Application Security Report

Report Release Date	26-April-2025
Type of Audit	Application Security
Type of Audit Report	First Audit Report
Period	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

Document Control

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

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

Document Distribution List			
Name	**Organization**	**Designation**	**Email Id**
Mr. Redacted	Redacted	Redacted	Redacted@gmail.com

Contents

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.

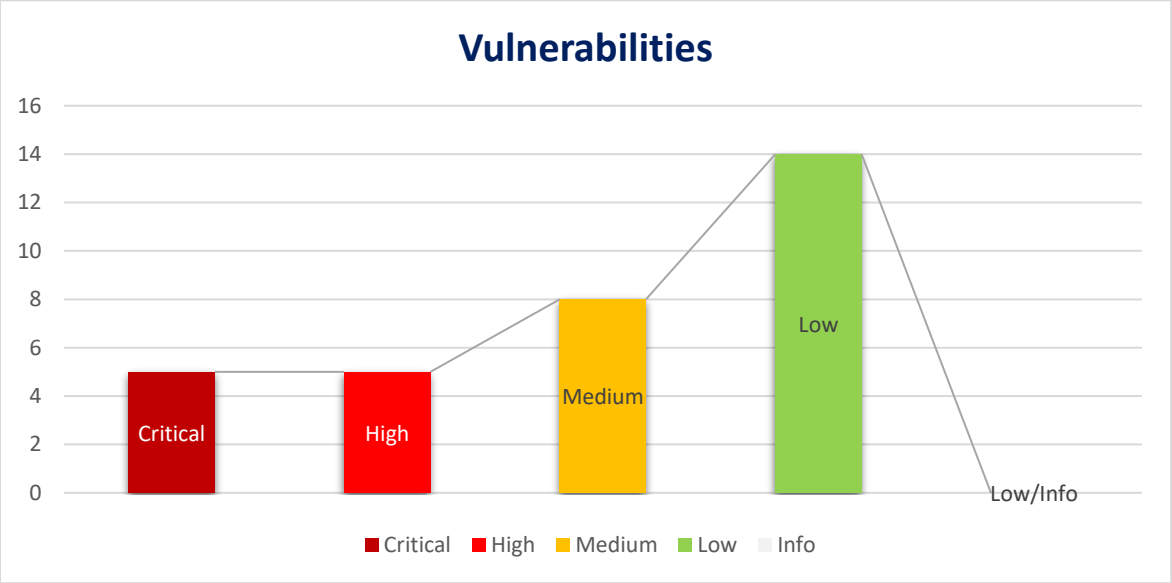
Tools/ Software used

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.



DarkNext
Cybersecurity

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

Vulnerability	Forced Browsing
Risk	**Critical**
Description	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
Solution	Avoid common file names and disable directory listing on the webserver. Verify user authentication and enforce access controls before granting access to sensitive functions.
CWE	CWE-425
Ref	N/A
Screenshot	Please refer to the screenshots section at the end of this document.

Vulnerability	Sensitive data exposed
Risk	**Critical**
Description	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.
Solution	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.
CWE	CWE-540
Ref	N/A
Screenshot	Please refer to the screenshots section at the end of this document.

Vulnerability	Potential Privilege Escalation through DLL Hijacking
Risk	**Critical**
Description	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.
Solution	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). Sign your DLLs and verify their signatures at runtime before loading them. This ensures only trusted, unaltered DLLs are used. Place your DLLs in secure, access-controlled directories (like C:\Program Files) and ensure the folder is not writable by non-admin users. Avoid LoadLibrary when not necessary. Limit dynamically loading DLLs at runtime unless essential. Validate and sanitize all inputs used in DLL loading logic.
CVE	CVE-2019-0859
Ref	N/A
Screenshot	Please refer to the screenshots section at the end of this document.

Vulnerability	Cleartext Credentials
Risk	**Critical**
Description	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.
Solution	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
CWE	CWE-319
Ref	N/A
Screenshot	Please refer to the screenshots section at the end of this document.

Vulnerability	Hardcoded Credentials
Risk	**Critical**
Description	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.
Solution	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.
CWE	N/A
Ref	N/A
Screenshot	Please refer to the screenshots section at the end of this document.

Vulnerability	Authentication Bypass via Reverse Engineering
Risk	**High**
Description	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.
Solution	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.
CWE	N/A
Ref	N/A
Screenshot	Please refer to the screenshots section at the end of this document.

Vulnerability	Unencrypted Credentials Stored in Database
Risk	**High**
Description	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.
Solution	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.
CWE	N/A
Ref	N/A
Screenshot	Please refer to the screenshots section at the end of this document.

Vulnerability	VoIP caller ID spoofing
Risk	**High**
Description	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
Solution	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.
CWE	N/A
Ref	N/A
Screenshot	Please refer to the screenshots section at the end of this document.

Vulnerability	VoIP Flood Attack
Risk	**High**
Description	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.
Solution	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.
CWE	N/A
Ref	N/A
Screenshot	Please refer to the screenshots section at the end of this document.

Vulnerability	SIP User Enumeration
Risk	**High**
Description	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.
Solution	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.
CWE	N/A
Ref	N/A
Screenshot	Please refer to the screenshots section at the end of this document.

Vulnerability	Rate Limiting not implemented
Risk	**Medium**
Description	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.
Solution	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.
CWE	CWE-307
Ref	N/A
Screenshot	Please refer to the screenshots section at the end of this document.

Vulnerability	Internal Path Disclosure
Risk	**Medium**
Description	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.
Solution	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.
CWE	CWE-200
Ref	https://cheatsheetseries.owasp.org/cheatsheets/Error_Handling_Cheat_Sheet.html
Screenshot	Please refer to the screenshots section at the end of this document.

Vulnerability	Session Fixation
Risk	**Medium**
Description	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.
Solution	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.
CWE	N/A
Ref	N/A
Screenshot	Please refer to the screenshots section at the end of this document.

Vulnerability	SIP Server Configured with POP3
Risk	**Medium**
Description	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.
Solution	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.
CWE	N/A
Ref	N/A
Screenshot	Please refer to the screenshots section at the end of this document.

Vulnerability	Brute Force Attack Against SIP Credentials
Risk	**Medium**
Description	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.
Solution	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.
CWE	N/A
Ref	N/A
Screenshot	Please refer to the screenshots section at the end of this document.

Vulnerability	Cleartext Transmission of SQL queries/Possible SQL injection
Risk	**Medium**
Description	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.
Solution	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.
CWE	N/A
Ref	N/A
Screenshot	Please refer to the screenshots section at the end of this document.

Vulnerability	Application is vulnerable to CSRF attack
Risk	**Medium**
Description	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.
Solution	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.
CWE	N/A
Ref	N/A

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

Vulnerability	Weak Password policy
Risk	**Medium**
Description	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.
Solution	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).
CWE	N/A
Ref	N/A
Screenshot	Please refer to the screenshots section at the end of this document.

Vulnerability	Application is vulnerable to Directory Listing Attack
Risk	**Low**
Description	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.
Solution	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.
CWE	CWE-548
Ref	N/A
Screenshot	Please refer to the screenshots section at the end of this document.

Vulnerability	Application displays web server banner
Risk	**Low**
Description	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.
Solution	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.
CWE	CWE-200
Ref	N/A
Screenshot	Please refer to the screenshots section at the end of this document.

Vulnerability	Application is vulnerable to simultaneous login
Risk	**Low**
Description	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.
Solution	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.
CWE	CWE-1018
Ref	N/A
Screenshot	Please refer to the screenshots section at the end of this document.

Vulnerability	Path attribute not set in session cookie
Risk	**Low**
Description	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.
Solution	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.
CWE	CWE-614
Ref	N/A
Screenshot	Please refer to the screenshots section at the end of this document.

Vulnerability	Default Passwords
Risk	**Low**
Description	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.
Solution	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.
CWE	CWE-200
Ref	N/A
Screenshot	Please refer to the screenshots section at the end of this document.

Vulnerability	Exposed web server logs
Risk	**Low**
Description	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.
Solution	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.
CWE	N/A
Ref	N/A
Screenshot	Please refer to the screenshots section at the end of this document.

Vulnerability	Framework Version Disclosure
Risk	**Low**
Description	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.
Solution	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.
CWE	CWE-1328
Ref	N/A
Screenshot	Please refer to the screenshots section at the end of this document.

Vulnerability	Improper error handling
Risk	**Low**
Description	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.
Solution	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.
CWE	CWE-209
Ref	N/A
Screenshot	Please refer to the screenshots section at the end of this document.

Vulnerability	OPTIONS method enabled
Risk	**Low**
Description	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.
Solution	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.
CWE	CWE-650
Ref	N/A
Screenshot	Please refer to the screenshots section at the end of this document.

Vulnerability	Secure flag is not set in session cookie
Risk	**Low**
Description	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.
Solution	Set the Secure flag in session cookies to ensure they are transmitted only over HTTPS, preventing interception over unencrypted connections.
CWE	CWE-613
Ref	https://cheatsheetseries.owasp.org/cheatsheets/Session_Management_Cheat_Sheet.html
Screenshot	Please refer to the screenshots section at the end of this document.

Vulnerability	Security Headers are missing
Risk	**Low**
Description	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. 1. X-Frame options Header is a security header to avoid UI redressing attack such as Clickjacking. 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. 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. 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. 5. HTTP Strict Transport Security HTTP Header is a mechanism that forces browsers to use a secure web connection.
Solution	Kindly implement all Security headers with all attributes and parameters.
CWE	CWE-1018

Ref	Please refer the following links: 1. https://geekflare.com/http-header-implementation/ 2. https://pentest-tools.com/blog/essential-http-security-headers/ 3. https://securityheaders.com/
Screenshot	Please refer to the screenshots section at the end of this document.

Vulnerability	Default web page found on the application server
Risk	**Low**
Description	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.
Solution	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.
CWE	CWE-200
Ref	N/A
Screenshot	Please refer to the screenshots section at the end of this document.

Vulnerability	Application is vulnerable to ClickJacking attack
Risk	**Low**
Description	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.
Solution	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.
CWE	CWE-1021
Ref	N/A
Screenshot	Please refer to the screenshots section at the end of this document.

