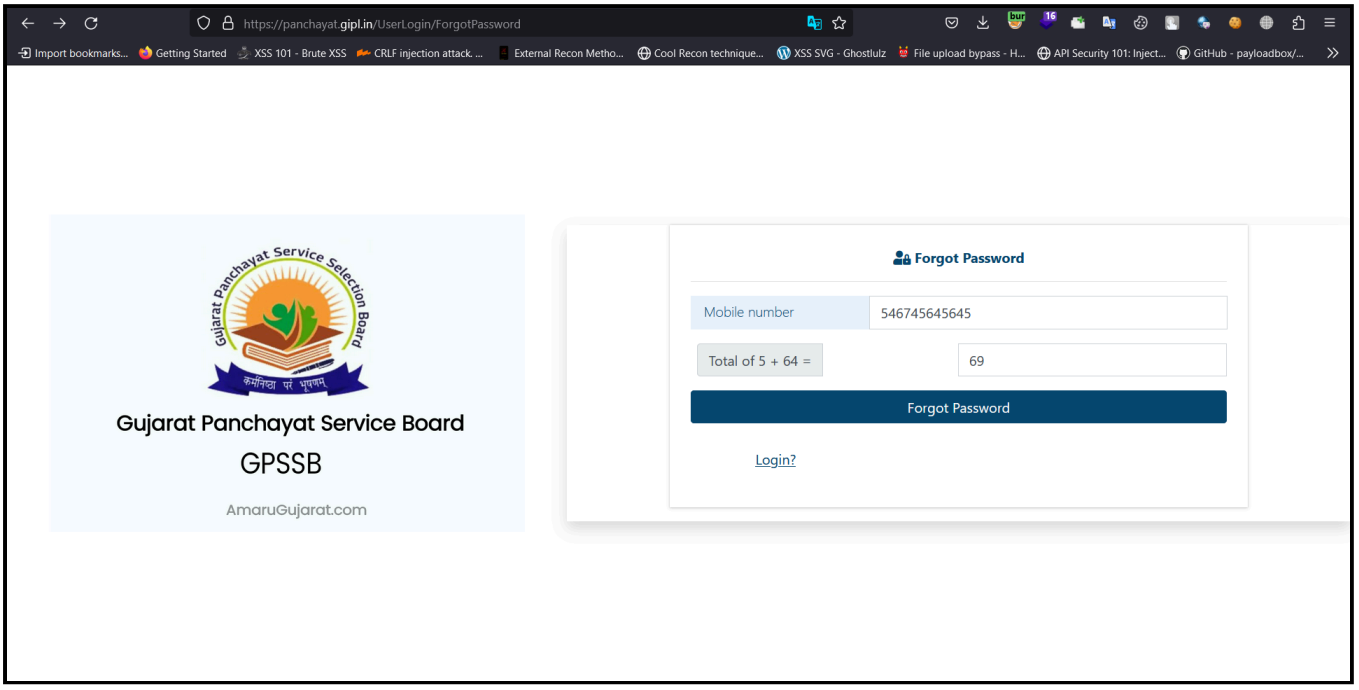


Figure 17 - Fill up the details and click on submit.

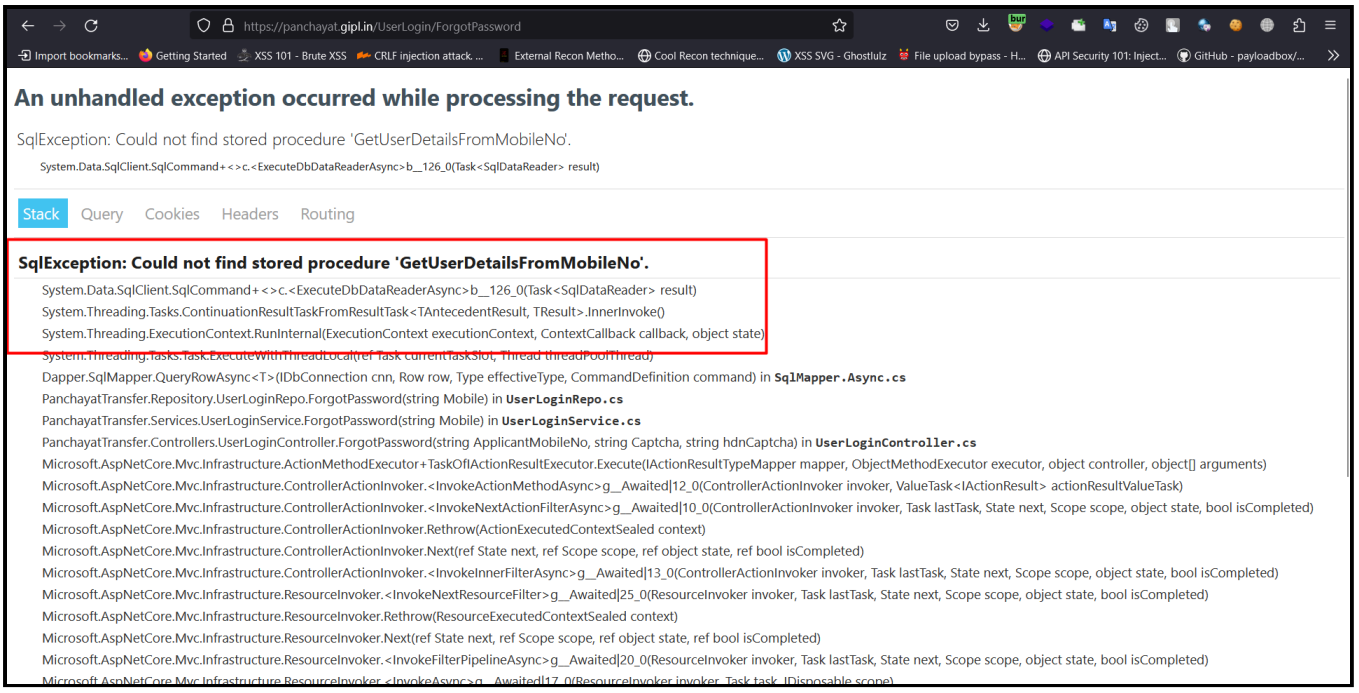


Figure 18 - As you can observe that the application is giving the SQL errors.

Observation ID & Title	8.9 TLS Fallback SCSV Support		
CVSS Risk Rating	Medium	Status	Open

Observation Details

The TLS Signalling Cipher Suite Value (SCSV) protects against TLS/SSL downgrade attack such as PODDLE. If enables the server, ensure that the strongest protocol that both client and server understand is used. It has been observed that the web application is not using or not enabled the TLS_FALLBACK_SCSV Support.

Risk

Attackers with the network level access can attempt downgrade attacks; where the attacker forces a client to negotiate to a weaker or known vulnerable encryption scheme. This encrypted session can then be later broken or decrypted by the malicious user.

Recommendations

The Following are the recommendations that should be implemented.

Ensure that all web servers and client devices support the TLS_FALLBACK_SCSV cipher suite. This will prevent attackers from being able to perform a downgrade attack and intercept sensitive information.

Regularly update the web servers and client devices to the latest versions of the TLS protocol. This will ensure that they are able to support the latest security features and protections against attacks.

OpenSSL 1.0.1 users should upgrade to 1.0.1j
OpenSSL 1.0.0 users should upgrade to 1.0.0o
OpenSSL 0.9.8 users should upgrade to 0.9.8zc

