CYBERPWN
EMPOWERING CYBER RESILIENCE

WEB APPLICATION PENETRATION TESTING REPORT

v1.1

Prepared for: Sutherland Global Services Pvt Ltd

17 June 2026

Disclaimer

This data shall not be disclosed and shall not be duplicated, used, or disclosed in whole or in part for any purpose. If a contract is awarded to 5Sec CyberPWN Technologies Pvt Ltd., as a result of or in connection with the submission of this data, the client or prospective client shall have the right to duplicate, use, or disclose this data to the extent provided in the contract. This restriction does not limit the client's or prospective client's right to use the information contained in the data if it is obtained from another source without restriction. The data subject to this restriction is contained in all marked sheets.

CONTENTS

DOCUMENT DETAILS **4**

INTRODUCTION **5**

SCOPE & DURATION 5

EXECUTIVE SUMMARY **6**

RISK RATING DEFINITIONS 8

APPROACH AND METHODOLOGY **9**

RESULT OVERVIEW **12**

HIGH LEVEL RECOMMENDATIONS **13**

VULNERABILITIES DETAILS: **14**

1. BROKEN AUTHENTICATION 14
2. HTML INJECTION 16
3. RATE LIMITING IS NOT IMPLEMENTED 18
4. PASSWORD REFLECTED IN PLAIN TEXT 20
5. BUSINESS LOGIC FLAW 21
6. INFORMATION DISCLOSURE 23
7. MISCONFIGURED CSP SECURITY HEADER 24
8. CONCURRENT LOGIN ENABLED 25
9. USER ENUMERATION 26
10. VULNERABLE COMPONENT IN USE 28

CONCLUSION **29**

ANNEXURE A – CHANGES TO ENVIRONMENT **30**

ANNEXURE B – TOOLS USED **30**

ANNEXURE C - LIST OF VAPT TESTS PERFORMED **31**

DOCUMENT DETAILS

DOCUMENT CONTROL	
Document Title	**Web Application Penetration Testing Report**
Document Classification	Final Report
Last Edit Date	17-June-2026

DOCUMENT HISTORY			
DATE	**VERSION**	**PREPARED BY**	**STATUS**
14-May-2026	1.0	Arham Jain	Initial Report
17-June-2026	1.1	Arham Jain	Final Report

CUSTOMER INFORMATION	
Company Name	Sutherland Global Services Pvt Ltd
Address	Gateway Office Parks Pvt. Ltd 2nd, 3rd, 4th ,5th Floor, Block B2 No:16, GST Road, Perungalathur Chennai - Tamil Nadu – 600063 - India
Website	https://www.sutherlandglobal.com/
Contact Name	Sarfraz Amin
Title	Assistant Vice President
Email	sarfrazamin.a@sutherlandglobal.com

CONSULTANT INFORMATION		
Name	**Role**	**Responsibility**
Arham Jain	Lead Security Engineer	Document Preparation
Prabhu G	Lead Security Engineer	Document Review
Anuranjan Garnaik	Senior Manager	Document Final Approval
Sarfraz Amin	Assistant Vice President	Document Acceptance

INTRODUCTION

CyberPWN Technologies conducted **Web Application Penetration testing** for Sutherland Global Services Pvt Ltd. Based upon the Authorization to Test the application provided by the Company, CyberPWN followed a testing methodology that sought to identify vulnerabilities and, through manual pen testing determine the impact to the Company's asset. CyberPWN assigned a risk level based on goals achieved during testing.

SCOPE & DURATION

The scope includes **Web Application Penetration testing** for below mentioned target.

Application Name	Application URL
Robility7	https://robility7.sutherlandglobal.com/Robility

The test was performed from 11-May-2026 to 14-May-2026 including reporting.
The Re-Test was performed on 01-June-2026 & 17-June-2026.

LIMITATIONS & CONSTRAINTS

Testing occurred under the following constraints:

- The assessment was performed with the knowledge shared by the Sutherland team about the application.
- The assessments were conducted from CyberPWN and the findings made are highly subjective to target application and services visibility and availability at that given point of time.

EXECUTIVE SUMMARY

CyberPWN was engaged by Sutherland to conduct grey-box web application penetration testing of external facing web application. The security assessment covered a web application that was conducted in Bangalore, India.

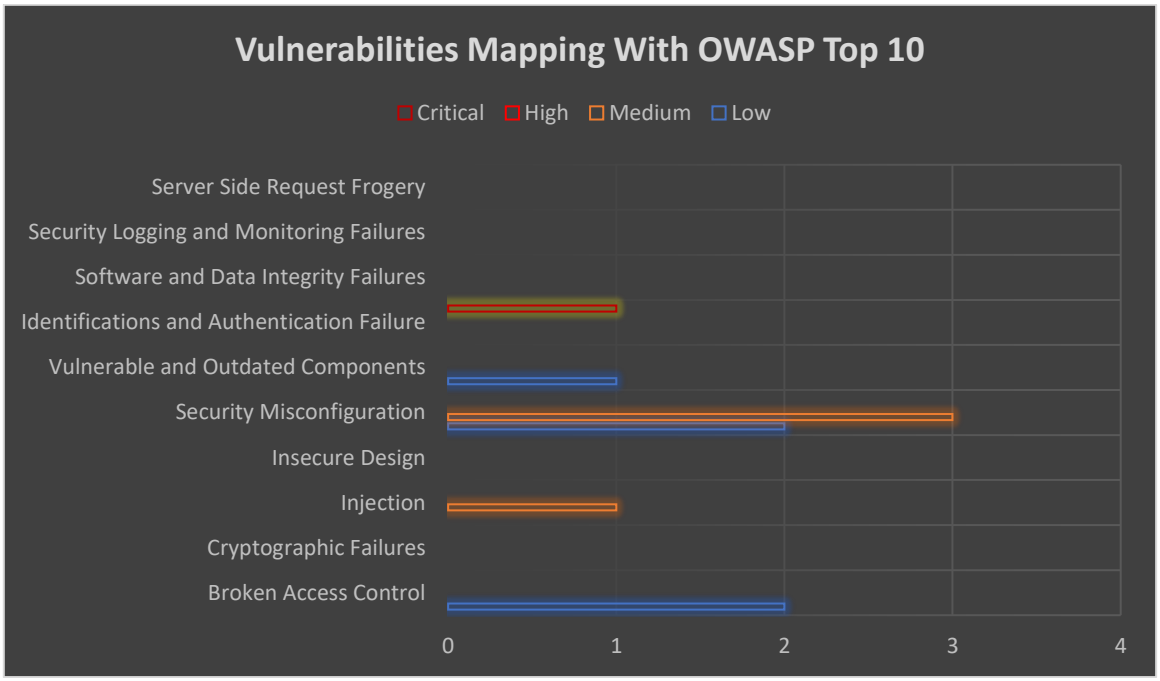
However, the assessment identified 01 High, 05 Medium and 04 Low risk findings as an outcome of application penetration testing.

- The 01 Critical rated risks pertaining to exploitation vulnerability is Broken Authentication.
- The 05 Medium rated risks pertaining to exploitation vulnerabilities are HTML Injection, Rate Limiting is Not Implemented, Business Logic Flaw, Information Disclosure & Password Reflected in Plain text.
- The 04 Low rated risks pertaining to vulnerabilities are Misconfigured CSP, Vulnerable Components in use, User Enumeration and Concurrent Login Enabled.

Summary of Finding

Total Vulnerabilities						
Testing	Critical	**High**	**Medium**	**Low**	**Info**	**Total**
Assessment	01	**00**	**05**	**04**	**00**	**10**
Revalidation	**00**	**00**	**00**	**00**	**00**	**00**

Finding Categorization



Introduction to Web Application Penetration Testing:

Web Application Penetration Testing focuses on evaluating the security of web application by simulating attacks from both external and internal threats. This assessment aims to uncover vulnerabilities, weaknesses, misconfigurations, and potential attack vectors in the application like cross-site request forgery, XXE, injection vulnerabilities, transport layer encryption weakness etc.

Vulnerability Scoring

A scoring system is used to grade all of the vulnerabilities listed in this report. Cyberpwn employs the industry-standard CVSSv4.0. It provides a system for determining the severity of vulnerabilities, regardless of the software/hardware platform or service function.