Vulnerability	SIP Server Fingerprinting Enabled
Risk	**Low**
Description	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.
Solution	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.
CWE	N/A
Ref	N/A
Screenshot	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
Critical	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.	• Loss of (confidentiality / integrity / availability) is likely to have a catastrophic effect on the organisation's operations and reputation. • Specialized access conditions or extenuating circumstances do not exist. Very little knowledge or skill is required to exploit
High	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	• Loss of (confdentiality / integrity / availability) is likely to have a catastrophic effect on the organisation • Specialized access conditions or extenuating circumstances do not exist. Very little knowledge or skill is required to exploit
Medium	Important findings that are to be resolved by management.	• 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 • Authentication is required to exploit the vulnerability • The access conditions are somewhat specialized. Some preconditions must be satisfied to exploit

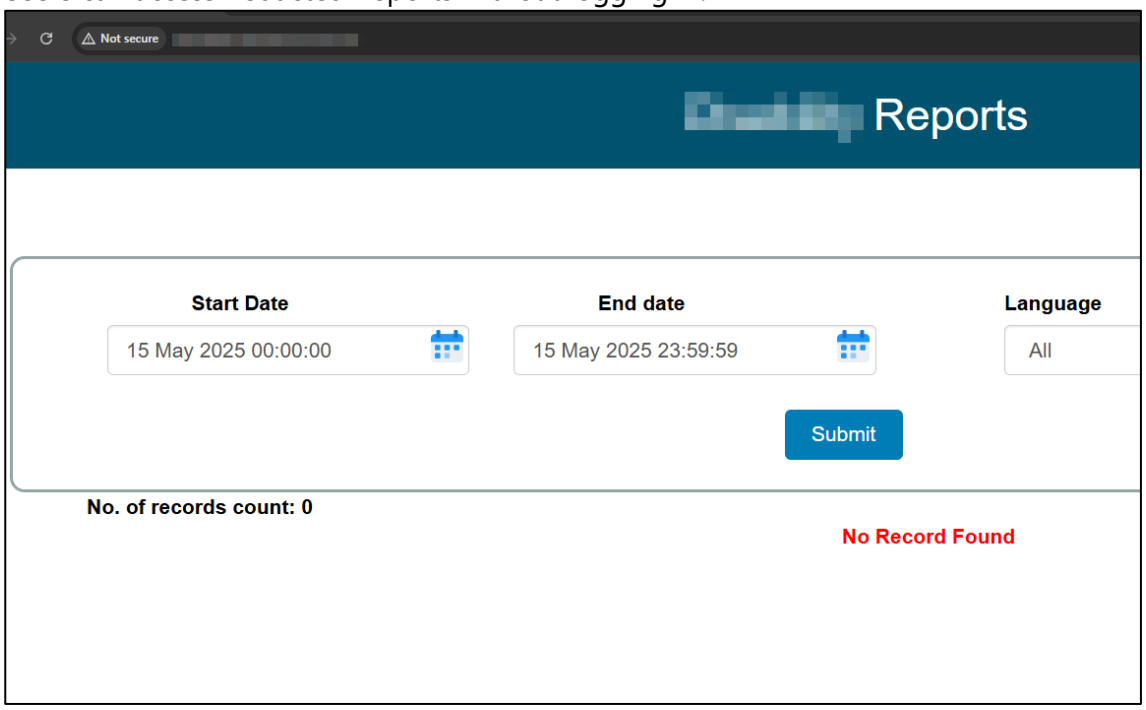
Low	Findings that identify an area for review with established services and good practice	• There is reduced performance or interruptions in resource availability • There is little to no impact to the integrity of the system • There is informational disclosure
Info	Findings that are limited in affect but are worthy of being noted for review- for example future proofing.	• Information for department management • Very limited risk - something that could be utilised with the relevant skillsets and resources to gather information

Appendix B: Screenshot

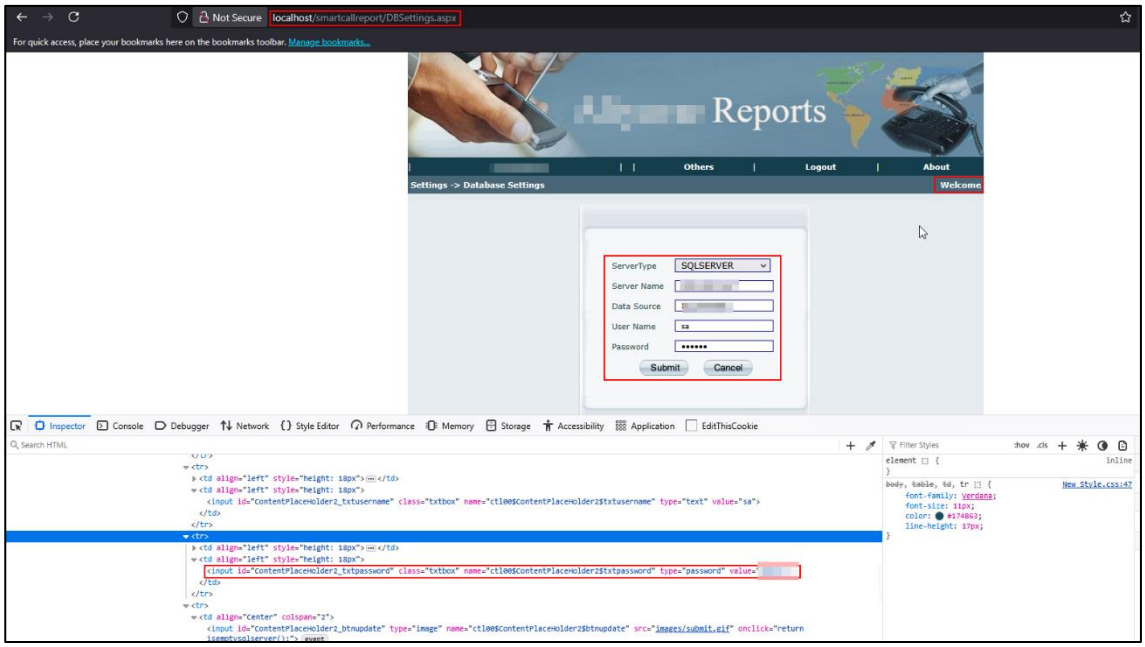
1. Forced Browsing

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.

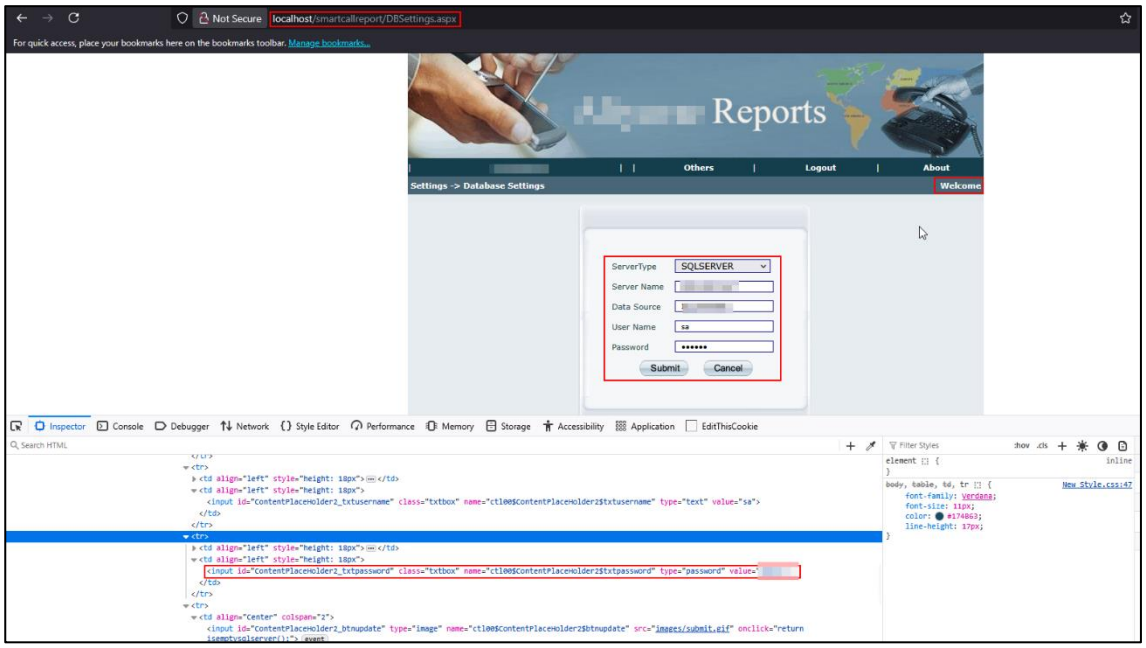


The Database Settings page is accessible without authentication, allowing unauthorized individuals to view database credentials.



2. Sensitive data exposed

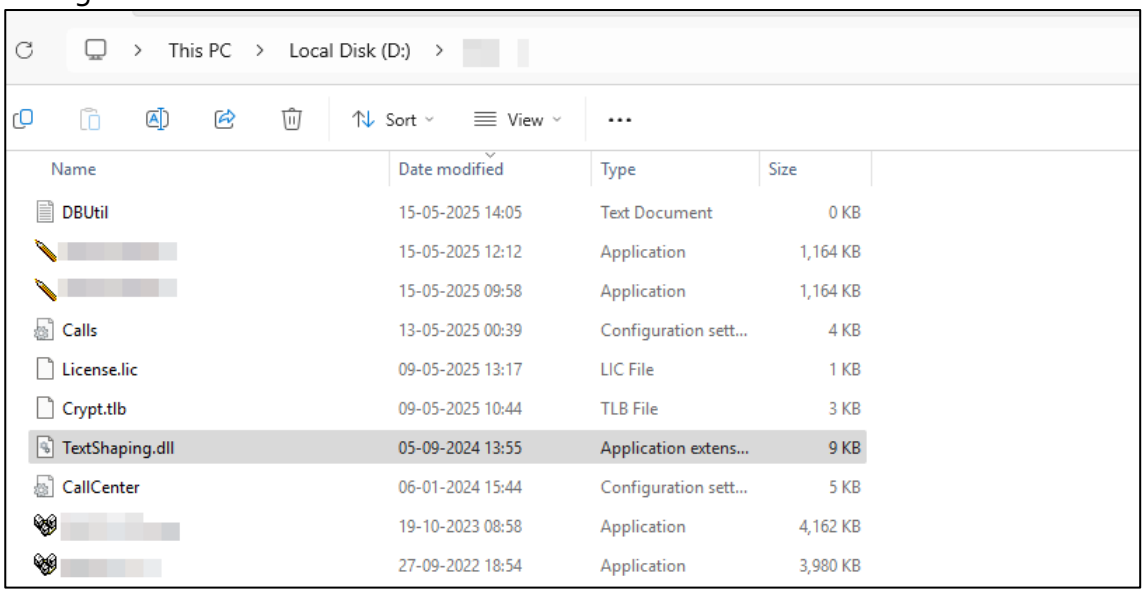
Passwords are embedded in the HTML source code in plaintext, making them easily accessible to anyone viewing the page. This poses a serious security risk, as attackers can retrieve credentials without authentication.