Affected URL & Parameter

Throughout the Application

CVSS Vector

4.7 CVSS:3.0/AV:L/AC:L/PR:N/UI:N/S:C/C:L/I:H/A:L

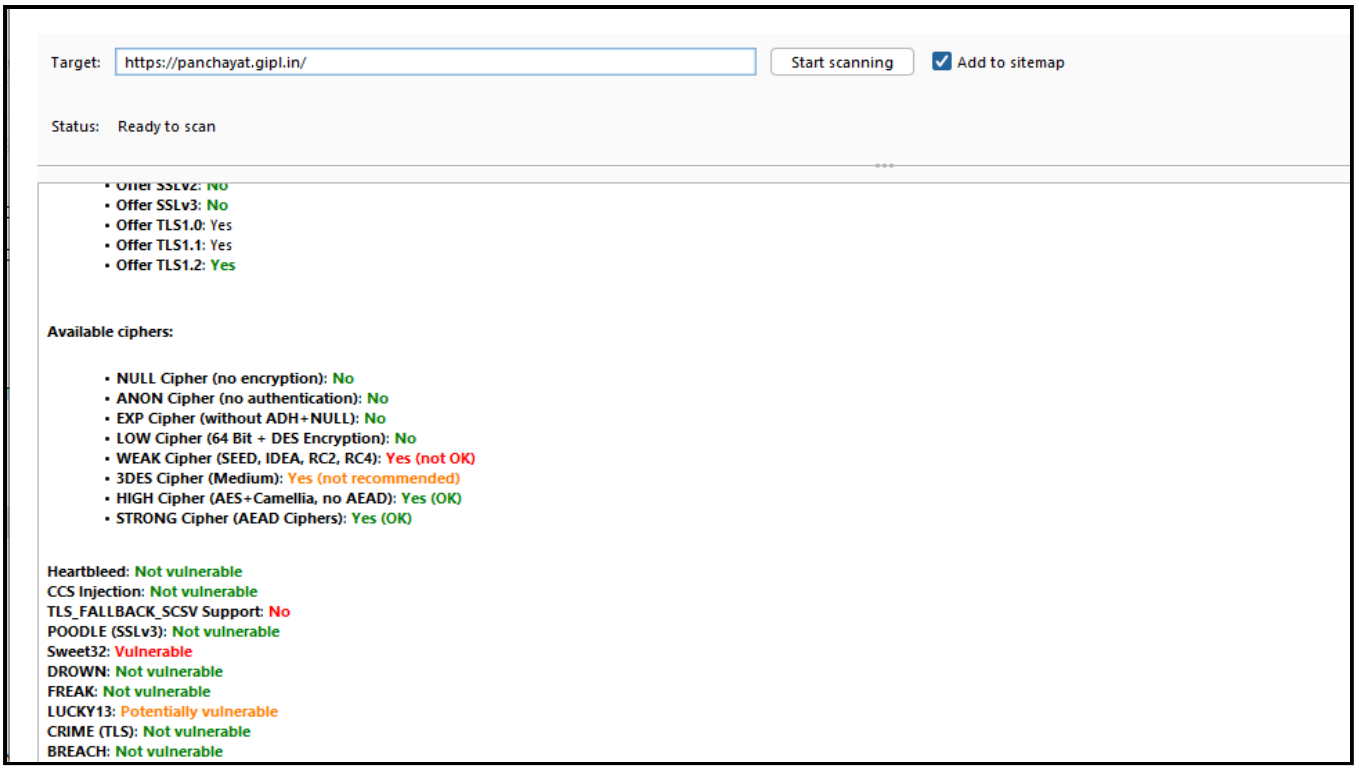


Figure 19 - As you can see that the TLS fallback is not supported.

Observation ID & Title	8.10 Banner Grabbing		
CVSS Risk Rating	Low	Status	Open

Observation Details

Web server fingerprinting is a critical task for the penetration tester. Knowing the version and type of a running web server allows testers to determine known vulnerabilities and the appropriate exploits to use during testing.
There are several different vendors and versions of web servers on the market today. Knowing the type of web server that is being tested significantly helps in the testing process and can also change the course of the test. This information can be derived by sending the web server specific commands and analysing the output, as each version of web server software may respond differently to these commands. By knowing how each type of web server responds to specific commands and keeping this information in a web server fingerprint database, a penetration tester can send these commands to the web server, analyse the response, and compare it to the database of known signatures.

Risk

Banner grabbing does not involve the leakage of critical data but rather information that may aid the attacker during the exploitation phase of the attack. For example, if the target leaks the version of PHP running on the server and that version happens to be vulnerable to remote command/code execution (RCE) because it wasn't updated, attackers may exploit the known vulnerability and take full control of the web application.

Recommendations

The Following are the recommendations that should be implemented.
While exposed server information is not necessarily in itself a vulnerability, it is information that can assist attackers in exploiting other vulnerabilities that may exist. Exposed server information can also lead attackers to find version-specific server vulnerabilities that can be used to exploit unpatched servers. For this reason, it is recommended that some precautions be taken. These actions include:
• Obscuring web server information in headers, such as with Apache's mod headers module.
• Using a hardened reverse proxy server to create an additional layer of security between the web server and the Internet.
• Ensuring that web servers are kept up to date with the latest software and security patches.

Affected URL & Parameter

Throughout the Application

CVSS Vector

2.7 CVSS:3.0/AV:L/AC:L/PR:N/UI:N/S:C/C:L/I:H/A:L

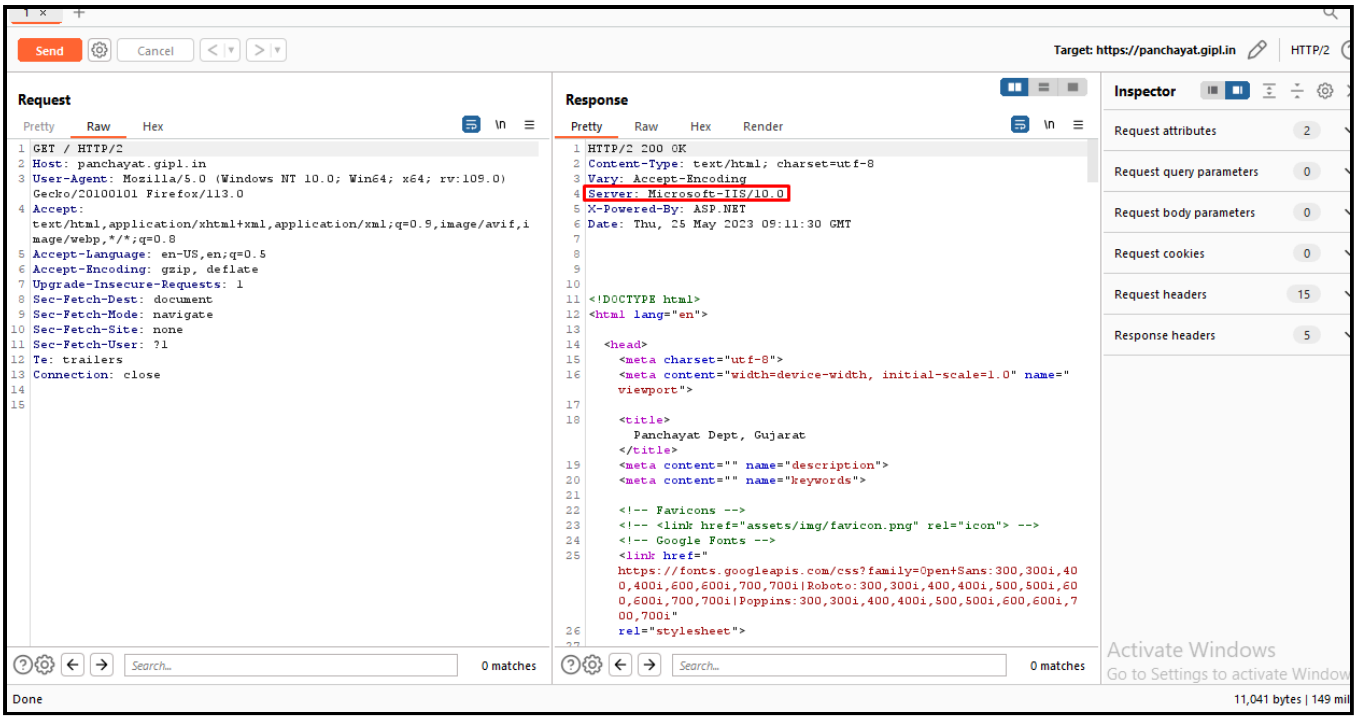


Figure 20 - As you can see that the application version is clearly visible.

Observation ID & Title	8.11 No Session Timeout		
CVSS Risk Rating	Low	Status	Open

Observation Details

A session timeout vulnerability is a security flaw that arises when a web application fails to properly manage the session expiration of a user's authenticated session. This can result in a user's session remaining active even after they have left the site, potentially allowing an attacker to gain unauthorized access to sensitive information.

Risk

Unauthorized Access: If an attacker is able to hijack an active session, they can gain unauthorized access to sensitive information such as user credentials, personal data, and financial information.
Data Theft: A session timeout vulnerability can allow an attacker to steal data from a user's session, potentially leading to identity theft, financial fraud, and other forms of cybercrime.
Data Manipulation: An attacker may be able to manipulate data within a user's active session, potentially leading to the insertion or alteration of sensitive information.

Recommendations

To avoid session timeout vulnerabilities, web applications should implement appropriate session management
techniques such as:
• Setting a reasonable session timeout duration, based on the sensitivity of the data being accessed and the likelihood of the user remaining idle for an extended period of time.
• Implementing an automatic logout feature that terminates a user's session after a set period of inactivity.
• Ensuring that session IDs are securely generated and transmitted using secure channels.
• Using secure session storage mechanisms, such as HTTP-only cookies or server-side storage.