Every vulnerability is assigned a score between 0 and 10, giving each discovered vulnerability a score that aids in identifying the most vulnerable systems and prioritizing responses to each problem. The National Vulnerabilities Database (NVD) uses the CVSS system to calculate scores for almost all known vulnerabilities, and these are the scores referred to in this report.
Further information can be found at https://www.first.org/cvss/calculator/4.0
https://nvd.nist.gov/

Severity Rating

Based on the severity of the vulnerability, they are assigned below ratings:

CVSS Severity Rating	CVSS Score
Critical	9.0 - 10.0
High	7.0 - 8.9
Medium	4.0 - 6.9
Low	0.1 - 3.9
Informational	0.0

RISK RATING DEFINITIONS

Severity Rating	Definitions
Critical	Critical severity vulnerabilities usually have most of the following traits: • A successful attack may lead to complete compromise of the system • Exploitation can be done remotely over an untrusted connection – such as the Internet • Exploiting the vulnerability is very easy or straightforward. It may not require privilege accounts or user interaction. • It has a very significant impact on confidentiality, integrity and/or availability of the targeted system
High	High severity vulnerabilities demonstrate some of the following characteristics: • It is not straight-forward to exploit and may require some user interaction and has minimum dependency. • Usually gives an attacker elevated privilege. • It has a significant impact on confidentiality, integrity and availability
Medium	• Vulnerabilities scored medium generally: • Require specific user privileges or conditions to execute the attack. • Grant attacker access to non-critical privileged functionality/data. • Moderately affects Confidentiality, Integrity or availability.
Low	Vulnerabilities scored low usually exhibit some of the following characteristics: • They have little impact on the confidentiality, integrity or availability of the organization's data or assets. • Requires a great amount of computational power to exploit the vulnerability. • Require excessive privileges or access to execute an attack • Exploits are not known publicly

APPROACH AND METHODOLOGY

Cyberpwn Technologies penetration testing methodology is based upon frameworks and standards mentioned below and it contains the following phases:

OWASP TOP 10
SANS TOP 25
The Penetration Testing Execution Standard (pentest-standard.org)
National Institute of Standards and Technology (NIST)
OSSTMM (Open Source Security Testing Methodology Manual)



PLANNING

- Cyberpwn Technologies prepares for initial planning sessions with the Sutherland global by reviewing the Company's business processes, key personnel, physical locations and Internet-accessible footprint.
- Cyberpwn Technologies and the Company collaborate to create the rules, attack scenarios, and goals for testing.
- The Company may provide additional documentation and access to applications, systems and networks to facilitate targeted testing.
- The Customer Company is responsible for ensuring that the scope contains all targets for testing and that the Company has the authority to permit Cyberpwn Technologies to perform penetration testing against the identified targets.

The assessment was done in a 'Grey-Box' method, in which consultants had no knowledge of the target systems. This type of review focuses on real world attacks from out siders. The assessment is performed by using manual approach followed by automated tools and scripts for identifying and further exploiting identified vulnerabilities. After the security audit, a detailed report with findings, remediation recommendations will be delivered.

Information Gathering

All security testing assessment starts with information gathering. We use Open-source intelligence (OSINT) framework to collect data from publicly available sources to be used in an intelligence context. Through information gathering a great deal of actionable and predictive intelligence can be obtained from public, open source, unclassified sources.

Enumeration

This process begins with detailed scanning and research into the architecture and environment, to discover potential attack vectors in the system, and the same can be used for further exploitation of the system.

Automated Testing

Once the target has been fully enumerated, the assessment team uses both vulnerability scanning tools and manual analysis to identify security flaws. With vast experience, in-depth technical knowledge and custom-built tools, our security engineers find weaknesses most automated scanners miss.

Exploration and Verification

At this stage of the assessment, our consultants review all previous data to identify and safely exploit identified application vulnerabilities. Once sensitive data has been obtained, the focus turns to escalation and movement to identify technical risk and total business impact. During each phase of the compromise, we keep client stakeholders informed of testing progress, ensuring asset safety and stability.

Privilege Escalation

Once a vulnerability is exploited, the privilege accrued through the exploitation is further exploited to gain higher privilege or escalate the access level. Privileges escalation demonstrates real world threat and attack to system in scope and other systems on the connected network.

Reporting

Cyberpwn Technologies regularly communicates on the progress and results of testing during the engagement. Cyberpwn Technologies immediately notifies the Client if a critical-risk finding is discovered so that the Company team can quickly remediate the issue.

Cyberpwn Technologies creates a report that contains, at minimum, the following items:

- **Executive Summary** - provides a high-level overview of the testing results and is intended to be read by executives, customers, and business partners.
- **Findings** - describes each exploitable vulnerability. The findings results are intended to be distributed to technical teams

- **Recommendations** - recommendations on how to resolve each identified issue
- **Risk Ranking** - each issue identified is assigned a risk ranking that is derived from the Common Vulnerability Scoring System (CVSS). The rating is based on the specific instances identified in the company environment.
- **Steps to Reproduce** - additional details that provide enough information such that the issue can be replicated by technical teams
- **Rescan** - updates about the finding, such as retesting status or management responses and revalidation report.

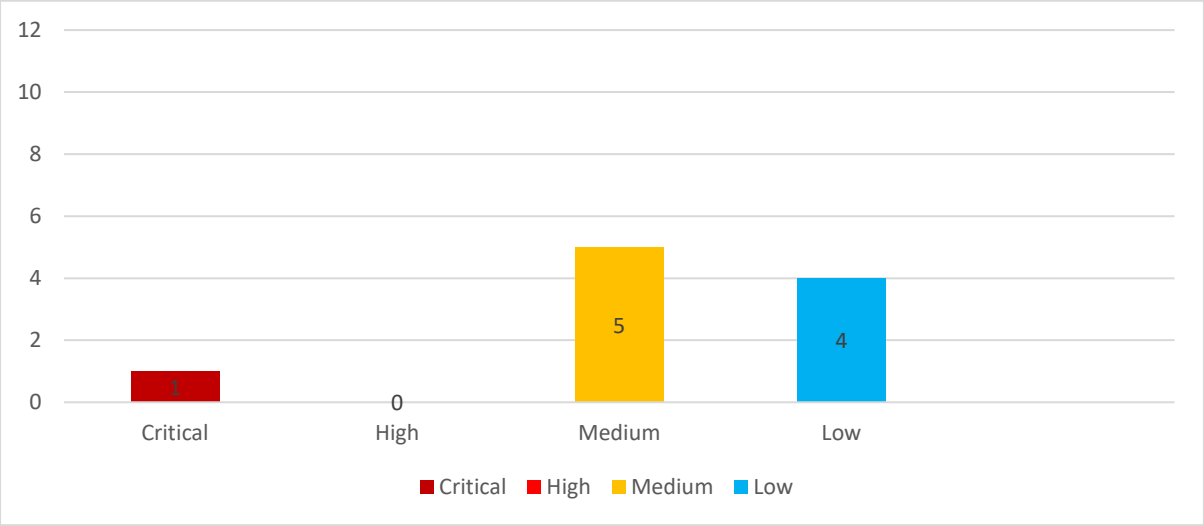
RESULT OVERVIEW

CyberPWN Technologies Security team discovered 10 risks and potential vulnerabilities in https://robility7.sutherlandglobal.com/Robility web Application.

The below table summarizes the list of vulnerabilities with corresponding risk ratings.

SL. No.	Vulnerability Name	Severity Rating	Revalidation status
1.	Authentication Bypass	Critical	CLOSED
2.	HTML Injection	Medium	CLOSED
3.	No Rate Limiting	Medium	CLOSED (EXCEPTION)
4.	Password in Plain Text	Medium	CLOSED
5.	Business Logic Flaw	Medium	CLOSED
6.	Information Disclosure	Medium	CLOSED
7.	Misconfigured CSP	Low	CLOSED (EXCEPTION)
8.	Concurrent Login	Low	CLOSED (EXCEPTION)
9.	User Enumeration	Low	CLOSED
10.	Vulnerable Component in use	Low	CLOSED

GRAPHICAL PRESENTATION



HIGH LEVEL RECOMMENDATIONS

The following recommendations offer guidance on enhancing the security posture of https://robility7.sutherlandglobal.com/Robility Web Application:

- Never return the PIN in the API response
- Implement input validation and sanitization on the front end and server side of the application.
- Ensure all user-supplied data is sanitized before CSV export by preventing cells that begin with =, +, -, or @ from being interpreted as formulas.
- Implement rate limiting on all endpoints to restrict excessive requests from a single user or IP.
- Implement rate limiting, account lockout policies.
- Update, and patch outdated or vulnerable components.
- Restrict or monitor concurrent sessions by enforcing session limits and implementing session management controls to prevent unauthorized simultaneous logins.
- Configure security headers properly.