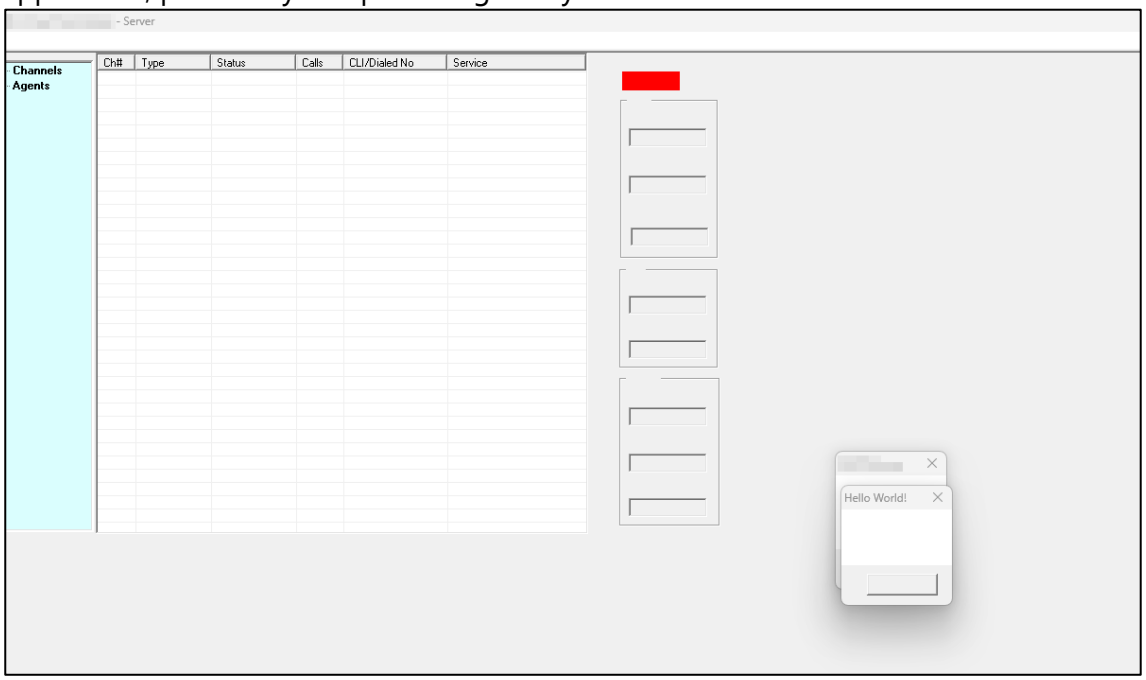
3. Potential Privilege Escalation through DLL Hijacking

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

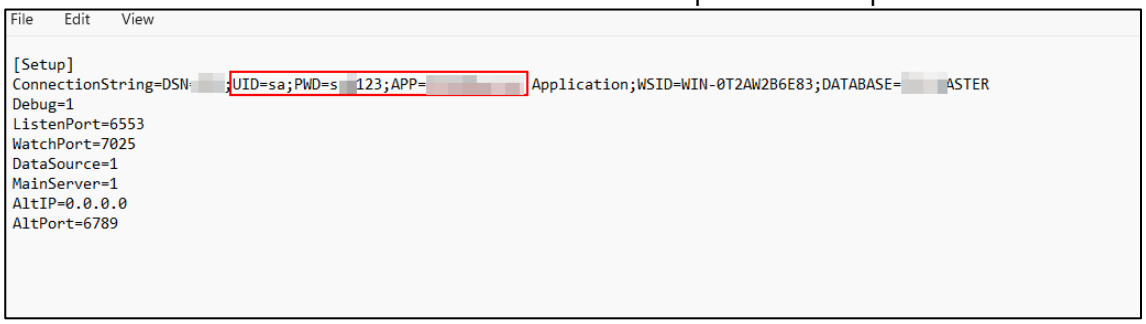


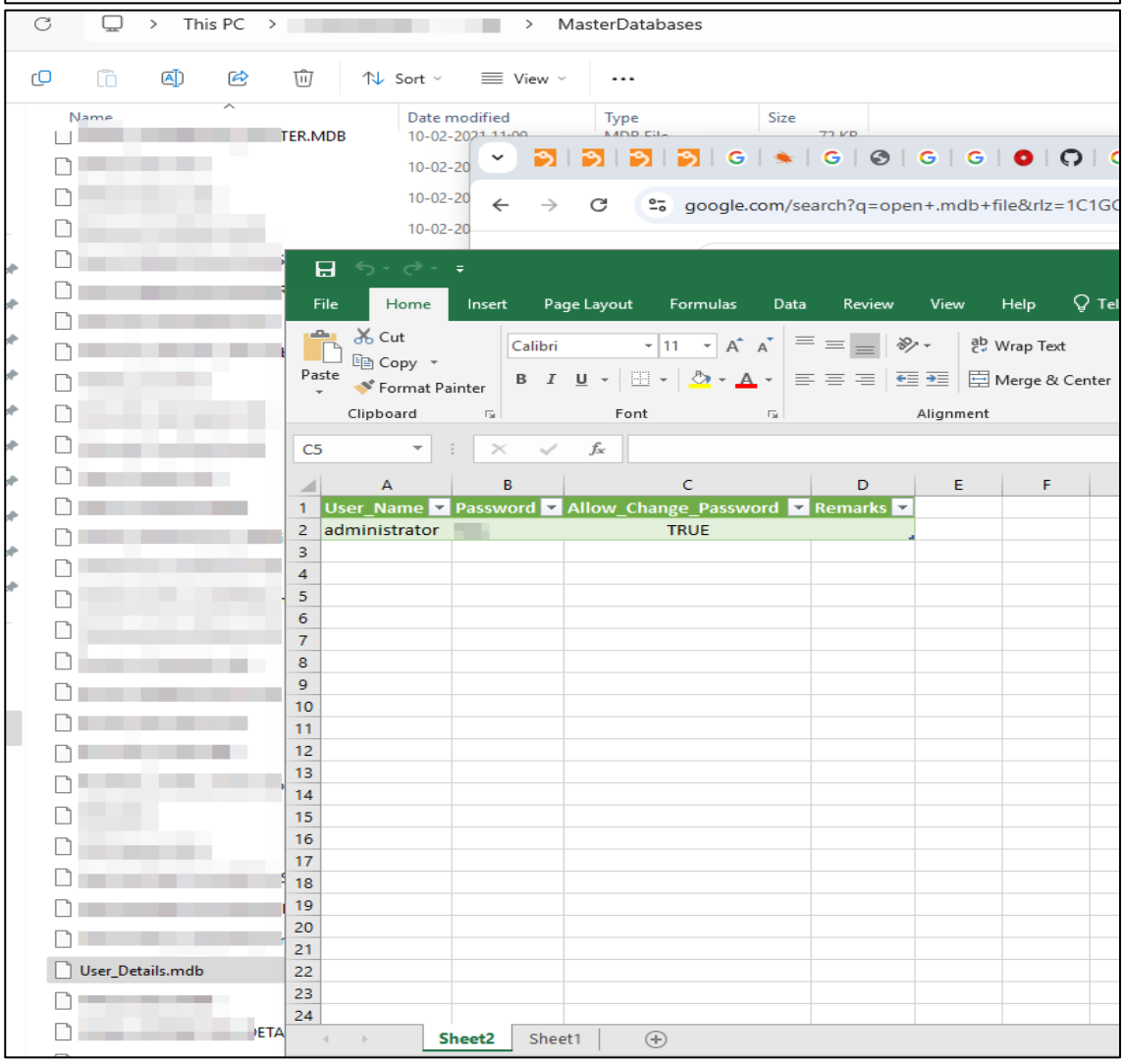
When the application starts, it unknowingly loads the attacker's DLL due to improper DLL search order or missing path validation. This malicious DLL then runs with the same privileges as the application, potentially compromising the system.

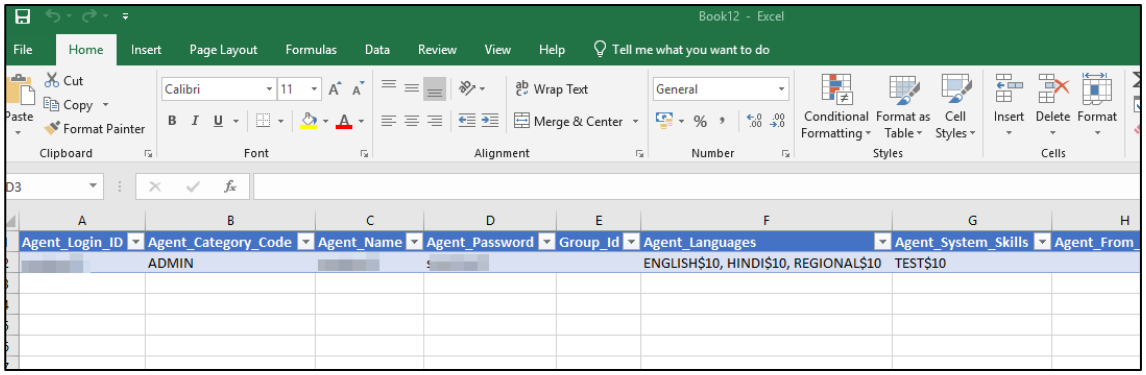


4. Cleartext Credentials

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.



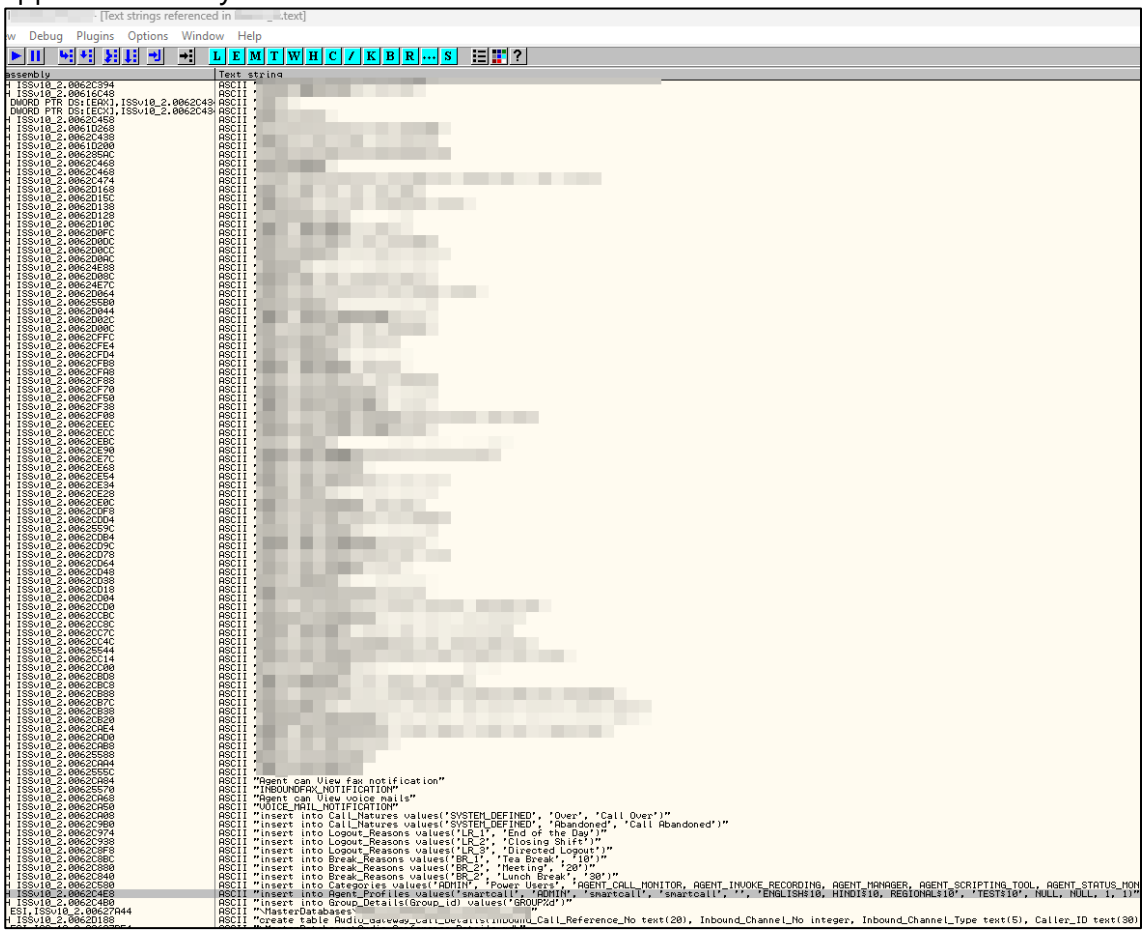


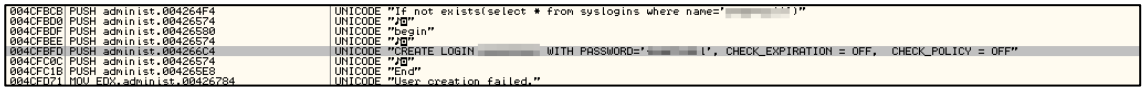

Book12 - Excel
File
Home
Insert
Page Layout
Formulas
Data
Review
View
Help
Tell me what you want to do
Cut
Copy
Format Painter
Paste
Clipboard
Calibri
11
Font
Wrap Text
Merge & Center
Alignment
General
Number
Conditional Formatting
Format as Table
Cell Styles
Styles
Insert
Delete
Format
Cells
D3
A
B
C
D
E
F
G
H
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

5. Hardcoded Credentials

The application contains hardcoded credentials (e.g., usernames, passwords, API keys) embedded in the source code or binary. This exposes sensitive information to attackers who reverse engineer the application or inspect the code, leading to unauthorized access.