Affected URL & Parameter

Throughout the Application

CVSS Vector

3.0 CVSS:3.0/AV:L/AC:L/PR:N/UI:N/S:C/C:L/I:H/A:L

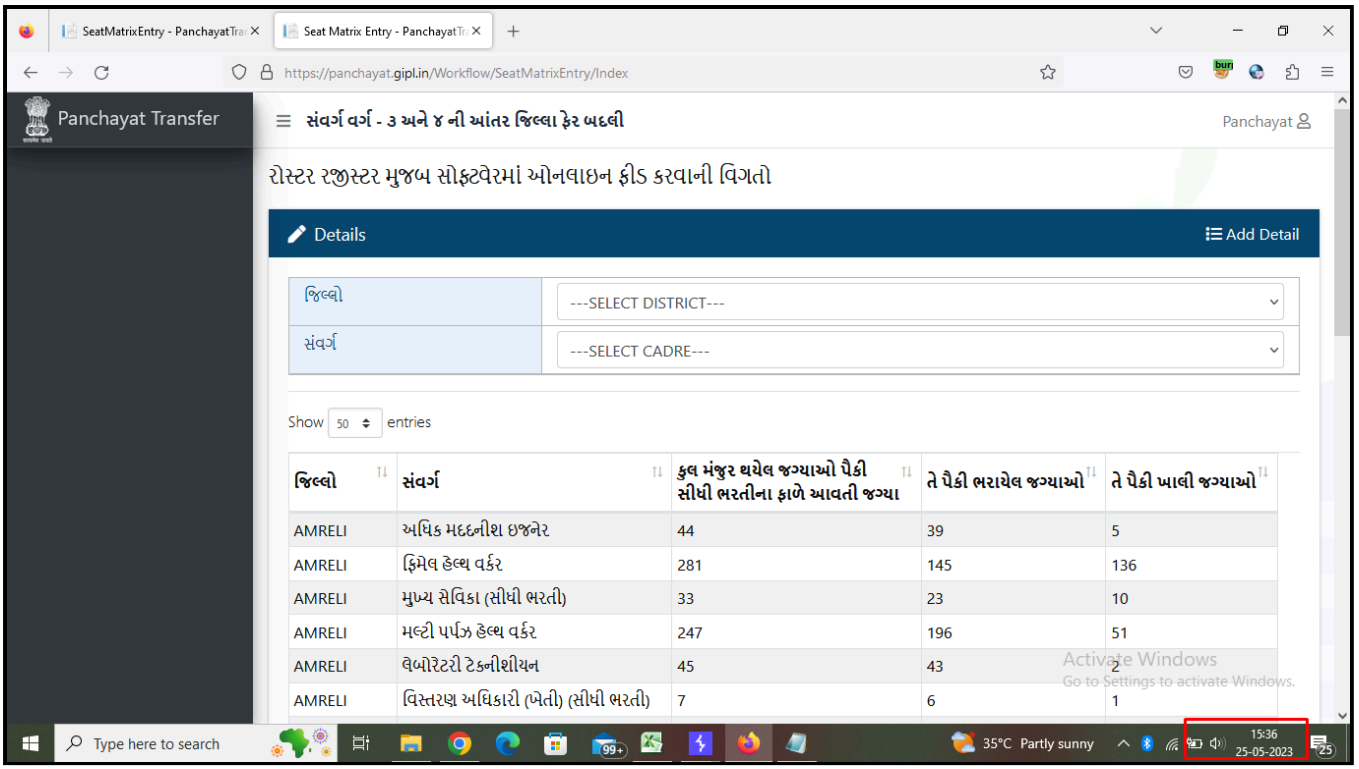


Figure 21 - As you can see that the application is not terminating the session even after 30 min of idle time.

Observation ID & Title	8.12 Dangerous Method Enabled		
CVSS Risk Rating	Low	Status	Open

Observation Details

While testing the application it came to our notice that the application is vulnerable is allowing dangerous methods like PUT, HEAD, DELETE these method allow an attacker to modify the files stored on the web server and, in some scenarios, steal the credentials of legitimate users

Risk

Web server accepting these methods may allow an attacker to gain full control over the application and its environment. The same methods can be also be used to cause Denial of Service (DoS) by destroying the application structure . An unauthenticated attacker can gain access to complete information held in databases that is not meant to be accessed by them. The attacker could also gain control over the entire database server which can leads to confidentially, integrity and availability.

Recommendations

The Following are the recommendations that should be implemented.

1) Always enable deny all option.
2) Configure your web and application server to disallow HEAD requests entirely

Affected URL & Parameter

Throughout the Application

CVSS Vector

3.3 CVSS:3.0/AV:L/AC:L/PR:N/UI:N/S:C/C:L/I:H/A:L

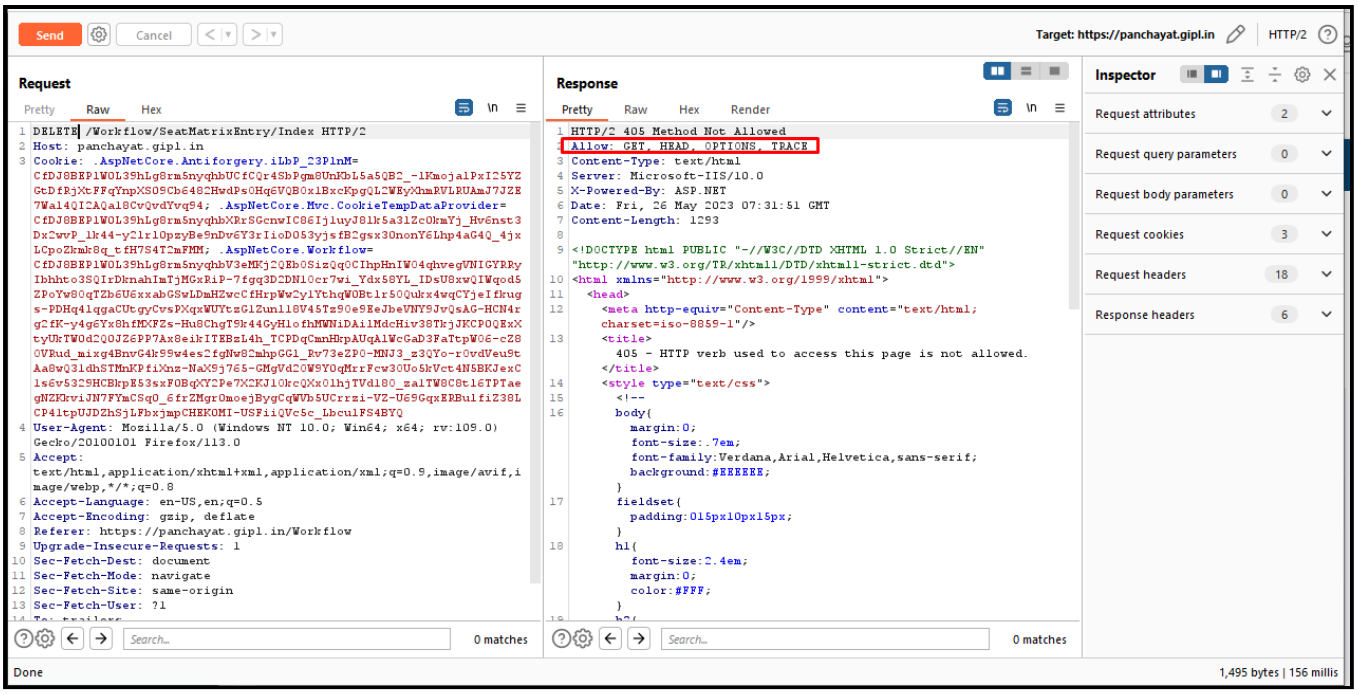


Figure 22 - As you can see that HEAD, TRACE method is allowed.

Observation ID & Title	8.13 HTTP Strict Transport Security		
CVSS Risk Rating	Low	Status	Open

Observation Details

During the testing phase, it came to our notice that the web application does not have HTTP Strict Transport Security mechanism implemented. The HTTP Strict Transport Security (HSTS) is a policy mechanism that is implemented through the use of a special response header. HSTS basically prevents any communications from being sent over HTTP and will only allow communications over HTTPS (encrypted channel). This mechanism will automatically redirect HTTP requests over to HTTPS for the target domain. It also prevents HTTPS click through prompts on browsers.

Risk

The attacker can conduct various attacks if the HSTS policy mechanism is not implemented. These attacks include:

Protocol Downgrade attacks: The Protocol downgrade attack is an attack through which, an attacker can downgrade the used protocol in the web application to communicate using any vulnerable protocol. Using the vulnerable protocol, the attacker can steal data sent between the server and the client.

Man-In-The-Middle attacks: The man-in-the-middle (MITM) attack is an attack through which an attacker can sniff data passed via the communication channel. The attacker can access sensitive information from login credentials to application's cookies.

Cookie-hijacking: The cookie hijacking attack is an attack through which, an attacker can hijack information present in the user's cookie from the communication channel. The cookie details include information about the user's session. The attacker can use this information to steal user's sessions.

Recommendations

The Following are the recommendations that should be implemented.

Configure the remote web server to communicate using HSTS.

If there is any preload directive in the application, it is recommended to switch back to HTTP. An attacker can send a preload directive from the application. These preload directives might have serious issues on the server. The preload directive can be used to prevent the users from accessing the web application along with any of its subdomains.

The web application must instruct the user's web browser to only access the application using HTTPS. To do this, the application must enable HTTP Strict Transport Security (HSTS). The HSTS can be enabled by adding the response header 'Strict-Transport-Security'. Set the value 'max-age=expireTime'. It is also recommended to add the 'includeSubDomains' flag.