VULNERABILITIES DETAILS:

1. BROKEN AUTHENTICATION	
Application / URL	https://robility7.sutherlandglobal.com/Robility
Revalidation Status	CLOSED
Severity	Critical
CVSS Score	9.1
CVSS String	CVSS:3.1/AV:N/AC:L/PR:N/UI:N/S:U/C:H/I:H/A:N
Description	During the security assessment, it was observed that while trying to login with the login PIN option, the application is sending login PIN in response body. An attacker can log in to a user's account without having access to their email account or the account password.
Impact	Attackers can read the JSON response in intercepted traffic and use it to complete the login process immediately resulting in complete account takeover.
Remediation	It is Recommended to: • Never return the PIN in the API response. Only return a success/failure status code (e.g., {"success":true}). • Server-Side Validation: Ensure all PIN validation occurs on the server, not the client-side, to prevent response tampering. • Masking & Security: Ensure user data is masked in the UI and validate all 2FA inputs securely.
Reference	https://valencynetworks.com/kb/broken-authentication-otp-in-api-response-vulnerability-risks-impact-fix.html

Proof of Concept:

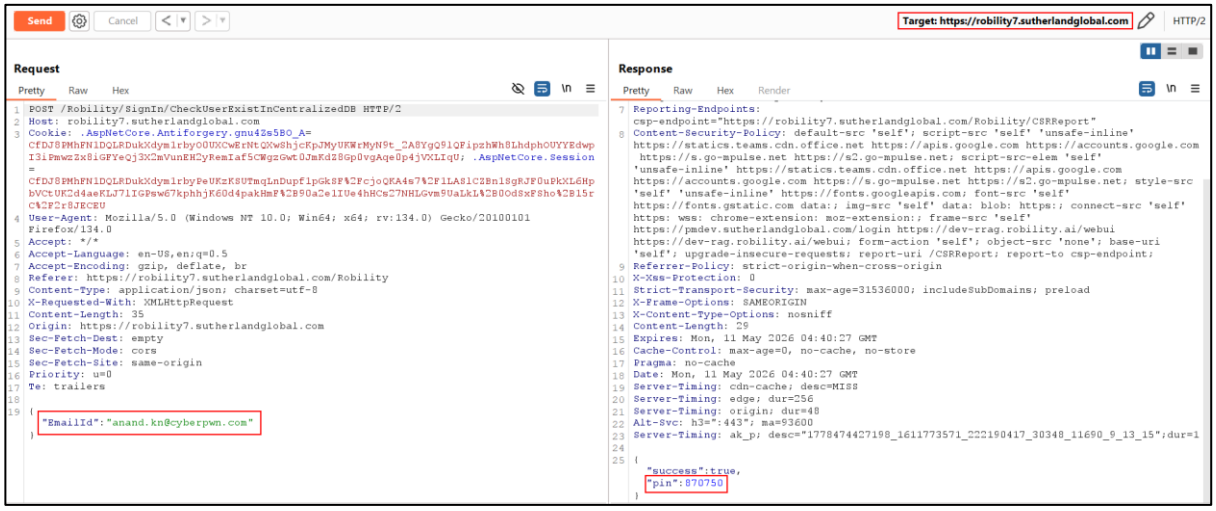


FIGURE 1: LOGIN PIN RECEIVED IN RESPONSE USING PROXY TOOL

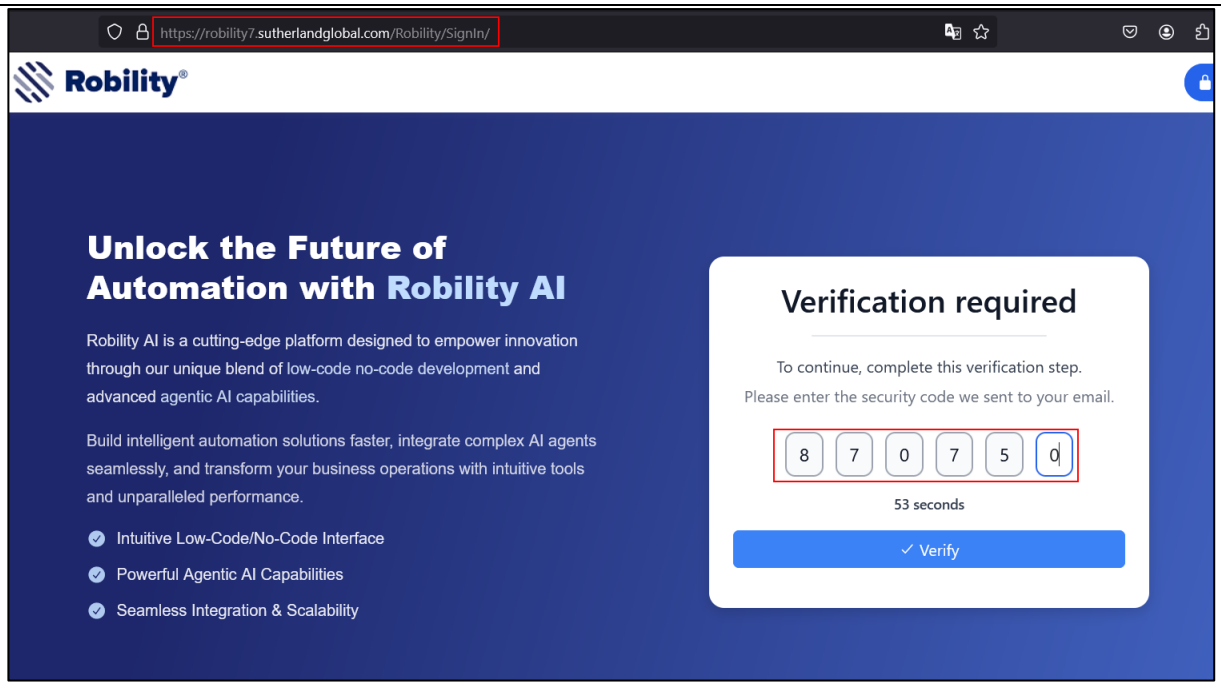


FIGURE 2: ENTERED THE PIN INTERCEPTED USING PROXY TOOL

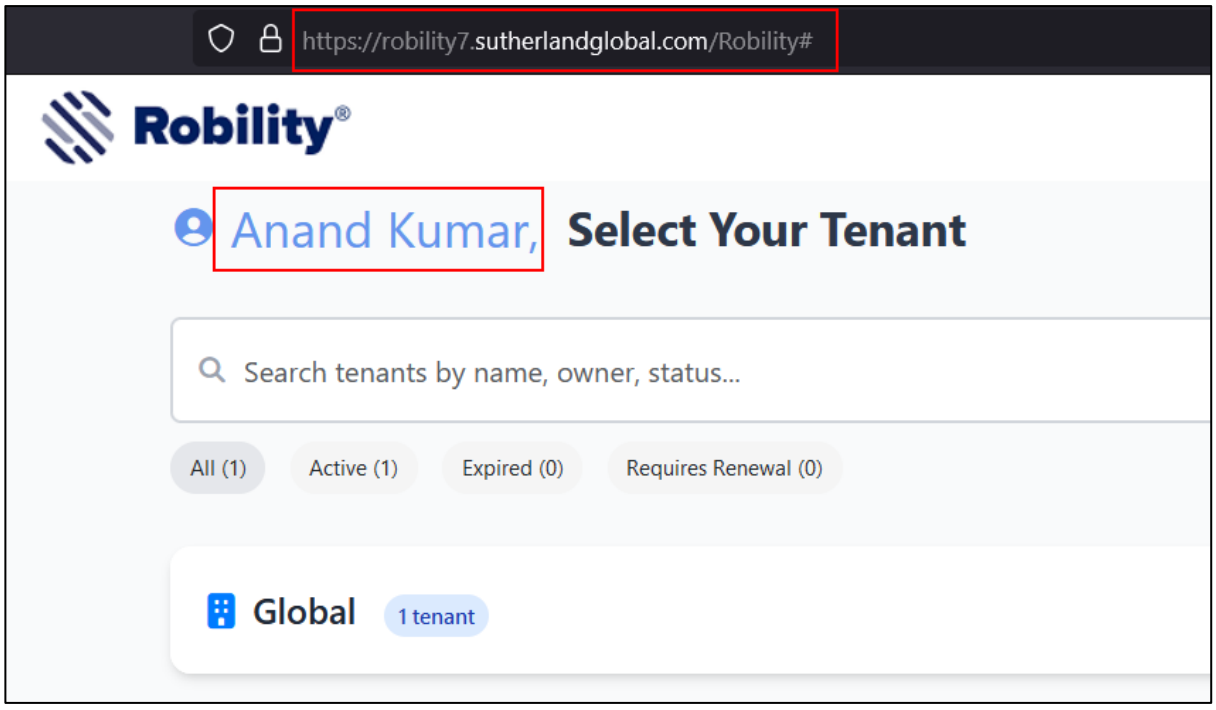


FIGURE 3: SUCCESSFULLY LOGGED INTO THE APPLICATION

2. HTML INJECTION	
Application / URL	https://robility7.sutherlandglobal.com/Robility/ProjectManagement/Project
Revalidation Status	CLOSED
Severity	Medium
CVSS Score	5.8
CVSS String	CVSS:4.0/AV:N/AC:H/AT:N/PR:L/UI:A/VC:H/VI:L/VA:L/SC:N/SI:N/SA:N
Description	During analysis, it was observed that the web application is vulnerable to HTML Injection Attack. This attack occurs on the input parameter fields. **Note: All the input parameters must be properly validated.**
Impact	HTML Injection is an attack that is similar to Cross-site Scripting (XSS). While in the XSS vulnerability the attacker can inject and execute Javascript code, the HTML injection attack only allows the injection of certain HTML tags. • Attacker discovers injection vulnerability and decides to use an HTML injection attack • Attacker crafts malicious link, including his injected HTML content, and sends it to a user via email.