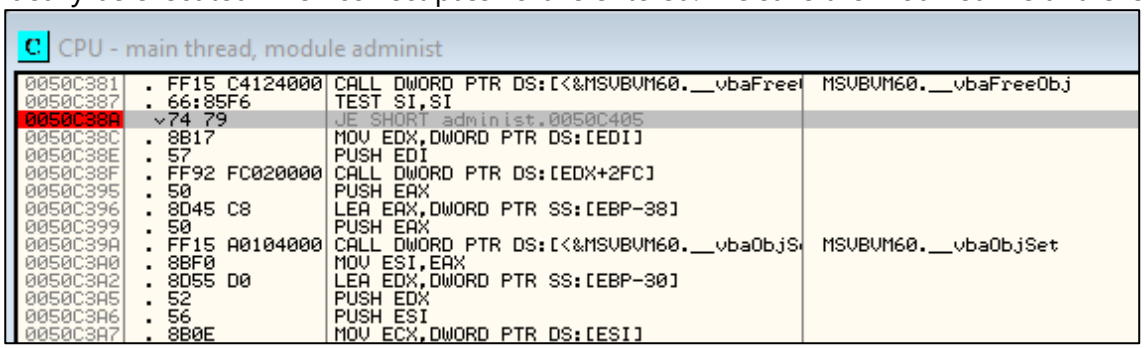
The user loads the application using Redacted and is able to view all runtime operations, including hardcoded credentials. This exposes sensitive information stored insecurely within the application binary.



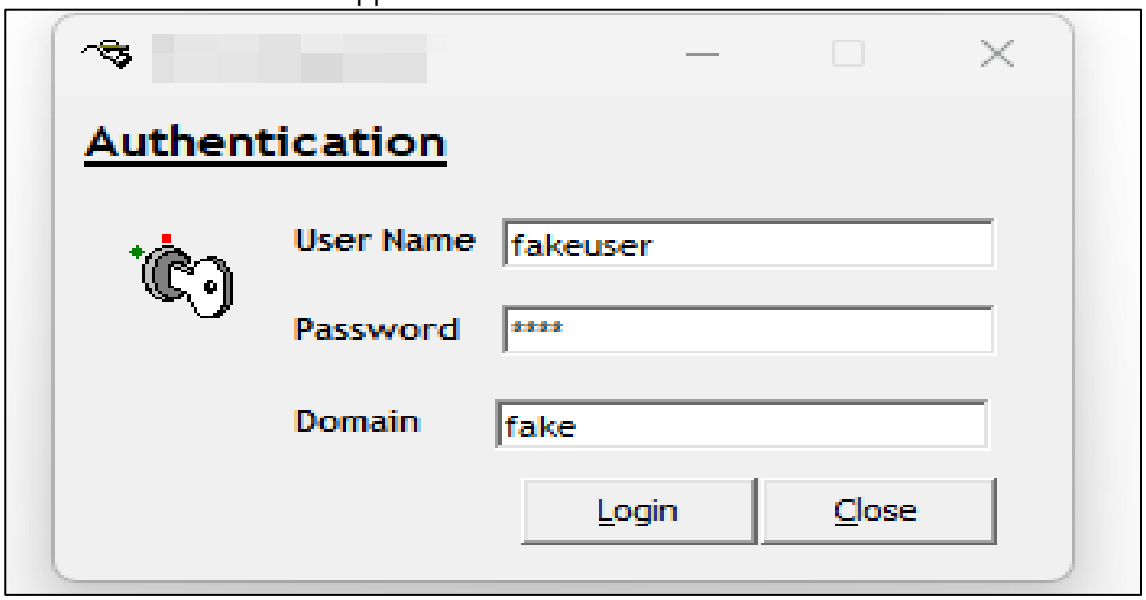


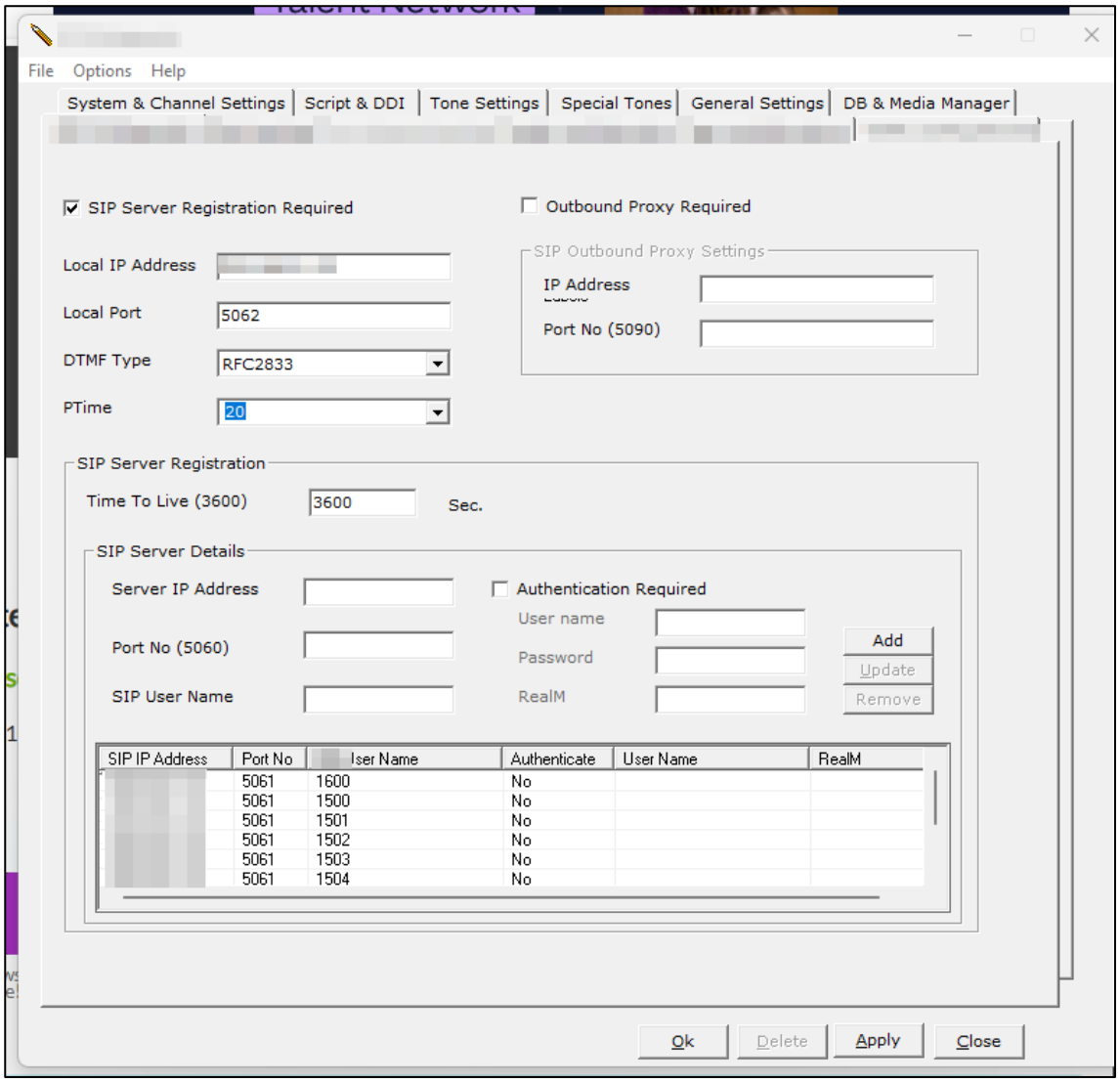
6. Authentication Bypass via Reverse Engineering

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.



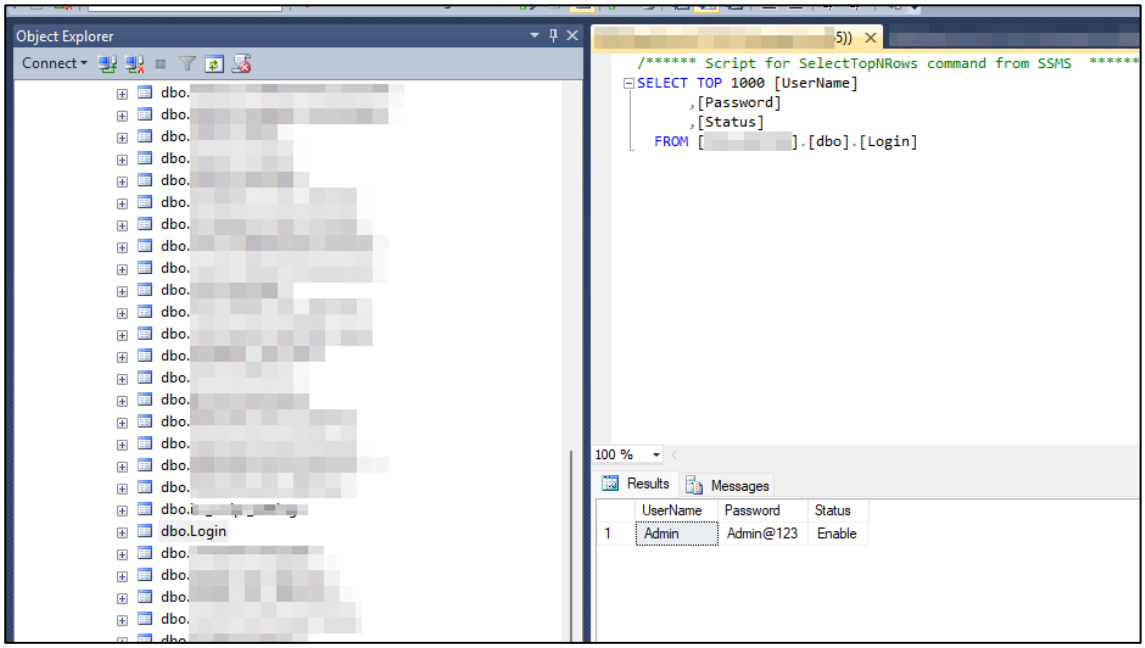
Below screenshots shows that we enter fake values instead of username, password and domain but are able to execute Redacted Application.