Affected URL & Parameter

Throughout the Application
CVSS Vector
3.0 CVSS:3.0/AV:L/AC:L/PR:N/UI:N/S:C/C:L/I:H/A:L

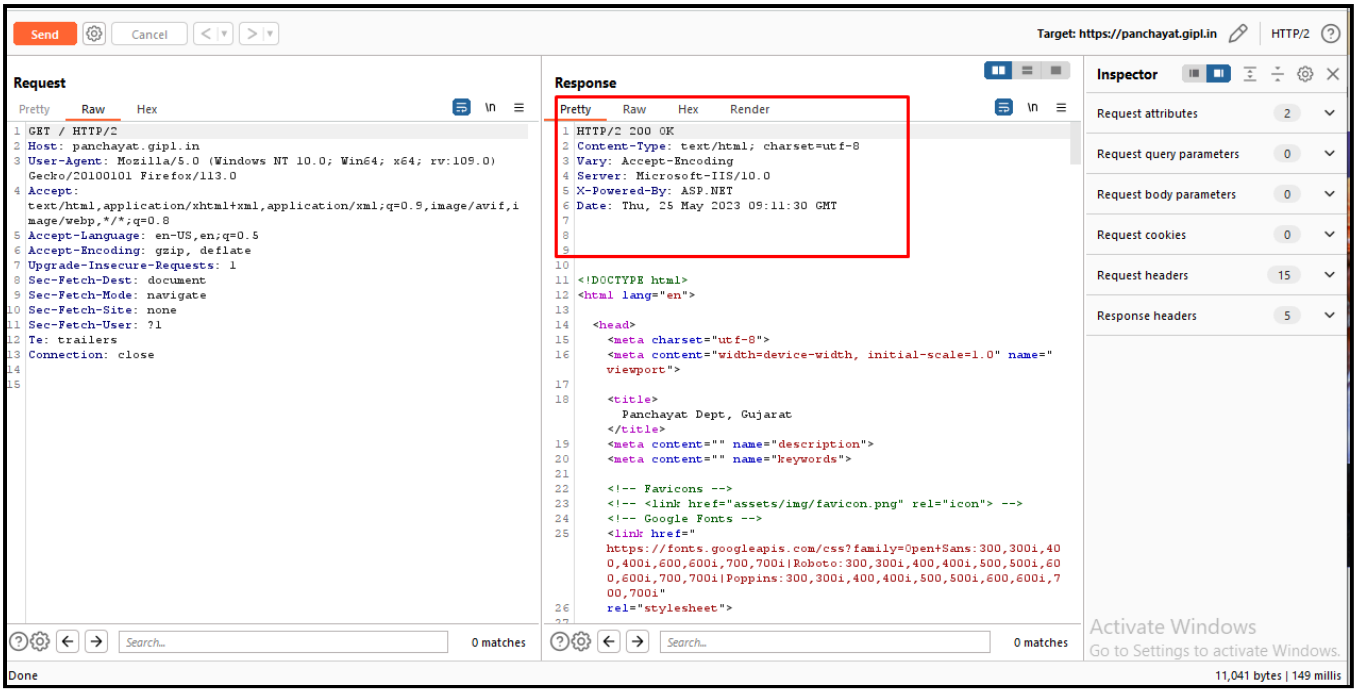


Figure 23 - As you can see that HTTP Strict Transport Policy is not enforced.

Observation ID & Title	8.14 Weak lock out mechanism		
CVSS Risk Rating	Low	Status	Open

Observation Details

A lock-out mechanism is a security feature that is designed to prevent brute-force attacks by temporarily locking an account after a certain number of failed login attempts. However, if the lock-out mechanism is weak, it may not be effective at preventing brute-force attacks, allowing an attacker to continue trying to guess a user's password until they are successful.

Risk

The impact of this vulnerability can be significant, as it can allow an attacker to gain unauthorized access to a user's account. This can be used to steal sensitive information, perform actions on behalf of the user, and potentially compromise the security of the entire system.

Recommendations

It is highly recommended to:-

Implement a strong lock-out mechanism that is effective at preventing brute-force attacks.

Use longer lock-out periods, requiring additional authentication factors.

Implement CAPTCHA challenges to verify that the user is human.

Affected URL & Parameter

Throughout the Application

CVSS Vector

3.9 CVSS:3.0/AV:L/AC:L/PR:N/UI:N/S:C/C:L/I:H/A:L

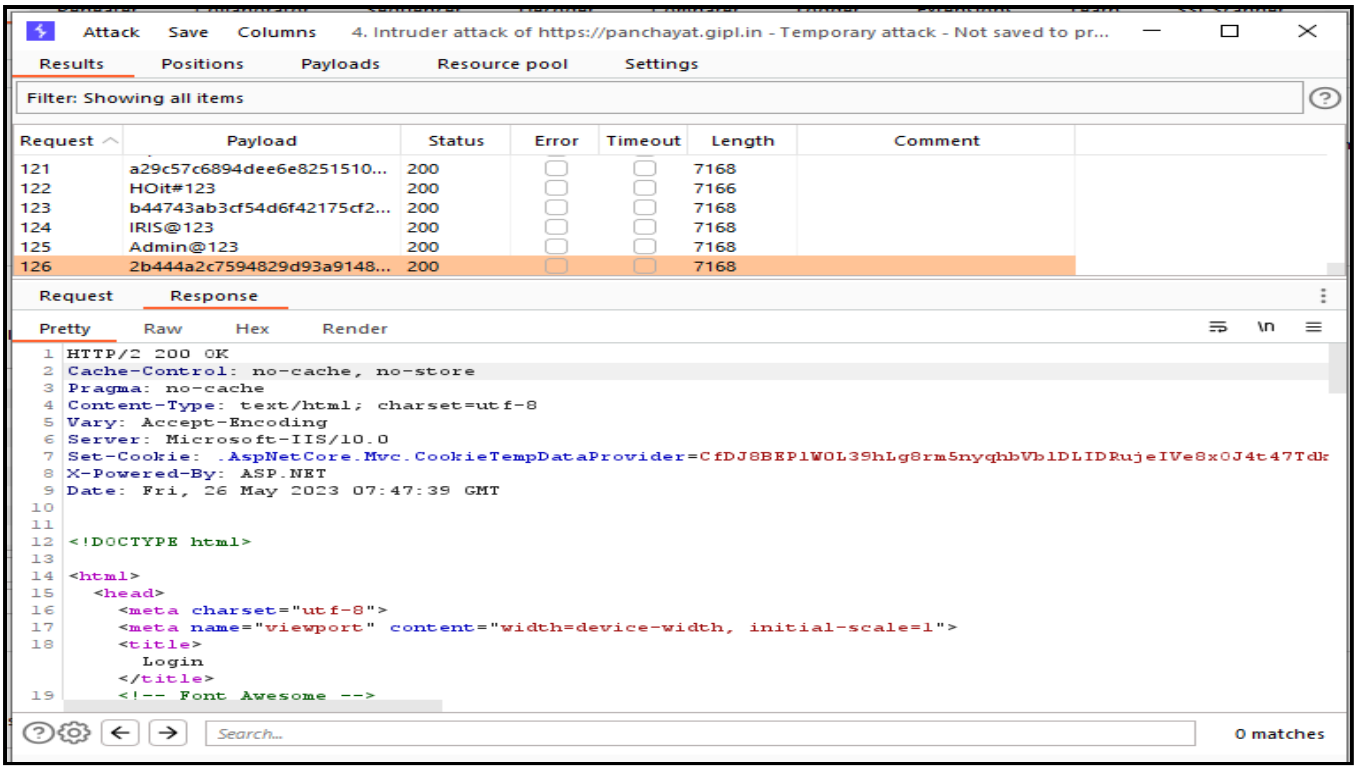


Figure 24 - As you can see that after attempting so many passwords, account is not getting lockout.

Observation ID & Title	8.15 AutoComplete Enabled		
CVSS Risk Rating	Low	Status	Open

Observation Details

AutoComplete is basically a feature which is provided by many web browsers / web applications. This eases the tiring process of typing the credentials again and again. Although seen as a feature, having AutoComplete enabled in a web browser can pose a potential security vulnerability. This is because AutoComplete automatically fills in login information (such as a username and password) on websites that the user has previously visited. This can make it easier for attackers to gain access to a user's account if they have physical access to the user's device.

Risk

If an attacker is able to gain access to the user's device, they could potentially open the web browser and navigate to a website where the user has previously logged in. If AutoComplete is enabled, the login information will be automatically filled in, and the attacker could potentially use this information to gain access to the user's account.

Recommendations

To mitigate the potential security vulnerability, it is advised to:-

Disable AutoComplete in the web browser.

Use a password manager.

Be careful about where you leave your device.