Remediation	It is Recommended that: • Encode output based upon input parameters. For instance, the character "< "would be encoded as "&lt" and, although appearing to the user as the less-than character, would not be interpreted by the client application as the start of a HTML tag. • Properly encode the special characters (" ' > < &) by using "htmlspecialchars" function. • Certain characters are of special significance when inserted into web pages or URL content. These characters are based upon the HTML specifications, context, and browser interpretation. If input to the application (or web site) is not correctly validated HTML Injection may occur. • Whitelist parameter values i.e., accept only the known good. The recommended method of implementing input filtering is to only select from the set of characters that is known to be safe rather than excluding the named special characters. • Regular expressions or escaping can be utilized to achieve above techniques. In case, application requires to process special characters for business requirements, use HTML and URL encoding techniques while displaying the response on the browser. Your script should filter metacharacters from user input. **Note:** This recommendation should be implemented in all the similar parameters that are present with in the application.
Reference	https://www.acunetix.com/vulnerabilities/web/html-injection/

Proof of Concept:C

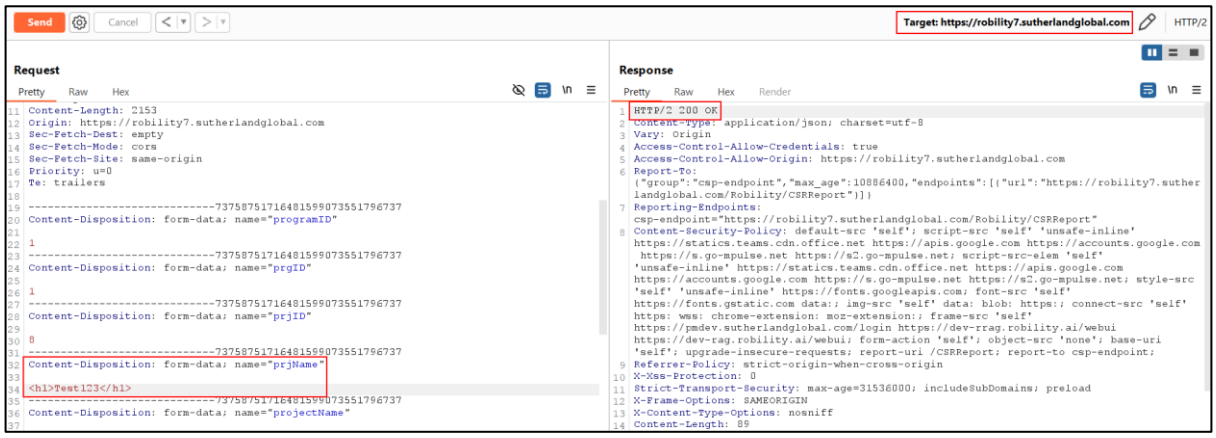


FIGURE 4: ADDED HTML PAYLOAD IN INPUT FIELDS

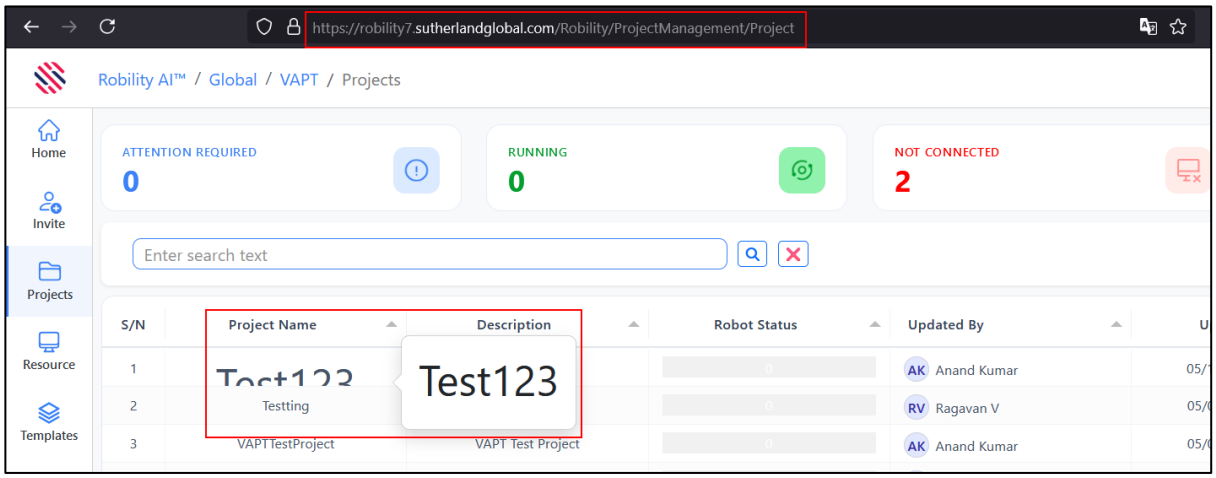


FIGURE 5: HTML PAYLOAD EXECUTED

3. RATE LIMITING IS NOT IMPLEMENTED

Application / URL	https://robility7.sutherlandglobal.com/Robility
Revalidation Status	**CLOSED (EXCEPTION)**
Severity	Medium
CVSS Score	5.3
CVSS String	CVSS:4.0/AV:N/AC:L/AT:N/PR:L/UI:N/VC:N/VI:N/VA:L/SC:N/SI:N/SA:N
Description	During the assessment, it was observed that the application is vulnerable to Rate Limiting is not implemented. Able to create multiple template at a time and getting 200 success response for all requests. **Note: Remediate in all necessary endpoints.**
Impact	**Denial of Service (DoS):** Attackers can flood the API with high volumes of requests, overwhelming the server and causing downtime or slow response for legitimate users. **Resource Exhaustion**: The server's computational resources, such as CPU, memory, and storage, can be quickly exhausted, leading to potential crashes or the need for expensive scaling

Remediation	It is recommended to implement the following
	• Add in rate limits preventing the request being submitted multiple times by the same user at the single time.
	• Implement Rate Limit Headers:
	  ○ X-RateLimit-Limit:
	  ○ X-RateLimit-Remaining:
	  ○ X-RateLimit-Reset:

Reference	https://cheatsheetseries.owasp.org/cheatsheets/Denial_of_Service_Cheat_Sheet.html

Proof of Concept:

Case 1:

Request	Payload	Status code	Response received	Error	Timeout	Length
12	null	200	401			2164
13	null	200	916			2163
14	null	200	936			2164
15	null	200	362			2163
16	null	200	387			2163
17	null	200	370			2163
18	null	200	381			2163
19	null	200	385			2163
20	null	200	894			2163