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
5062
DTMF Type
RFC2833
PTime
20
SIP Outbound Proxy Settings
IP Address
Port No (5090)
SIP Server Registration
Time To Live (3600)
3600
Sec.
SIP Server Details
Server IP Address
Port No (5060)
SIP User Name
Authentication Required
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

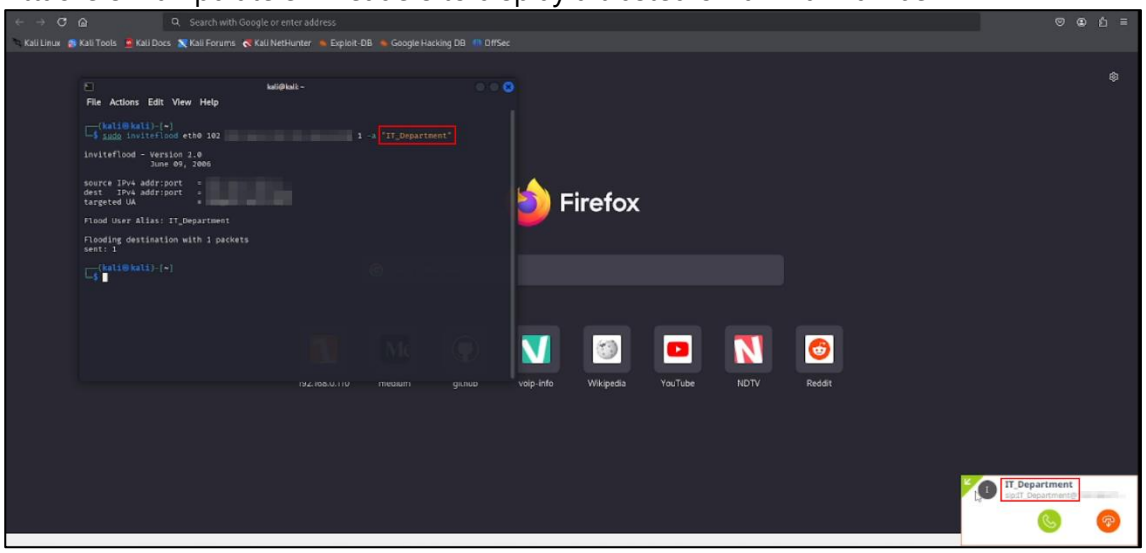
User credentials are stored in the database in plaintext without encryption or hashing. This exposes sensitive data to attackers in the event of database compromise, increasing the risk of credential theft and unauthorized access.



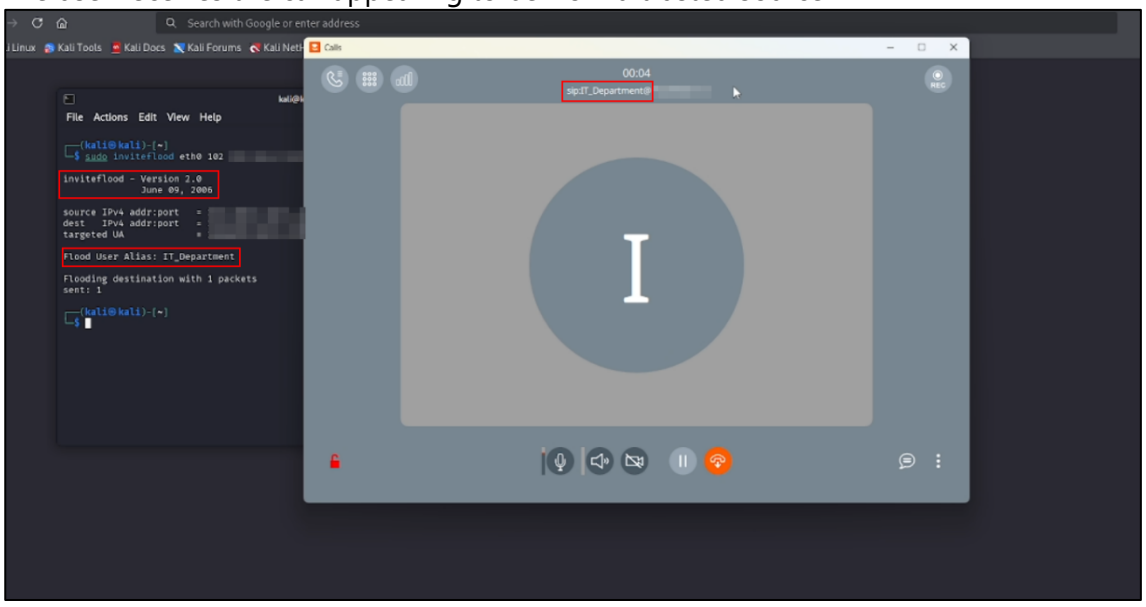
8. VoIP caller ID spoofing

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

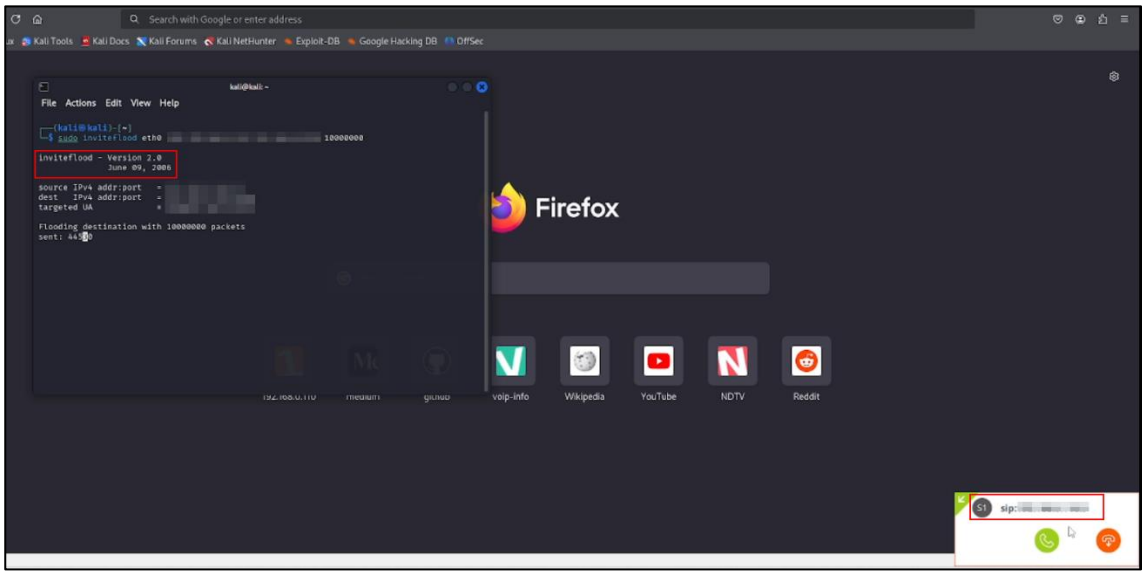


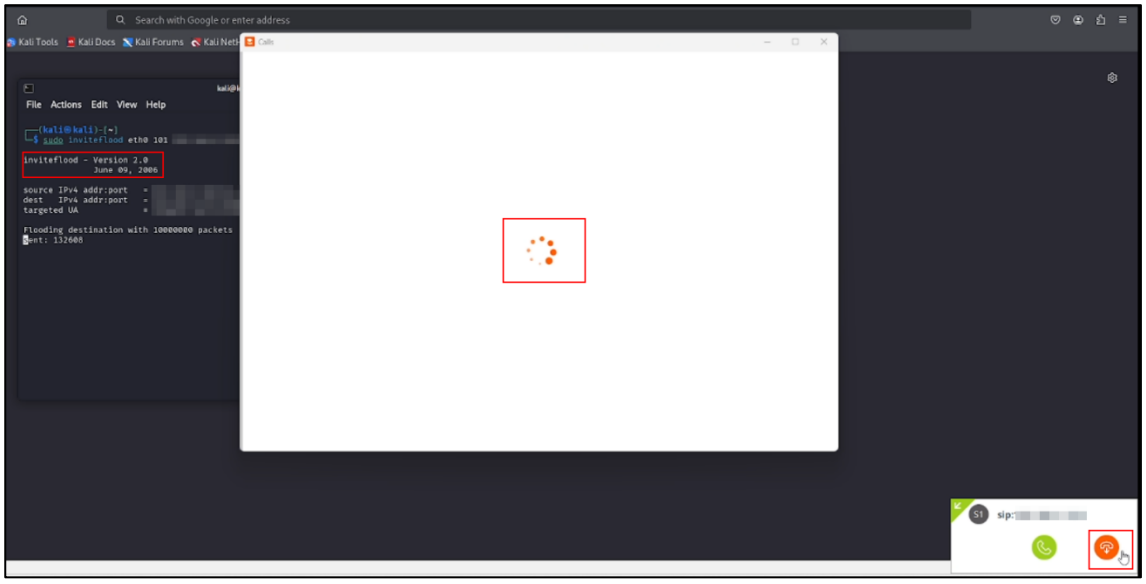
The user receives the call appearing to be from a trusted source



9. VoIP Flood Attack

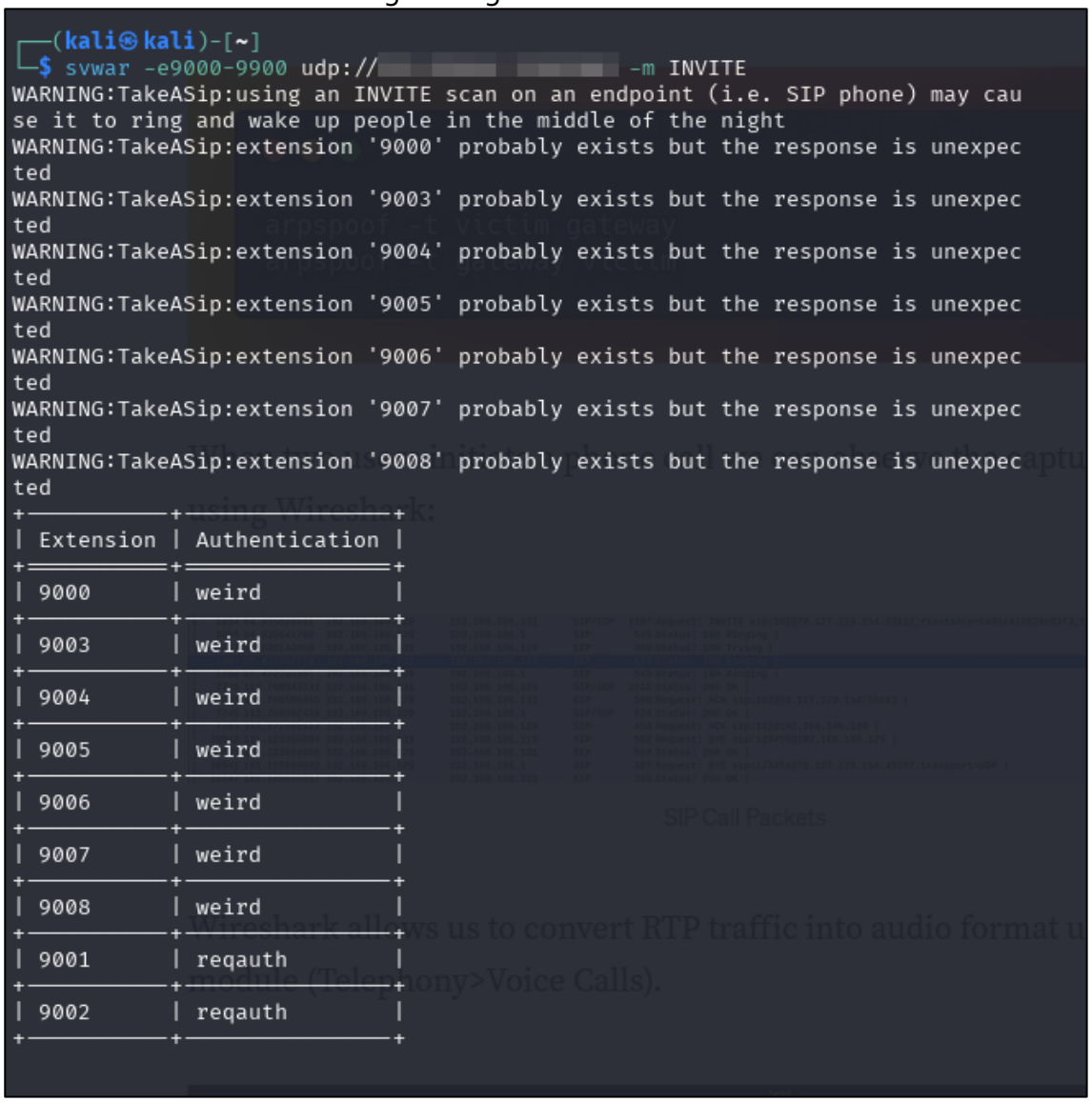
The VoIP server is susceptible to flood attacks such as SIP INVITE or REGISTER floods, where a large volume of bogus requests overwhelms the server. This can lead to degraded performance or denial of service, disrupting legitimate call traffic.