Affected URL & Parameter

Throughout the Application

CVSS Vector

3.0 CVSS:3.0/AV:L/AC:L/PR:N/UI:N/S:C/C:L/I:H/A:L

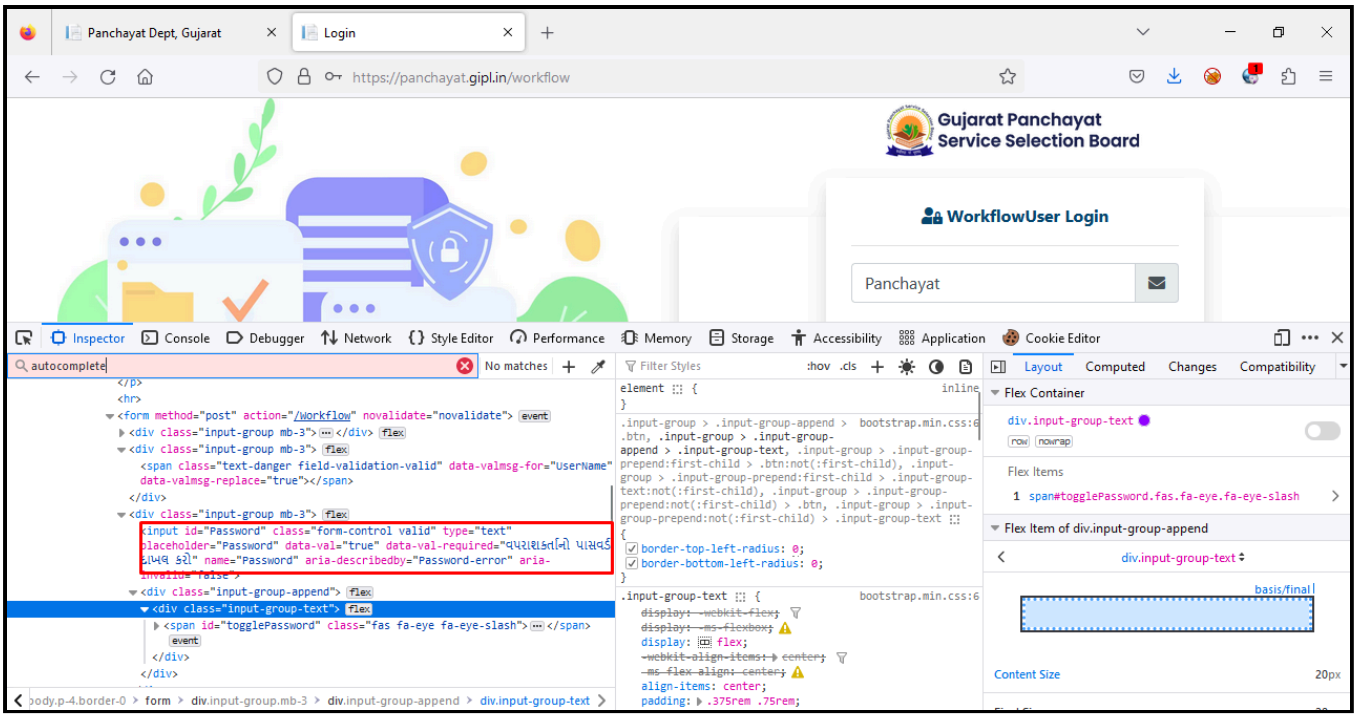


Figure 25 - As you can see that the autocomplete is not set to OFF.

Observation ID & Title	8.16 Weak password policy		
CVSS Risk Rating	Low	Status	Open

Observation Details

While testing the web application, it came to our notice that the application has weak password policies. Weak password policies vulnerability comes under the category of Broken authentication, and this happens due to insufficient mention of password hardening rules in the policy. Password hardening rules such as keeping the minimum length of password of 8 characters, using alphanumeric characters along with special characters etc. This ensures that passwords cannot be easily guessed or brute forced.

Risk

Attackers have to gain access to only a few accounts or just one admin account to compromise the whole system. Depending on the domain of the application this may allow social security fraud, or identity theft and disclose legally protected highly sensitive information.

Recommendations

The Following are the recommendations that should be implemented.

Tight password policy, not allowing weak or well-known passwords and not the usage of default admin credentials.

Ensure passwords have a minimum length of 8, consist alphanumeric, special characters and both lowercase and uppercase characters.

Affected URL & Parameter

Throughout the Application

CVSS Vector

3.6 CVSS:3.0/AV:L/AC:L/PR:N/UI:N/S:C/C:L/I:H/A:L

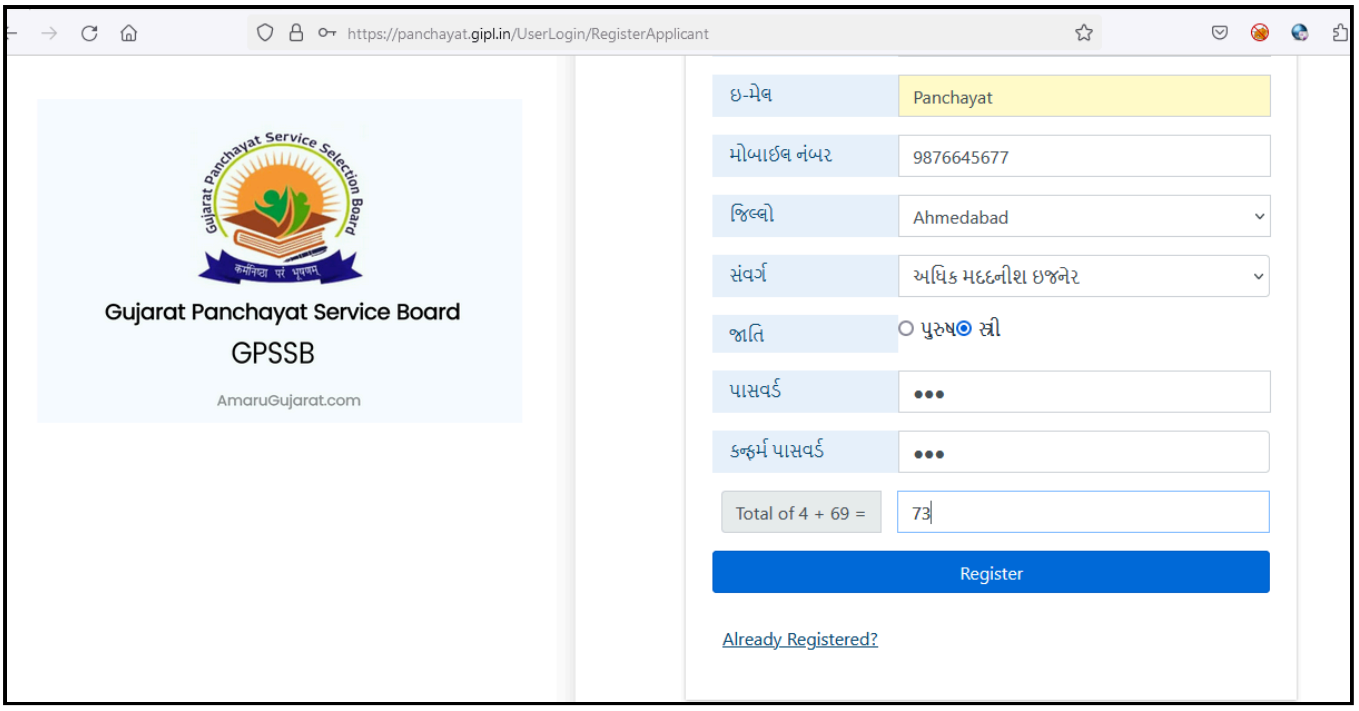


Figure 26 - As you can see that the application is accepting only 3 char of password.

Observation ID & Title	8.17 Missing secure Flag Not Set		
CVSS Risk Rating	Low	Status	Open

Observation Details

During the of the web application, it was observed that there was no secure flag set in the web application. When an HTTP protocol is used for communication between client and server, the data traffic is sent in plaintext. An HHTP allows the attacker to see/modify the traffic using a Man-In-The-Middle attack (MITM). HTTPS is a secure version of HTTP. This protocol uses SSL/TLS to protect the data in the application layer. HTTPS is used for better authentication and data integrity. A secure flag is set by the application server while sending a new cookie to the user using an HTTP Response. The secure flag is used to prevent cookies from being observed and manipulated by an unauthorized party or parties. This is because the cookie is sent as a normal text. A browser will not send a cookie with the secure flag that is sent over an unencrypted HTTP request. That is, by setting the secure flag the browser will prevent/stop the transmission of a cookie over an unencrypted channel.

Risk

Using this vulnerability, an attacker can redirect the user to a malicious site to steal information/data or even show user false data which will, in turn, affect the credibility of the website.

Recommendations

It is highly recommended to set the HTTP Secure Flag to prevent cookies from being manipulated.

Affected URL & Parameter

Throughout the Application

CVSS Vector

3.7 CVSS:3.0/AV:L/AC:L/PR:N/UI:N/S:C/C:L/I:H/A:L

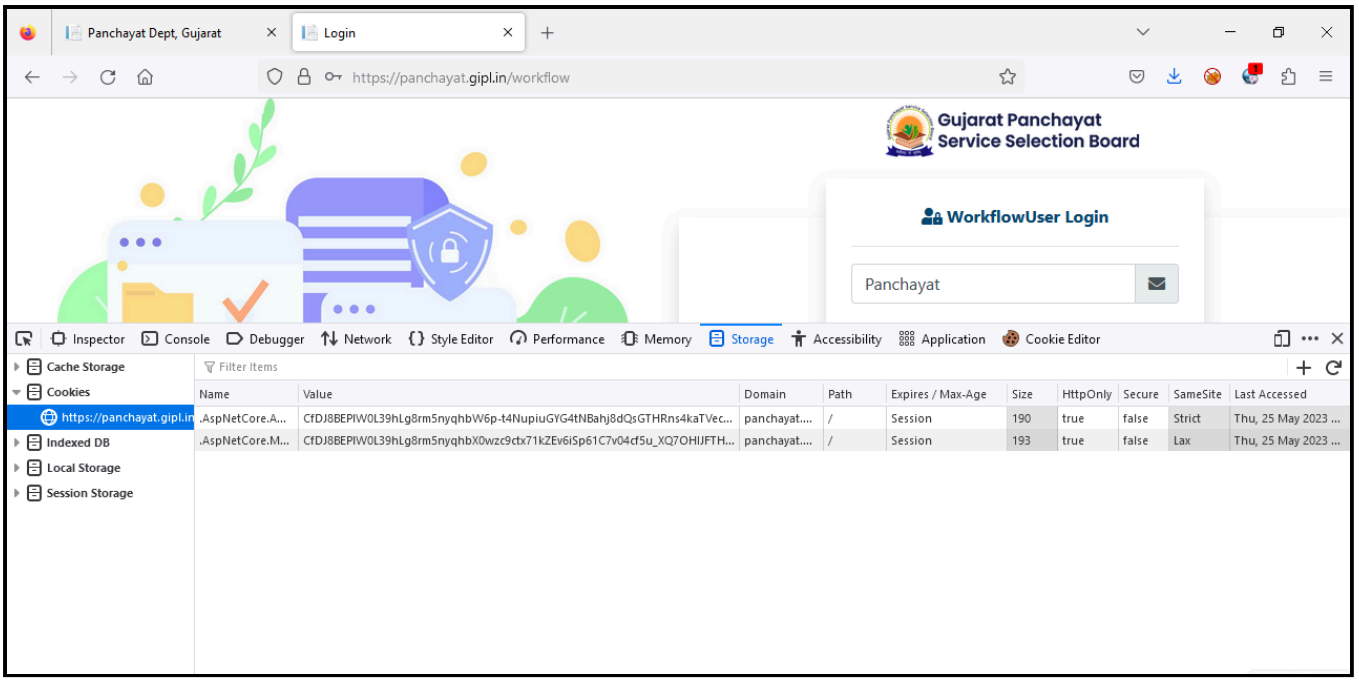


Figure 27 - As you can see that the secure flag is set to false.