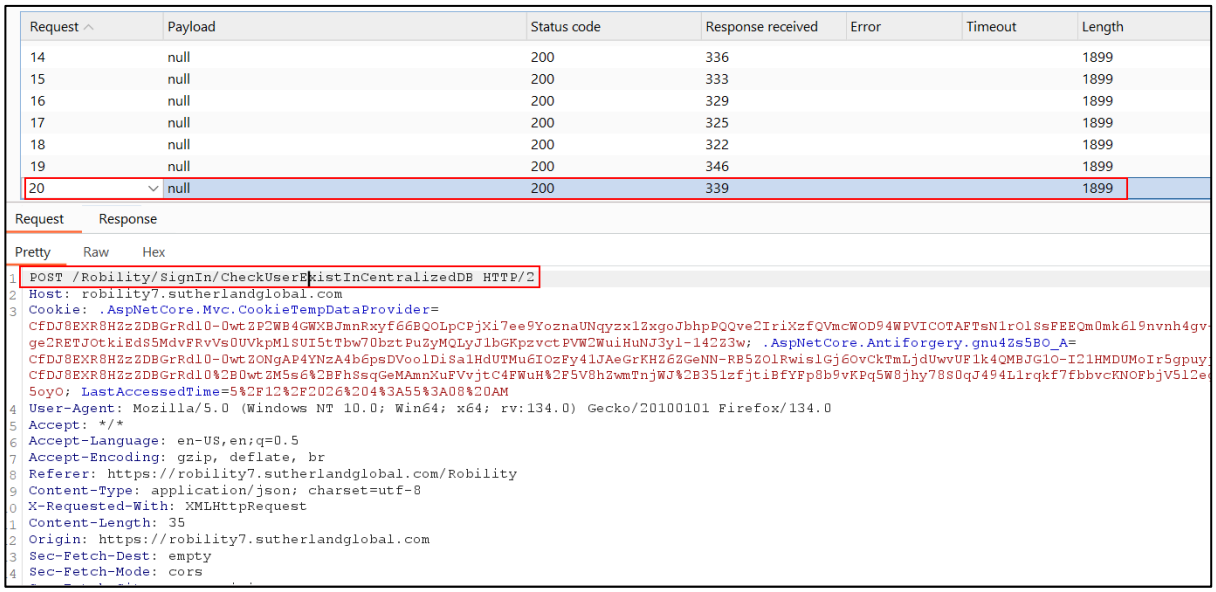
Request Response

Pretty Raw Hex Render

```
HTTP/2 200 OK
Content-Type: application/json; charset=utf-8
Vary: Origin
Access-Control-Allow-Credentials: true
Access-Control-Allow-Origin: https://robility7.sutherlandglobal.com
Report-To: {"group":"csp-endpoint","max_age":10886400,"endpoints":[{"url":"https://robility7.sutherlandglobal.com/Robility/CSRReport"}]}
Reporting-Endpoints: csp-endpoint="https://robility7.sutherlandglobal.com/Robility/CSRReport"
Content-Security-Policy: default-src 'self'; script-src 'self' 'unsafe-inline' https://statics.teams.cdn.office.net https://apis.google.c
https://s.go-mpulse.net https://s2.go-mpulse.net; script-src-elem 'self' 'unsafe-inline' https://statics.teams.cdn.office.net https://api
https://s.go-mpulse.net https://s2.go-mpulse.net; style-src 'self' 'unsafe-inline' https://fonts.googleapis.com; font-src 'self' https://
data: blob: https:; connect-src 'self' https: wss: chrome-extension: moz-extension:; frame-src 'self' https://pmdev.sutherlandglobal.com/
https://dev-rag.robility.ai/webui; form-action 'self'; object-src 'none'; base-uri 'self'; upgrade-insecure-requests; report-uri /CSRRepo
Referrer-Policy: strict-origin-when-cross-origin
X-Xss-Protection: 0
Strict-Transport-Security: max-age=31536000; includeSubDomains; preload
X-Frame-Options: SAMEORIGIN
X-Content-Type-Options: nosniff
```

FIGURE 6: GETTING 200 OK AFTER 20 CONSECUTIVE FORM SUBMISSIONS

Case 2:



Request	Payload	Status code	Response received	Error	Timeout	Length
14	null	200	336			1899
15	null	200	333			1899
16	null	200	329			1899
17	null	200	325			1899
18	null	200	322			1899
19	null	200	346			1899
20	null	200	339			1899

Request Response

Pretty Raw Hex

```
POST /Robility/SignIn/CheckUserExistInCentralizedDB HTTP/2
Host: robility7.sutherlandglobal.com
Cookie: .AspNetCore.Mvc.CookieTempDataProvider=
CfDJ8EXR8HZzZDBGrRdl0-0wtZP2WB4GWXBJmnRxyf66BQOLpCPjXi7ee9YoznaUNqyzx1ZxgoJbhpPQQve2IriXzfQVmcWOD94WPVICOTAFTsN1rOlSsFEEQm0mk6l9nvnh4gv
ge2RETJOtkiEdS5MdvFRvVsOUVkpMlSUI5tTbw7ObztPuZyMQLyJlbGKpzvctPVW2WuiHuNJ3yl-14223w; .AspNetCore.Antiforgery.gnu4Zs5BO_A=
CfDJ8EXR8HZzZDBGrRdl0-0wtZONgAP4YNzA4b6psDVoolDiSa1HdUTMu6IOzFy41JAeGrKH262GeNN-RB52OlRwislGj6OvCkTmLjdUwvUFlk4QMBJG1O-I21HMDUMoIr5gpuy
CfDJ8EXR8HZzZDBGrRdl0%2B0wtZM5s6%2BFhSsqGeMAmnXuFVvjtC4FWuH%2F5V8hZwmTnjWJ%2B351zfjtiBfYFp8b9vKPq5W8jhy78S0qJ494L1rqkf7fbbvcKNOFbjV5l2e
5oyO; LastAccessedTime=5%2F12%2F2026%204%3A55%3A08%20AM
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:134.0) Gecko/20100101 Firefox/134.0
Accept: */*
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate, br
Referer: https://robility7.sutherlandglobal.com/Robility
Content-Type: application/json; charset=utf-8
X-Requested-With: XMLHttpRequest
Content-Length: 35
Origin: https://robility7.sutherlandglobal.com
Sec-Fetch-Dest: empty
Sec-Fetch-Mode: cors
```

FIGURE 7: LOGIN VIA PIN REQUEST SENT MULTIPLE TIMES

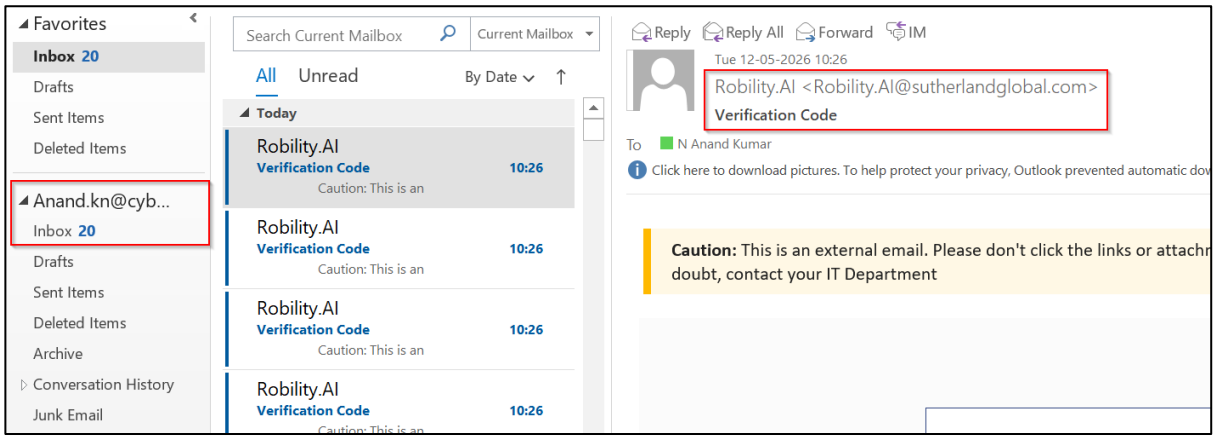


FIGURE 8: RECEIVED 20 PIN EMAILS RESULTING IN EMAIL FLOODING

4. PASSWORD REFLECTED IN PLAIN TEXT

Application / URL	https://robility7.sutherlandglobal.com/Robility/RoboMonitoring/Machine
Revalidation Status	CLOSED
Severity	Medium
CVSS Score	5.1
CVSS String	CVSS:4.0/AV:N/AC:L/AT:N/PR:H/UI:N/VC:N/VI:N/VA:L/SC:N/SI:N/SA:N
Description	During the analysis, it was observed that the application is reflecting machine password on the resource details page in response.
Impact	Anyone with access to the client-side environment—including through browser Developer Tools, malicious browser extensions, or intercepting proxies can see the password in plain text. Many infrastructure tools (load balancers, reverse proxies, and SIEMs) log response bodies for debugging. A plaintext password in the response will be recorded in these logs, exposing it to any administrator or attacker with log access.
Remediation	It is recommended to implement the below points: • Replace plaintext storage with a strong, adaptive hashing algorithm. • Authenticate by comparing the hash of the user-provided password with the stored hash; never retrieve the original password. • Ensure API models specifically exclude the password field from being serialized into JSON or HTML responses.
Reference	https://github.com/ChurchCRM/CRM/security/advisories/GHSA-p98h-5xcj-5c6x