10.SIP User Enumeration

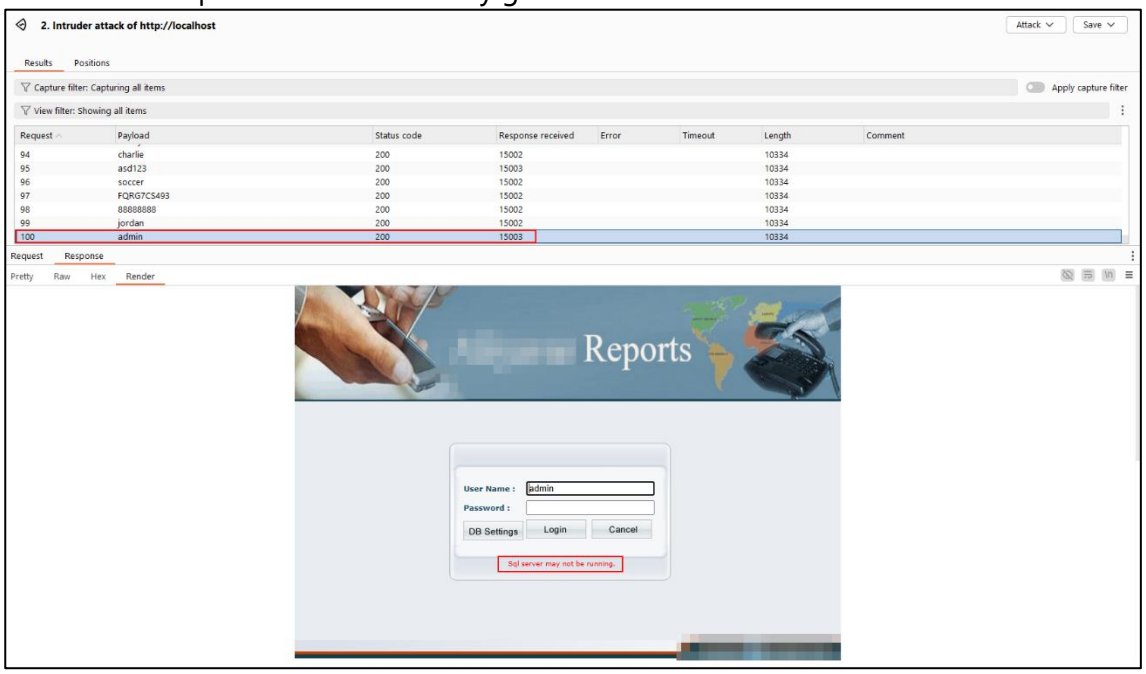
The SIP server responds differently to valid and invalid usernames during authentication attempts, allowing an attacker to enumerate valid SIP user accounts. This can be used to target accounts for brute-force attacks or social engineering



11. Rate Limiting not implemented

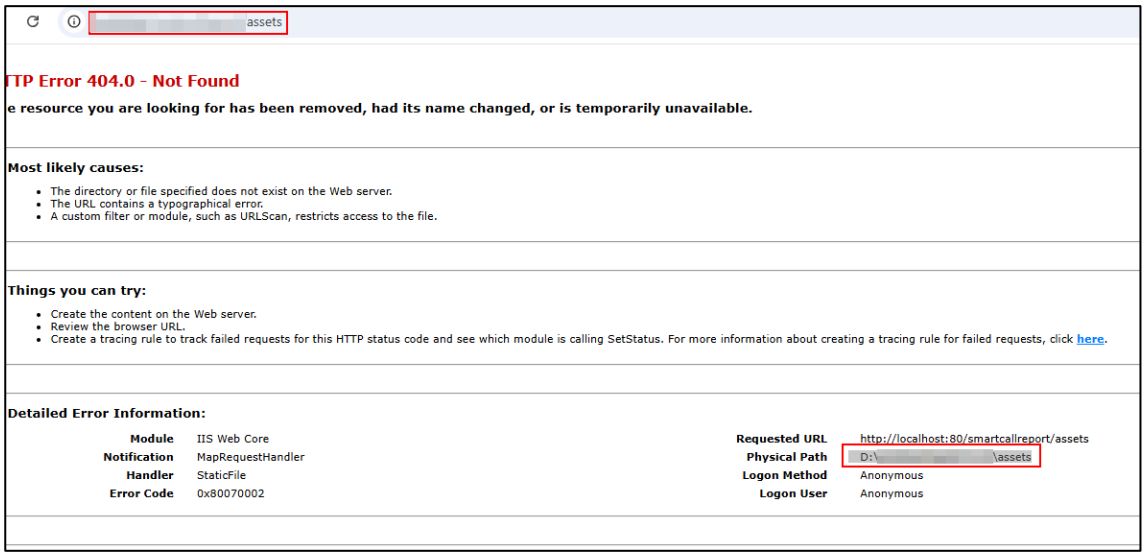
The application does not enforce rate limiting on critical endpoints such as login, password reset, or API requests. This allows attackers to perform brute-force or enumeration attacks without restriction, increasing the risk of unauthorized access.

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.



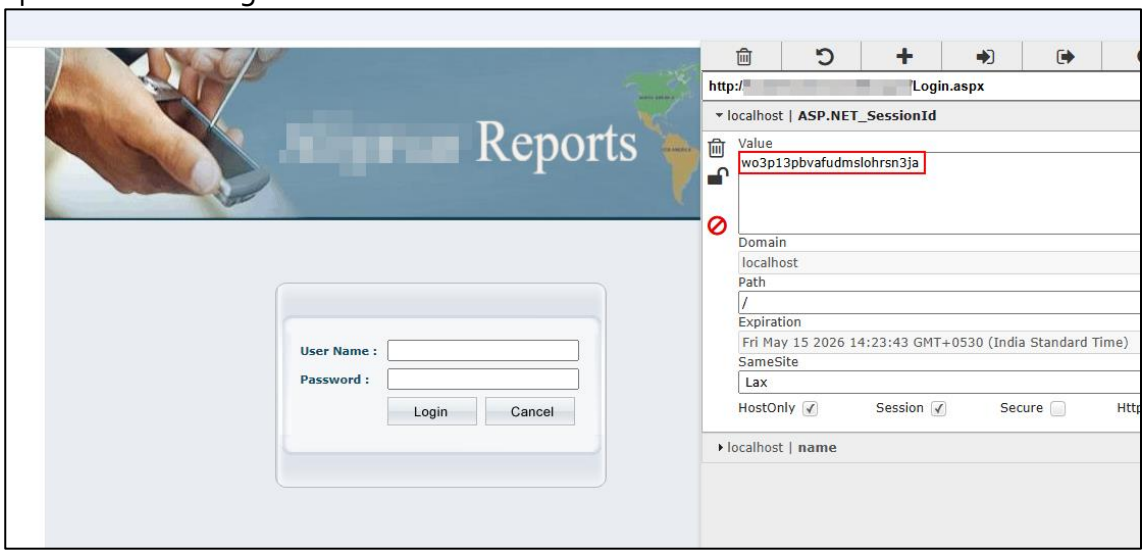
12.Internal Path disclosure

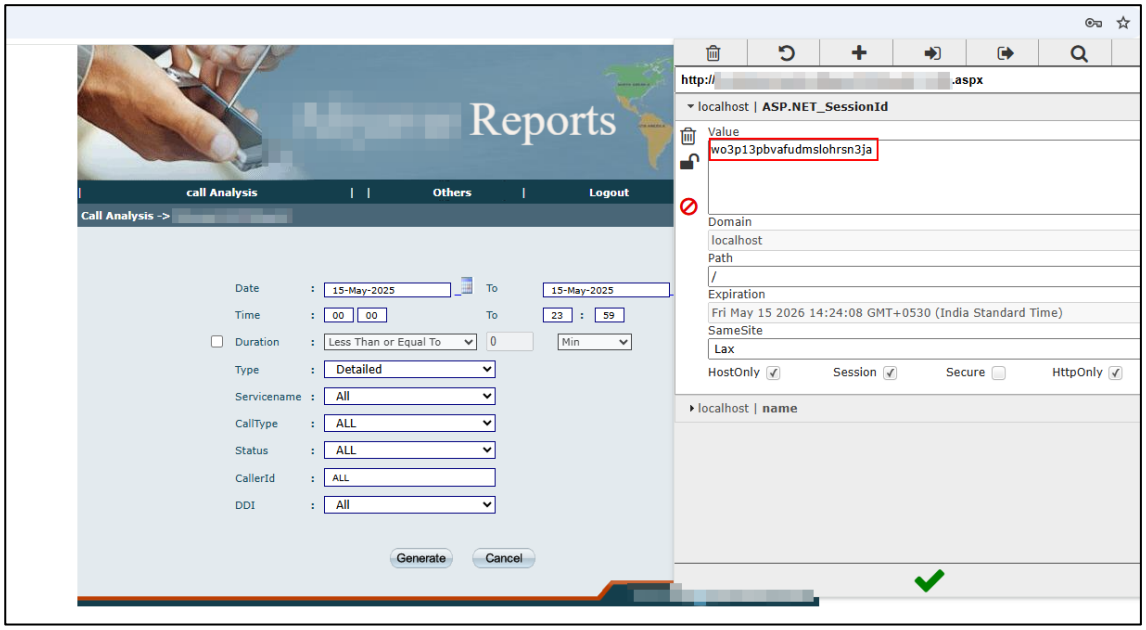
The application reveals internal server file paths in error messages or responses. This information disclosure can aid attackers in crafting targeted attacks by providing insights into the server's directory structure and technology stack.