Observation ID & Title	8.18 Improper Session Expiry		
CVSS Risk Rating	Low	Status	Open

Observation Details

Improper session expiry refers to the failure of a website or web application to properly manage the expiration of user sessions. This can occur if the website or application does not properly set the expiration time for a user's session, or if it fails to invalidate the user's session when it expires. The given web application was found to have improper session expiry.

Risk

If an organization is successfully targeted by an attack that exploits improper session expiry, the impact can be significant. The attacker may be able to gain unauthorized access to user accounts, steal sensitive information, or perform unauthorized actions on the user's behalf. This can result in loss of confidentiality, integrity, and availability of the organization's data, as well as damage to its reputation and financial losses.

Recommendations

To mitigate the risks associated with improper session expiry, appropriate security controls should be implemented, such as using strong encryption for data transmission and authentication mechanisms for verifying the identity of users. They should also ensure that their website or web application properly manages the expiration of user sessions, including setting appropriate expiration times and invalidating sessions when they expire.

Affected URL & Parameter

Throughout the Application

CVSS Vector

3.7 CVSS:3.0/AV:L/AC:L/PR:N/UI:N/S:C/C:L/I:H/A:L

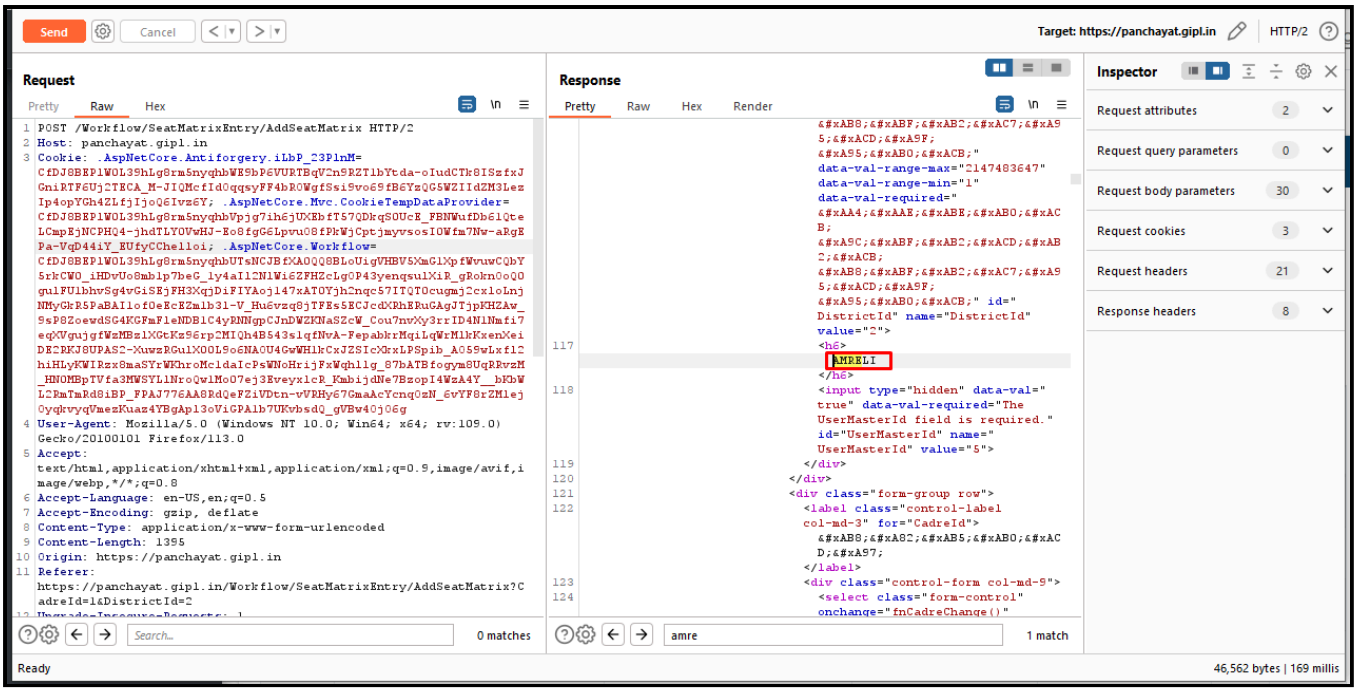


Figure 28 - Intercept the request of the page containing the information > Send the request to repeater

Figure 29 - Signout the account.

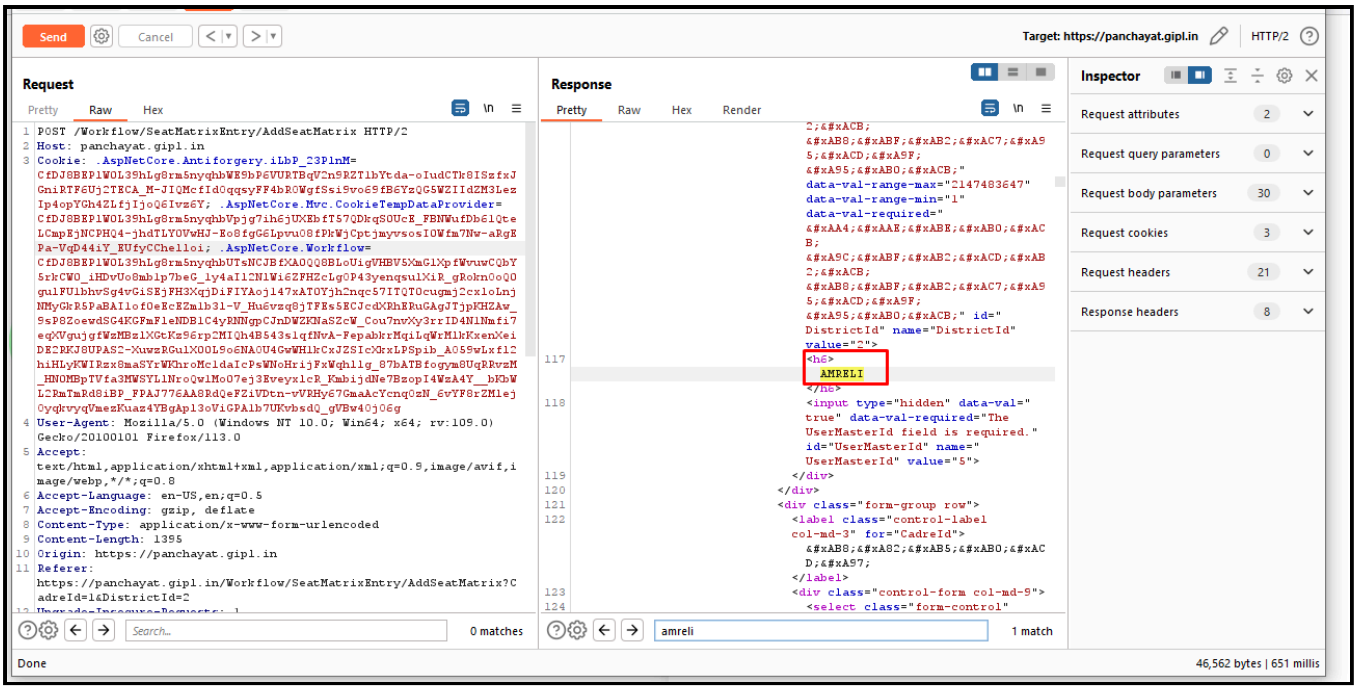


Figure 30 - Now again go to Repeater and send the request > As you can see that the application is showing the data.

Observation ID & Title	8.19 Clickjacking		
CVSS Risk Rating	Low	Status	Open

Observation Details

While testing the application it came to our notice that the application is vulnerable to clickjacking that is UI redress attack. In this attack click is hijacked that means any one can put multiple transparent or opaque layers to trick user into clicking on a button or link of another page when they were intending to click on the top level of page.

Risk

By exploiting this vulnerability, an attacker can trick the user to click on their desired page most likely owned by another application. Thus an attacker can steal sensitive information such as user credentials, credit card details etc.