Proof of Concept:



FIGURE 9: PASSWORD REFLECTED IN PLAIN TEXT

5. BUSINESS LOGIC FLAW	
Application / URL	https://robility7.sutherlandglobal.com/Robility/UserManagement/InviteUsers
Revalidation Status	CLOSED
Severity	Medium
CVSS Score	5.1
CVSS String	CVSS:4.0/AV:N/AC:L/AT:N/PR:H/UI:N/VC:N/VI:N/VA:L/SC:N/SI:N/SA:N
Description	During the analysis, it was observed that the application is defying the business logic. On the invite user page, the role selection functionality can be manipulated to assign both the RPA Developer & Tenant Admin role to the same user.
Impact	An attacker can intercept the request & change the RoleIds parameter to assign all the roles to a particular user hence defying the business logic of the application.
Remediation	It's recommended to implement strict server-side authorization checks to ensure that users can only assign roles they are explicitly permitted to manage. The application should not rely on client-side controls or request parameters for access control decisions.
Reference	https://owasp.org/Top10/2021/A01_2021-Broken_Access_Control/index.html

Proof of Concept:

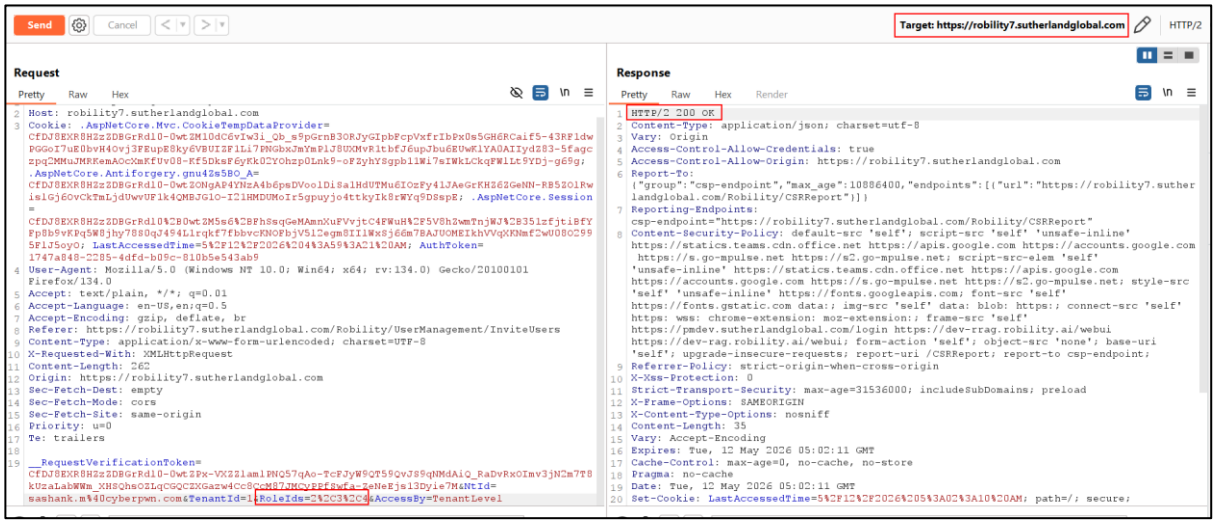


FIGURE 10: INTERCEPTED THE ROLE UPDATE REQUEST & CHANGED THE ROLEIDS

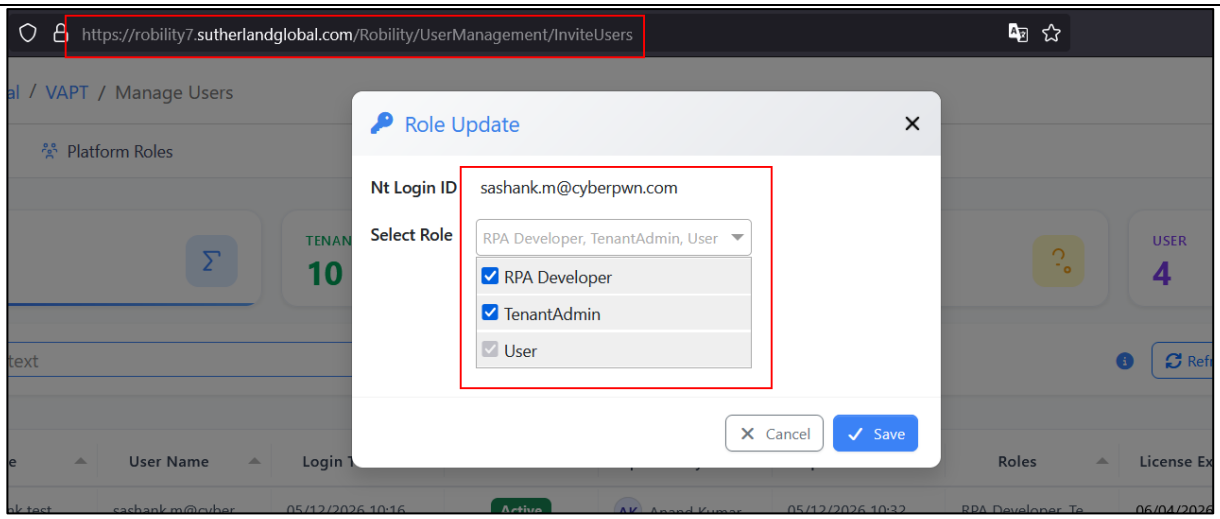


FIGURE 11: USER ASSIGNED WITH BOTH RPA DEVELOPER & TENANT ADMIN, DEFYING THE BUSINESS LOGIC

6. INFORMATION DISCLOSURE

Application / URL	https://robility7.sutherlandglobal.com/Robility/SecOps/StorageType
Revalidation Status	CLOSED
Severity	Medium
CVSS Score	5.3
CVSS String	CVSS:3.1/AV:N/AC:L/PR:N/UI:N/S:U/C:L/I:N/A:N
Description	During assessment, it was observed that application is disclosing internal information in the response body.
Impact	Attackers use automated tools to scan public repositories and website files for keys. Once found, they can impersonate your application or user account. Depending on the key's permissions, attackers can read, modify, or delete sensitive user data, financial records, or proprietary internal information.
Remediation	It is recommended to: • Store all sensitive credentials in environment variables rather than hardcoding them in source files. • Never store API keys or database credentials directly in frontend JavaScript code or public repository files.
Reference	https://checkmarx.com/learn/how-to-detect-and-remove-leaked-api-keys-tokens-and-passwords-from-code-repositories/

Proof of Concept:

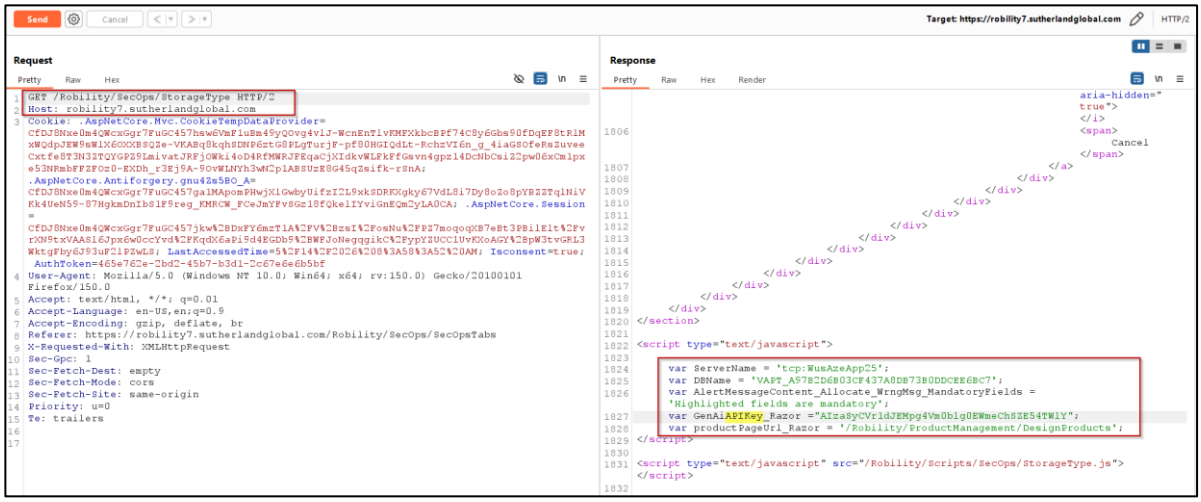


FIGURE 12: INFORMATION DISCLOSURE

7. MISCONFIGURED CSP SECURITY HEADER

Application / URL	https://robility7.sutherlandglobal.com/Robility
Revalidation Status	**CLOSED (EXCEPTION)**
Severity	Low
CVSS Score	3.0
Description	During analysis, it was observed that CSP security header is misconfigured. A Content Security Policy (CSP) requires careful tuning and precise definition of the policy. By using unsafe-eval, all the application will allow the use of string evaluation functions like eval. By using unsafe-inline, the application will allow the execution of inline scripts, which almost defeats the purpose of CSP.
Impact	When this is allowed, it's very easy to successfully exploit a Cross-site Scripting vulnerability on the website. An attacker can bypass CSP and exploit a Cross-site Scripting vulnerability successfully. script-src: Restricts the script loading resources to the ones it is declared. By default, it disables inline script executions unless it is permit to the evaluation functions and inline scripts by the unsafe-eval and unsafe-inline keywords.
Remediation	It is recommended to implement following features in HTTP Response • Remove unsafe-eval and unsafe-inline from CSP directives. • Move JavaScript into its own file • Switch to Report-Only and use a report-uri service • Use Hash / Nonce • Whitelist the domains that are allowed to inject the scripts instead of using (*)
CVSS String	CVSS:3.0/AV:N/AC:L/PR:N/UI:R/S:C/C:L/I:L/A:N
Reference	https://cheatsheetseries.owasp.org/cheatsheets/Content_Security_Policy_Cheat_Sheet.html

Proof of Concept:

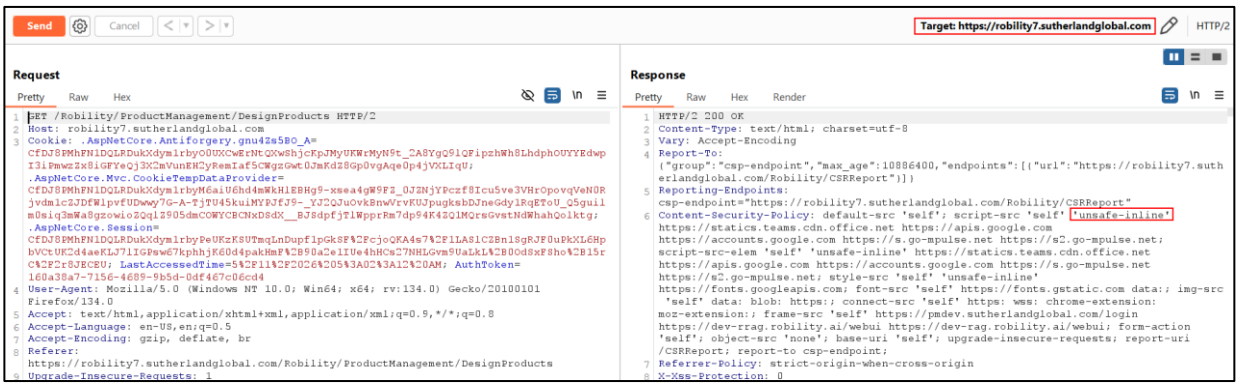


FIGURE 13: MISCONFIGURED CSP SECURITY HEADER

8. CONCURRENT LOGIN ENABLED

Application / URL	https://robility7.sutherlandglobal.com/Robility
Revalidation Status	CLOSED (EXCEPTION)
Severity	Low
CVSS Score	2.1
CVSS String	CVSS:4.0/AV:N/AC:H/AT:N/PR:H/UI:N/VC:L/VI:L/VA:L/SC:N/SI:N/SA:N
Description	During assessment, it was observed that the application is not restricting concurrent login requests. Although this is not treated as a vulnerability in many low-risk applications, we recommend not to allow the same. However, this can be relaxed based on the criticality of the application.
Impact	This could lead to security issues within the organization like misuse of the user's personal information or resources to perform unauthorized actions.
Remediation	It is recommended that not to allow concurrent logins to the application or user should be alerted through pop up messages when logged in to different browsers with same credentials.
Reference	https://owasp.org/www-project-web-security-testing-guide/latest/4-Web_Application_Security_Testing/06-Session_Management_Testing/11-Testing_for_Concurrent_Sessions

Proof of Concept:

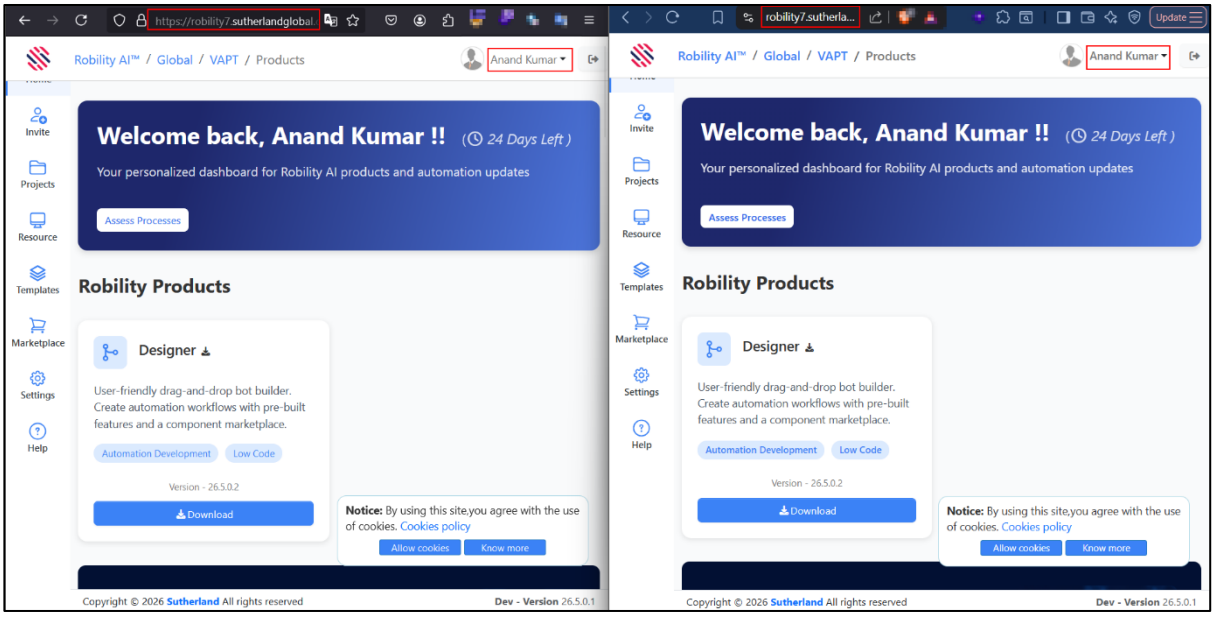


FIGURE 14: CONCURRENT SESSIONS ENABLED

9. USER ENUMERATION

Application / URL	https://robility7.sutherlandglobal.com/Robility
Revalidation Status	CLOSED
Severity	Low
CVSS Score	2.1
CVSS String	CVSS:4.0/AV:N/AC:H/AT:N/PR:H/UI:N/VC:L/VI:L/VA:L/SC:N/SI:N/SA:N
Description	During assessment, it was observed that application is vulnerable to user enumeration while trying to loin with PIN.
Impact	Attackers often use automated tools to submit lists of potential usernames. If a server returns an error such as "User does not exist," they know to eliminate that username. Conversely, a generic error like "Invalid username or password" for both existing and non-existing users helps prevent this.
Remediation	It is recommended to: • Use identical, ambiguous error messages for both failed usernames and passwords. • Implement rate limiting and account lockout mechanisms to stop automated probing.
Reference	https://owasp.org/www-project-web-security-testing-guide/latest/4-Web_Application_Security_Testing/03-Identity_Management_Testing/04-Testing_for_Account_Enumeration_and_Guessable_User_Account

Proof of Concept:

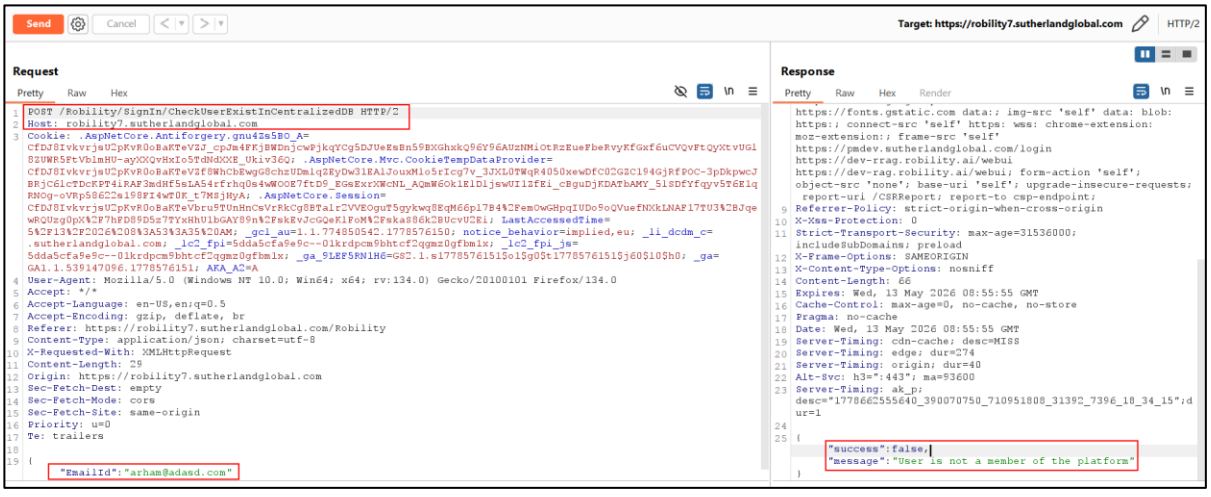


FIGURE 15: RESPONSE SHOWING THAT THE USER DOESN'T EXIST

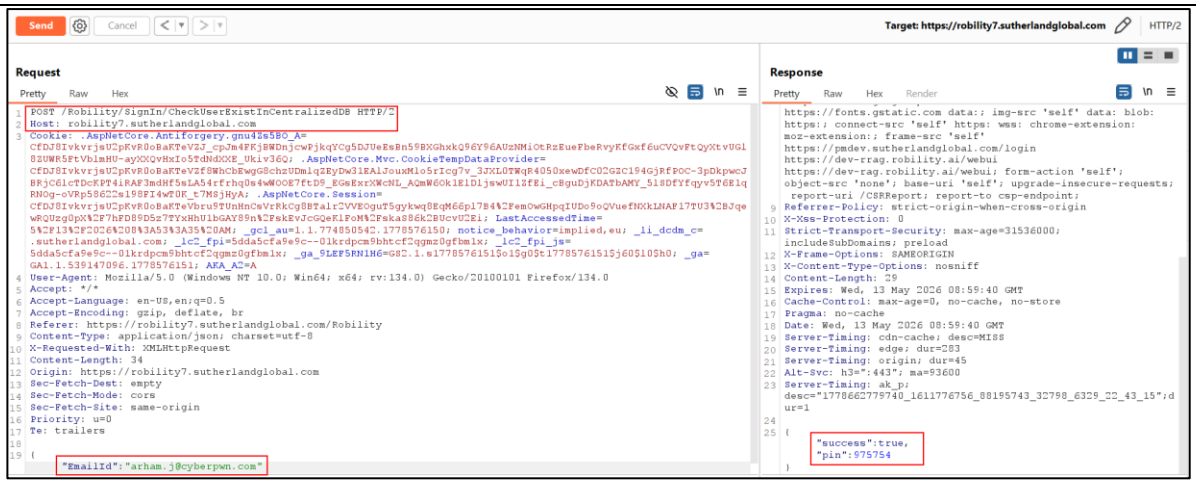


FIGURE 16: RESPONSE FOR A LEGITIMATE USER

10. VULNERABLE COMPONENT IN USE	
Application / URL	https://robility7.sutherlandglobal.com/Robility/js/Chart/Chart.min.js
Revalidation Status	CLOSED
Severity	Low
CVSS Score	2.1
CVSS String	CVSS:4.0/AV:N/AC:H/AT:N/PR:H/UI:N/VC:L/VI:L/VA:L/SC:N/SI:N/SA:N
Description	During assessment, it was observed that application has been using vulnerable version of Chart.js 2.7.2.
Impact	**CVE-2020-7746:** This affects the package chart.js before 2.9.4. The options parameter is not properly sanitized when it is processed. When the options are processed, the existing options (or the defaults options) are deeply merged with provided options. However, during this operation, the keys of the object being set are not checked, leading to a prototype pollution.
Remediation	It is recommended to upgrade Chart.js to v2.9.4 or later.
Reference	https://nvd.nist.gov/vuln/detail/CVE-2020-7746

Proof of Concept:

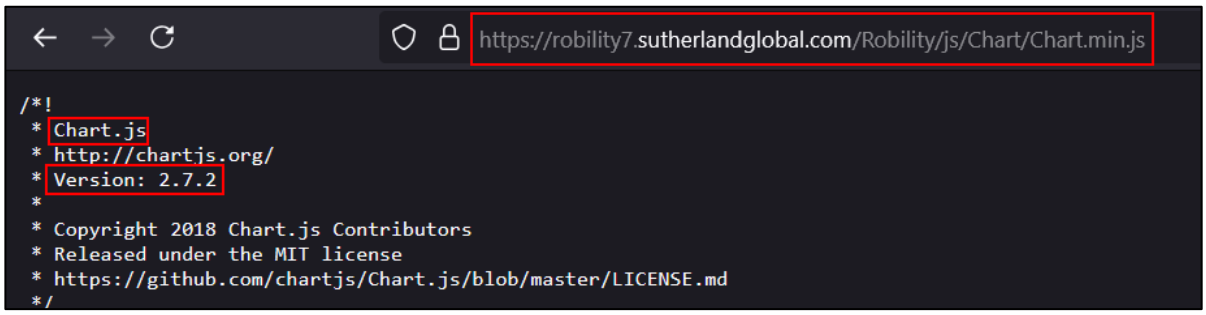


FIGURE 17: VULNERABLE VERSION OF CHART.JS IN USE

CONCLUSION

Our security assessment revealed total of 10 vulnerabilities in the **Robility7** application - Web application issues are related to Authentication Bypass, HTML Injection, Password in Plain Text, Rate Limiting is Not Implemented, Misconfigures CSP, Hardcoded API Key, Business Logic Flaw, User Enumeration, Vulnerable Components in use and Concurrent Login Enabled.

ANNEXURE A – CHANGES TO ENVIRONMENT

We have created records and uploaded files into the application for testing purpose.

ANNEXURE B – TOOLS USED

Tools	Description
Burp Suite Professional	Burp Suite is a Java based Web Penetration Testing framework. It has become an industry standard suite of tools used by information security professionals. Burp Suite helps you identify vulnerabilities and verify attack vectors that are affecting web applications
Nikto	Nikto is an Open Source (GPL) web server scanner which performs comprehensive tests against web servers for multiple items, including over 6400 potentially dangerous files/CGIs, checks for outdated versions of over 1200 servers, and version specific problems on over 270 servers.
Nmap	Nmap is a powerful tool for discovering information about machines on a network or the Internet. It allows you to probe a machine with packets to detect everything from running services and open ports to the operating system and software versions. Nmap can go a long way in helping to secure your own network. It can also help you to ensure that your servers are properly configured and don't have any open and unsecured ports.
Qualys SSL labs	This free online service performs a deep analysis of the configuration of any SSL web server on the public Internet.
Kali Linux	One of the most complete penetration testing Linux distributions available. Includes many of the more popular free pen-testing tools but is based on Ubuntu so it's also easily expandable.
Custom Exploit Scripts	

CYBERPWN
EMPOWERING CYBER RESILIENCE

ANNEXURE C - LIST OF VAPT TESTS PERFORMED

Test Cases
Error Handling and Information Disclosure
Input Validation and Manipulation
HTML Injection
Testing HTTP Methods
Client-Side Security
Host Header Injection
Unvalidated Redirects and Forwards
Test For No-Rate Limiting
Reflected Cross Site Scripting
Third-Party Integration
Server-Side Template Injection
Clickjacking
Command Injection
CORS - Misconfigurations & Bypass
File Inclusion/Path traversal
IDOR
Open Redirect
Weak Cryptography Testing
Regular expression Denial of Service
Test for forced browsing
Browser Cache Weakness
Business Logic
SPF and DMARC
SQL Injection
Privilege Escalation
Security Headers
Brute Force Attack
Weak Password Policy
Session Management
Concurrent Login Allowed
Third party and vulnerable components

WEB APPLICATION PENETRATION TESTING

A REPORT BY CYBERPWN
EMPOWERING CYBER RESILIENCE