13. Session Fixation

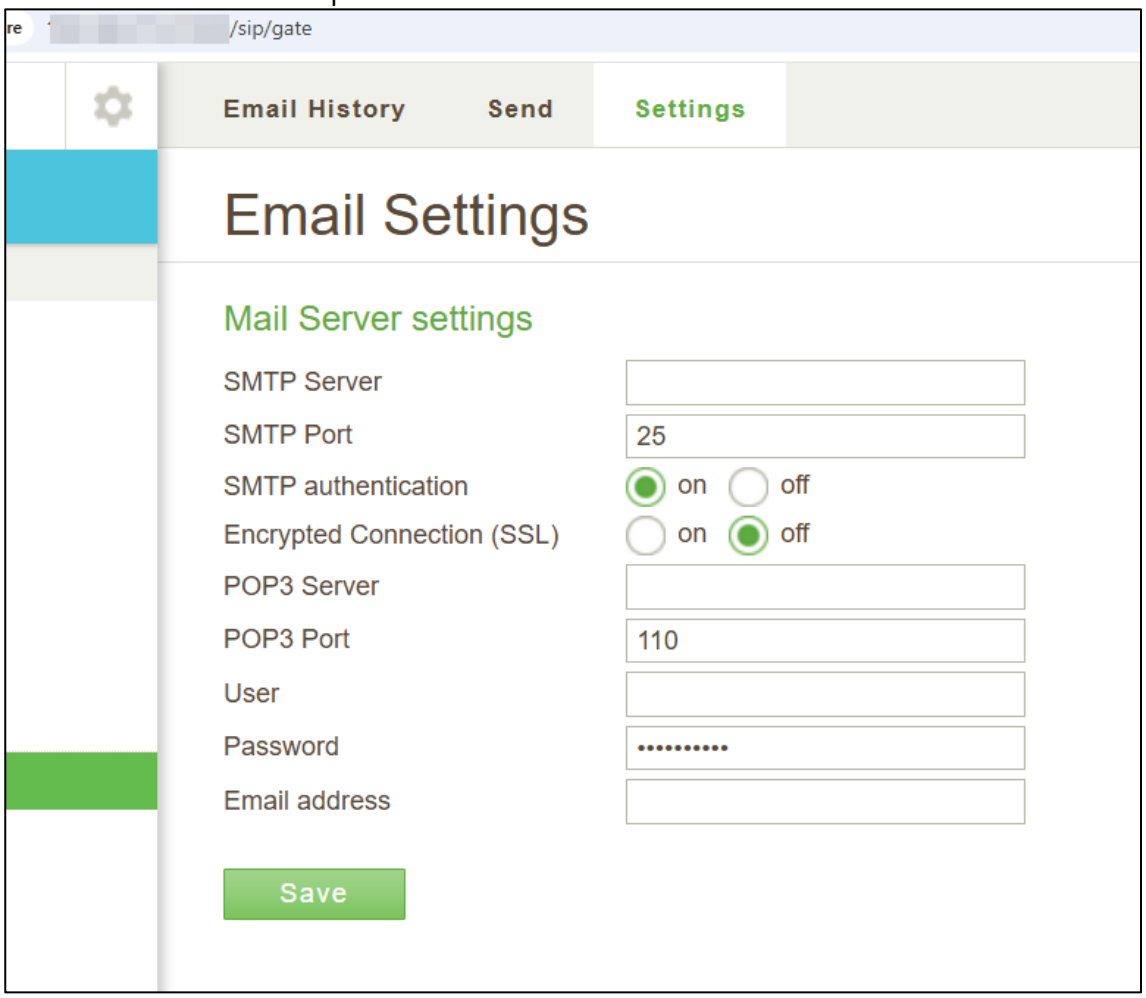
The application allows an attacker to fixate a user's session ID before login, enabling the attacker to hijack the user's session after authentication. This occurs when session IDs are not regenerated upon successful login.





14. SIP Server Configured with POP3

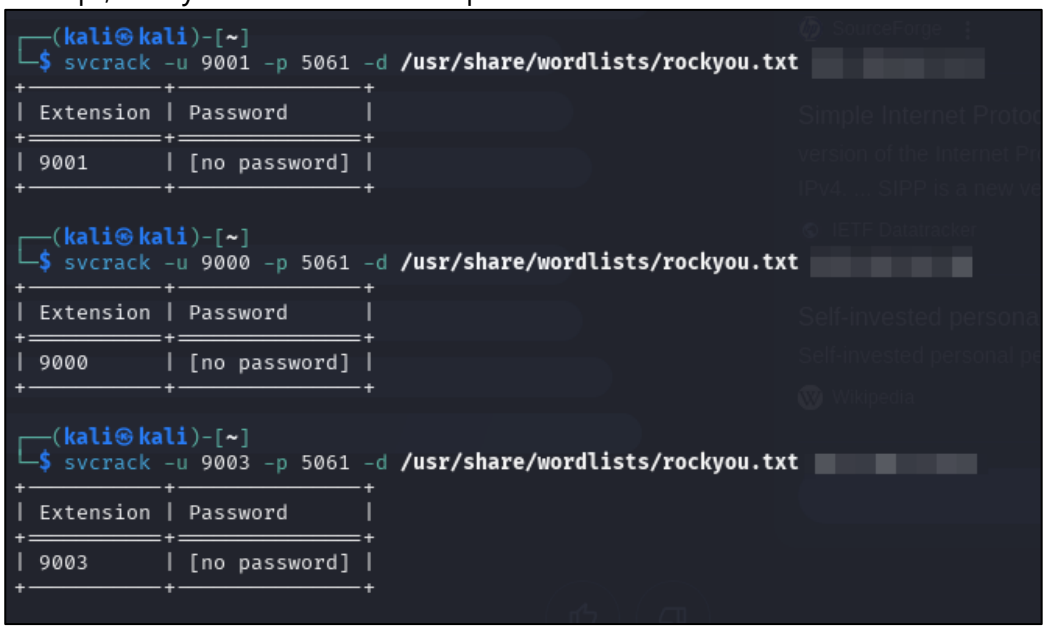
The SIP server is configured to use POP3 for authentication or message retrieval, which transmits credentials and data in plaintext. This exposes sensitive information to interception and increases the risk of credential compromise.



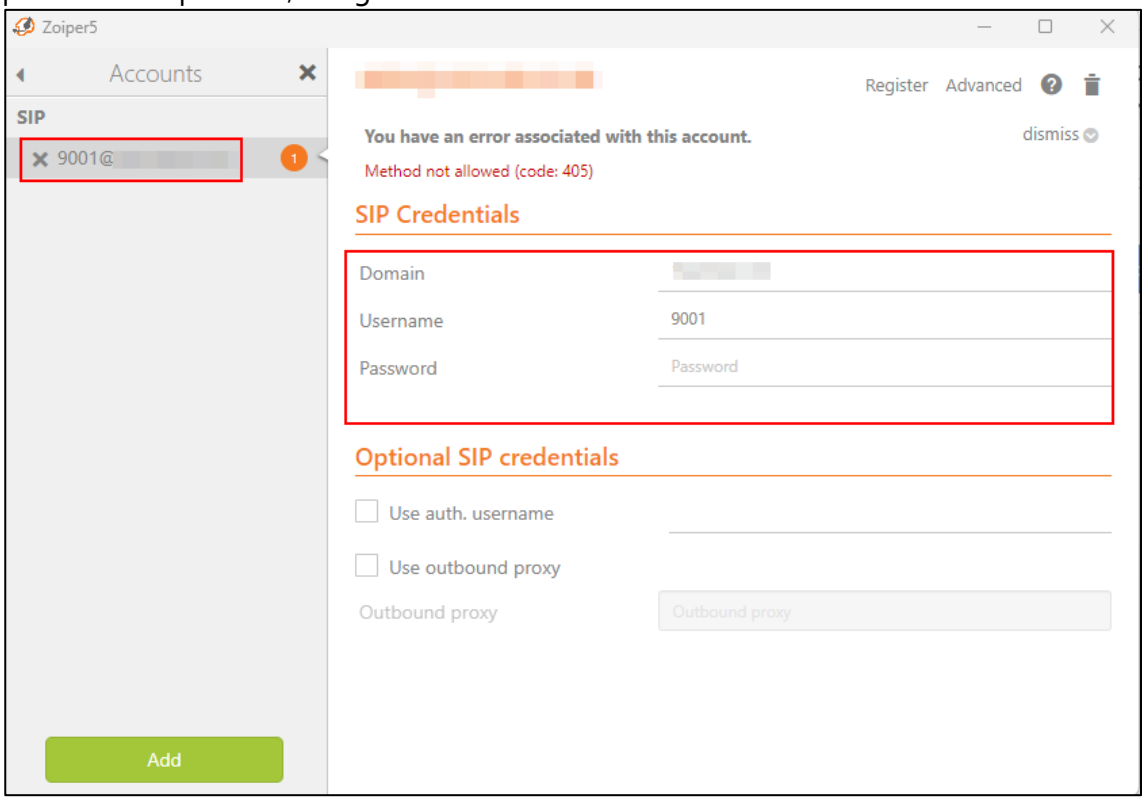
15. Brute Force Attack Against SIP Credentials

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

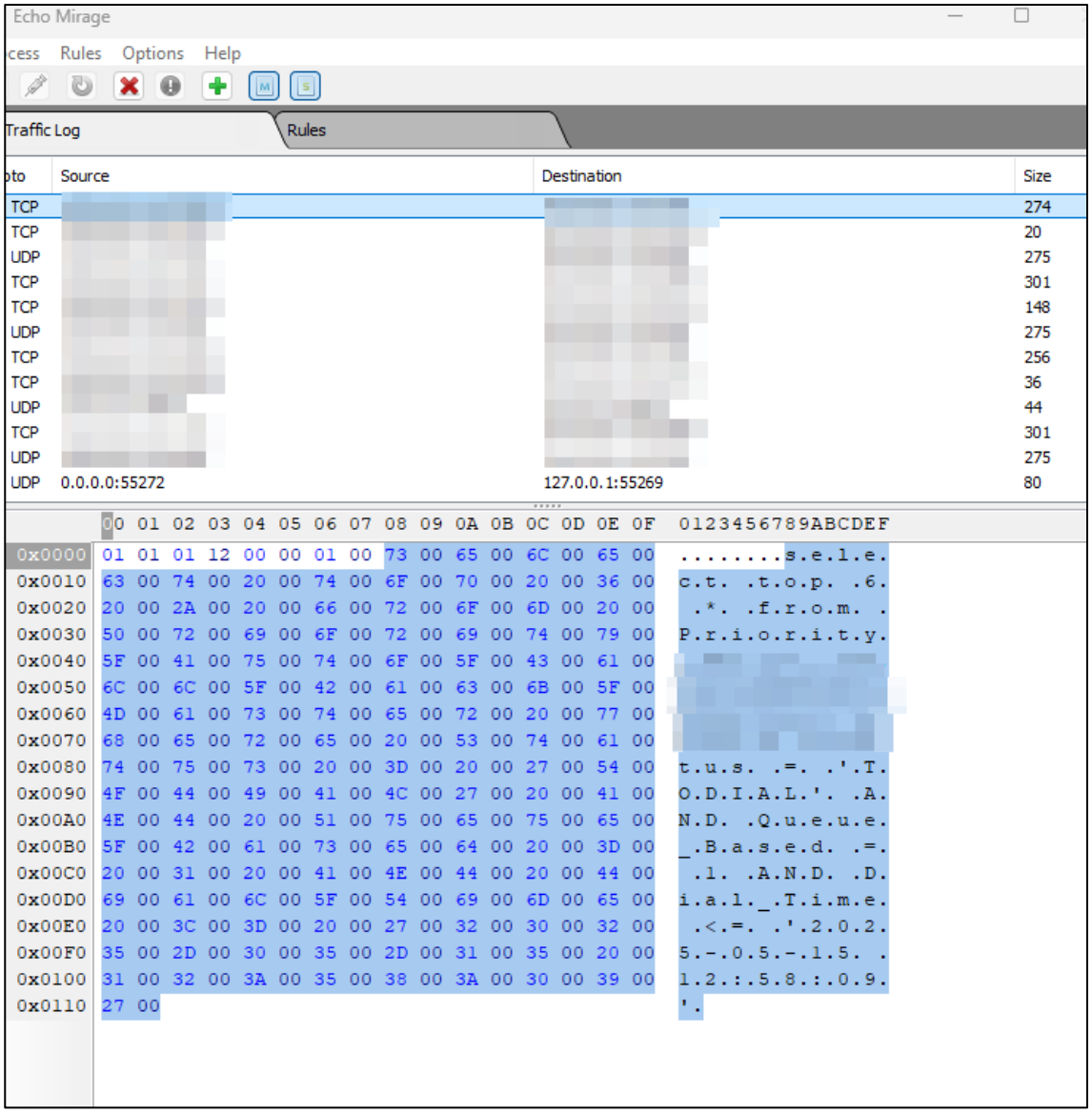


The user logs in to Zoiper5, a free softphone application for making VoIP calls through their PBX or preferred SIP provider, using valid credentials.