Recommendations

The Following are the recommendations that should be implemented.

Implementing security policy such as Content Security Policy (CSP), or implementing the X-Frame option headers that will instruct the browser to not allow framing from other domains is recommended.

Affected URL & Parameter

Throughout the Application

CVSS Vector

3.2 CVSS:3.0/AV:L/AC:L/PR:N/UI:N/S:C/C:L/I:H/A:L

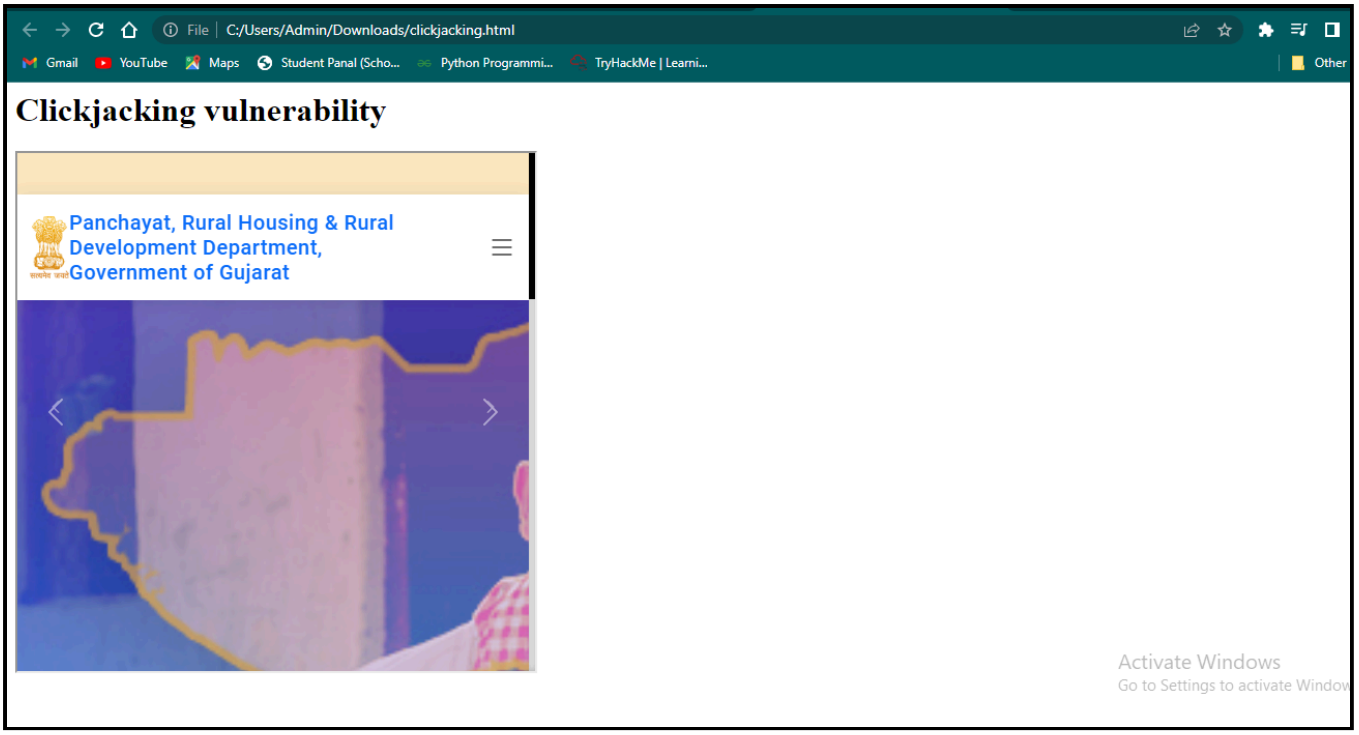


Figure 31 - As you can see that the application is vulnerable to clickjacking attack.

Observation ID & Title	8.20 TLS 1.0 And TLS 1.1 Enabled		
CVSS Risk Rating	Low	Status	Open

Observation Details

Transport Layer Security (TLS) are cryptographic protocols that provide communications security over a computer network. Several versions of the protocols find widespread use in applications such as web browsing, email, instant messaging, and voice over IP (VoIP). Websites can use TLS to secure all communications between their servers and web browsers. Since applications can communicate either with or without TLS (or SSL), it is necessary for the client to indicate to the server the setup of a TLS connection. During the testing of the web application, it was found that TLS version 1.0 / 1.1 was being used for communication in the web application. These versions, as discussed above are outdated.

Risk

TLS 1.0 version is vulnerable to many implementations and it fails to shield against attacks such as BEAST and POODLE. This version of TLS can be easily breached by the attackers.

TLS 1.1 is an outdated version. The pseudo random function in TLS is based on a combination on a MD5 and SHA-1.The attacker can easily break these function and in return can cause severe damage to the server. TLS 1.1 was defined in RFC 4346 in April 2006 It is an update from TLS version 1.0. Using even any one of the above mentioned versions of TLS, the attacker could access sensitive information, with attacks such MiTM, etc.

Recommendations

The Following are the recommendations that should be implemented.

As the web application is using an outdated version of TLS, it is highly recommended to update TLS to the latest version, TLS 1.3.

Affected URL & Parameter

Throughout the Application

CVSS Vector

3.0 CVSS:3.0/AV:L/AC:L/PR:N/UI:N/S:C/C:L/I:H/A:L

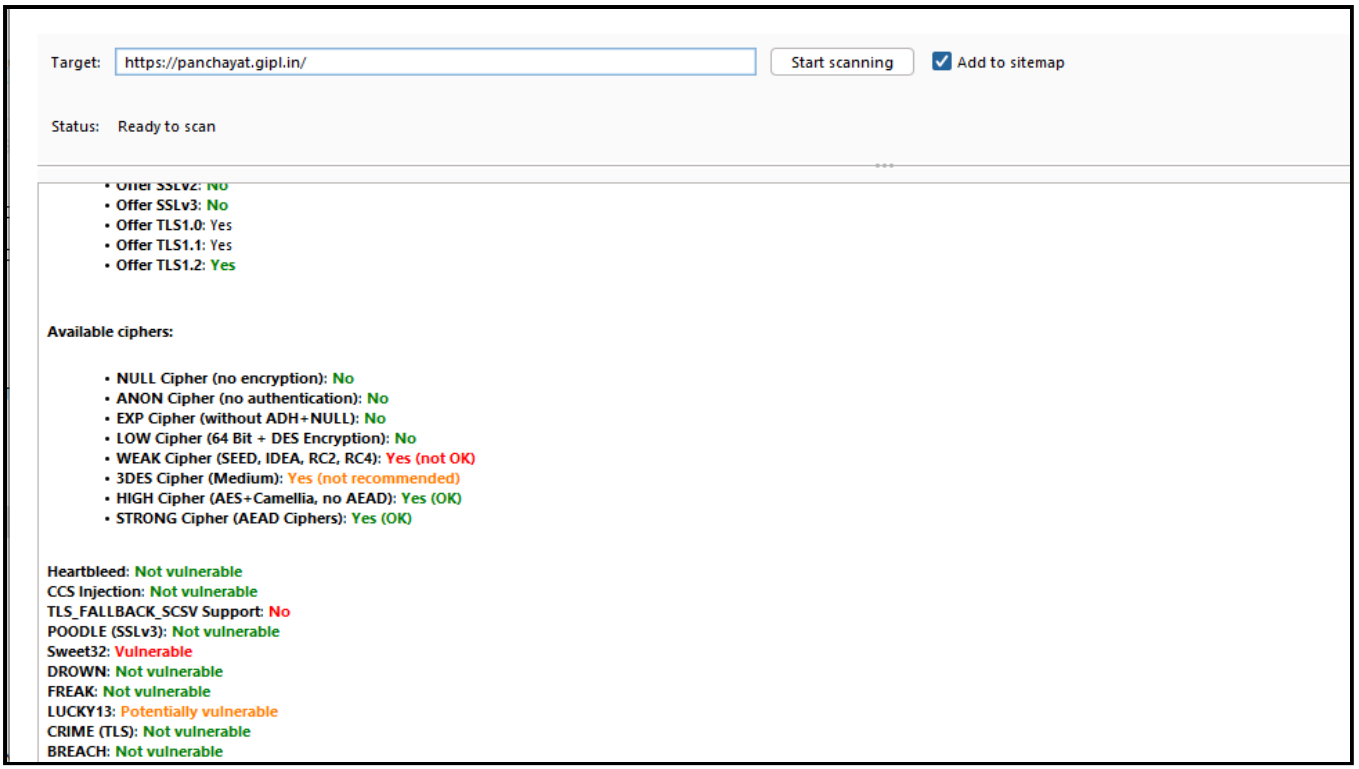


Figure 32 - As you can see TLS 1.0 & 1.1 is enabled.

Observation ID & Title	8.21 Sweet32 Attack		
CVSS Risk Rating	Low	Status	Open

Observation Details

The Sweet32 is an attack first found by researchers at the French National Research Institute for Computer Science (INRIA). The attack targets the design flaws in some ciphers that are used in TLS, SSH, IPsec, and OpenVPN. The Sweet32 attack allows an attacker to recover small portions of plaintext. It is encrypted with 64-bit block ciphers (such as Triple-DES and Blowfish), under certain circumstances. The SWEET32 attack can be used to exploit the communication that uses a DES/3DES based cipher suite. The SWEET32 attack affects the commonly used algorithm like AES (Advanced Encryption Standard), Triple-DES (Data Encryption Standard) and Blowfish for encrypting communication for TLS, SSH, IPsec and OpenVPN protocol. The given web application was found to be having this vulnerability.

Risk

An attacker can exploit this vulnerability to conduct many attacks. A man-in-the-middle attacker could use this flaw to recover some plaintext data. The attacker can steal large amounts of encrypted traffic between TLS/SSL server and client.

Recommendations

It is highly recommended to :-

Use OpenSSL security update RHSA-2016:1940.

Try to avoid the usage of legacy 64-bit block ciphers.

Use 128-bit ciphers for encryption in servers and VPNs.

Affected URL & Parameter

Throughout the Application

CVSS Vector

3.0 CVSS:3.0/AV:L/AC:L/PR:N/UI:N/S:C/C:L/I:H/A:L

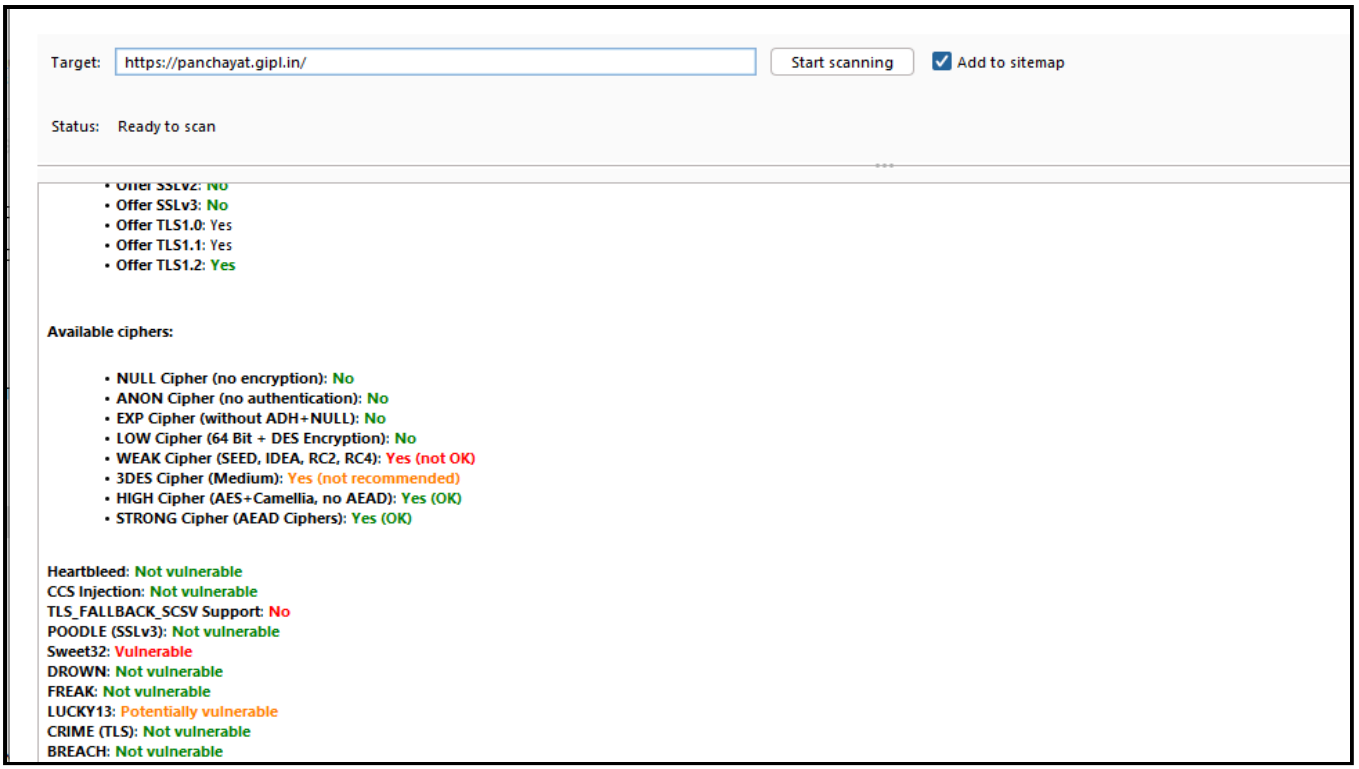


Figure 33 - As you can see that the application is vulnerable to TLS Sweet 32.

Observation ID & Title	8.22 Improper Input Validation		
CVSS Risk Rating	Low	Status	Open

Observation Details

While testing the web application we found out that this application is vulnerable to Improper Input Validation vulnerability. Improper Input Validation can allow an attacker to supply malicious user input that is then executed by the vulnerable web application. Improper input validation can be used to bypass security mechanisms, such as authentication and authorization controls. It can also be used to inject malicious code into the web application, which can be executed by the server or client.

Risk

Improper Input Validation allows an attackers to craft the input in a form that is not expected by the rest of the application. This will lead to parts of the system receiving unintended input, which may result in altered control flow, arbitrary control of a resource, or arbitrary code execution, leakage of sensitive information, injections attacks that split database.

Recommendations

The Following are the recommendations that should be implemented.

Sanitize data enter by user before processing.

They can be removed by constrain input, reject known bad input, validate data for type, length and range

Affected URL & Parameter

Throughout the Application

CVSS Vector

3.6 CVSS:3.0/AV:L/AC:L/PR:N/UI:N/S:C/C:L/I:H/A:L

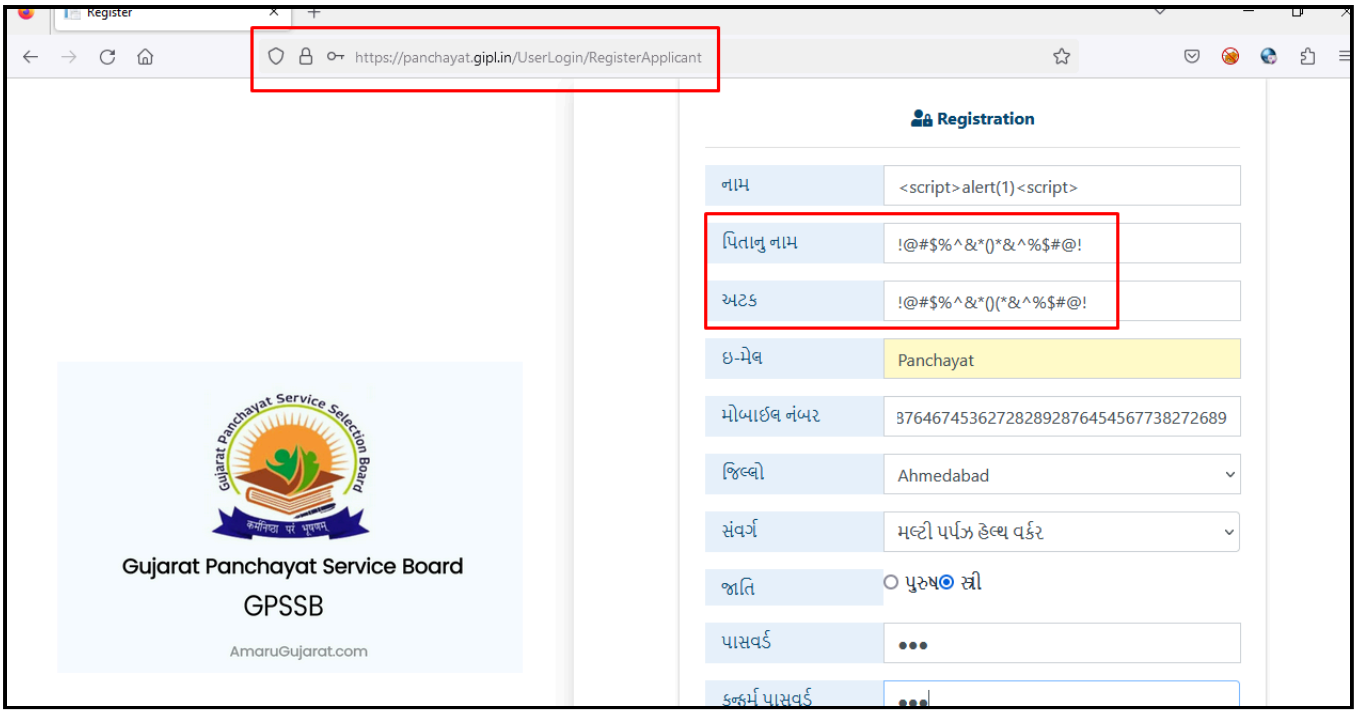


Figure 34 - As you can see that the application is accepting the special characters and those which has to be blacklisted.

9. Tools & Reference

9.1 Tools

The following tools were used for Application Security Testing

- Burp Suite Proxy
- Mozilla Web Browser with Firebug
- SQL Map
- IBM Rational AppScan
- Some other proprietary scripts and tools were also used.

9.2 References

The following tools were used for Application Security Testing

- For application security visit www.owasp.org
  - http://cert.org/other_sources/tool_sources.html
  - http://securityfocus.com/
  - http://projects.webappsec.org/w/page/13246978/Threat Classification
- Security advisories