16. Cleartext Transmission of SQL queries/Possible SQL injection

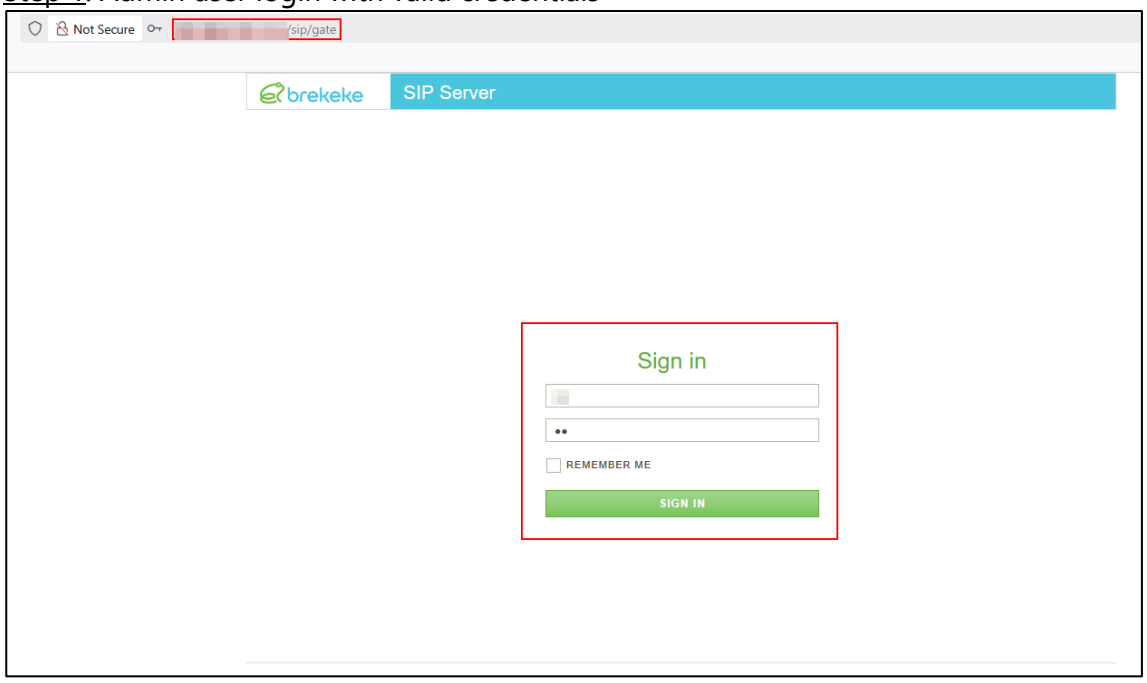
The thick client application transmits SQL queries to the database server in cleartext over the network. This allows an attacker with network access to intercept sensitive information such as database structure, query logic, and potentially user data or credentials.

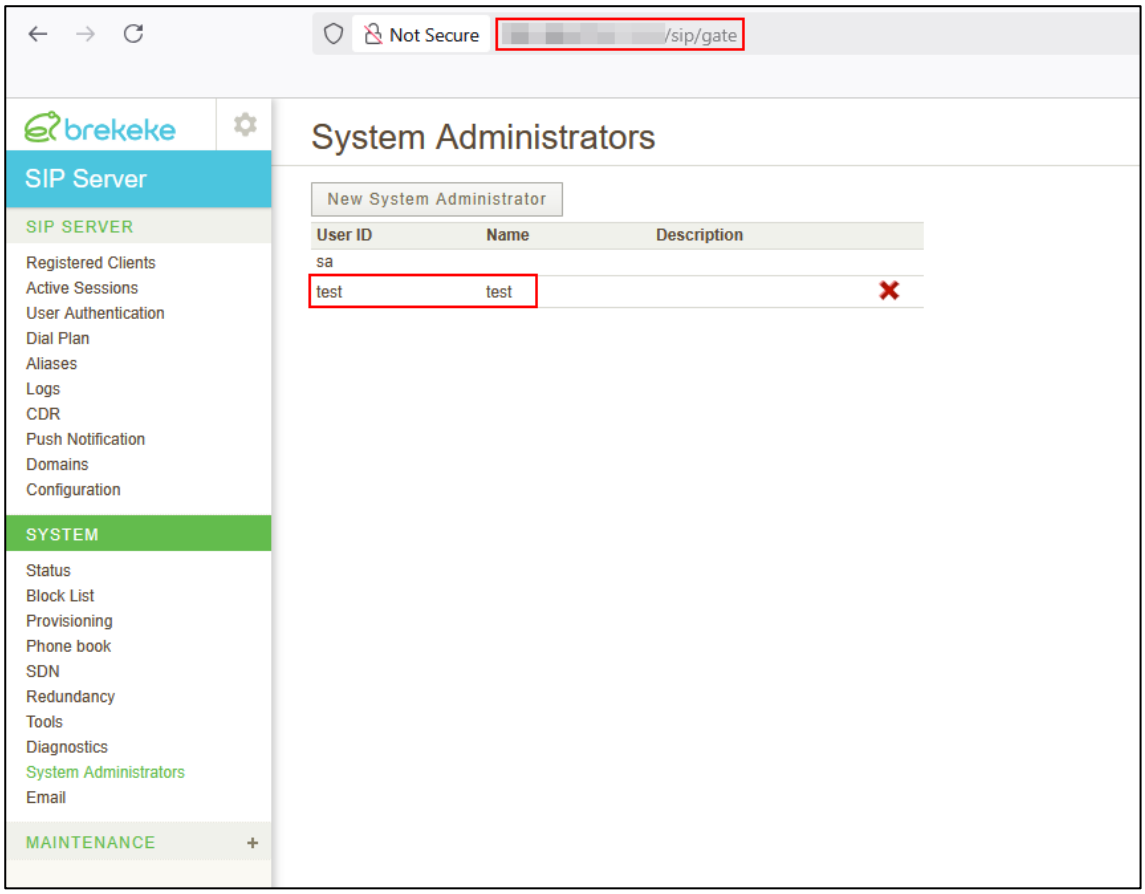


17. Application is vulnerable to CSRF attack

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

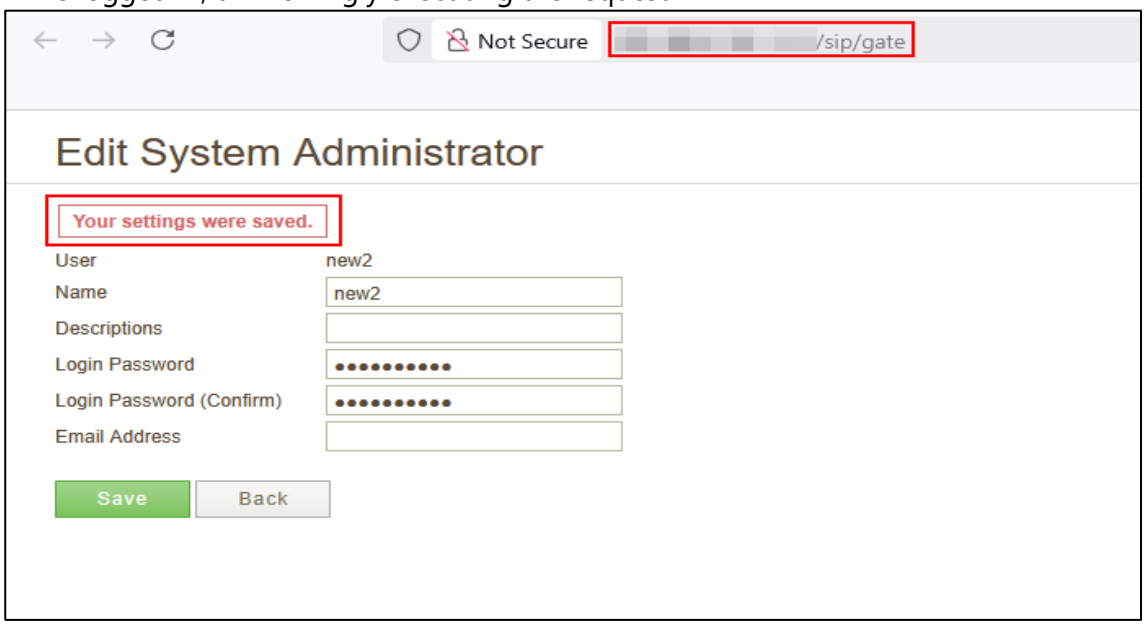




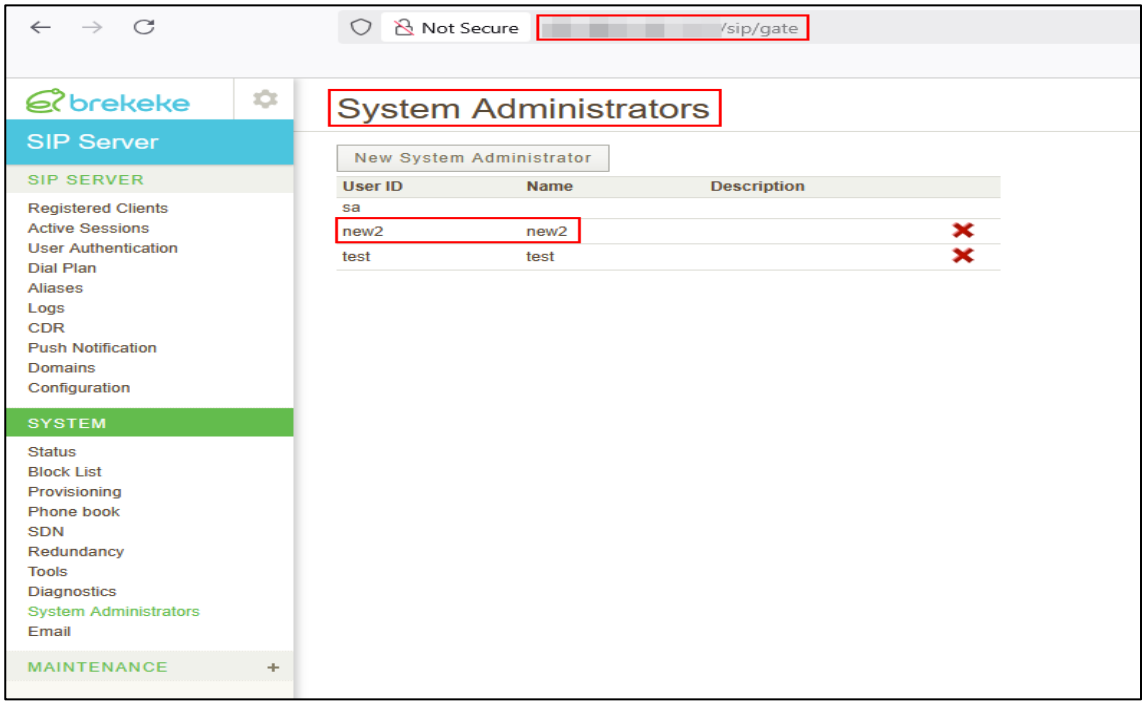
Step 2: The attacker crafts a malicious CSRF proof-of-concept (PoC) link that triggers the addition of a new admin account.

```
<!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>
```

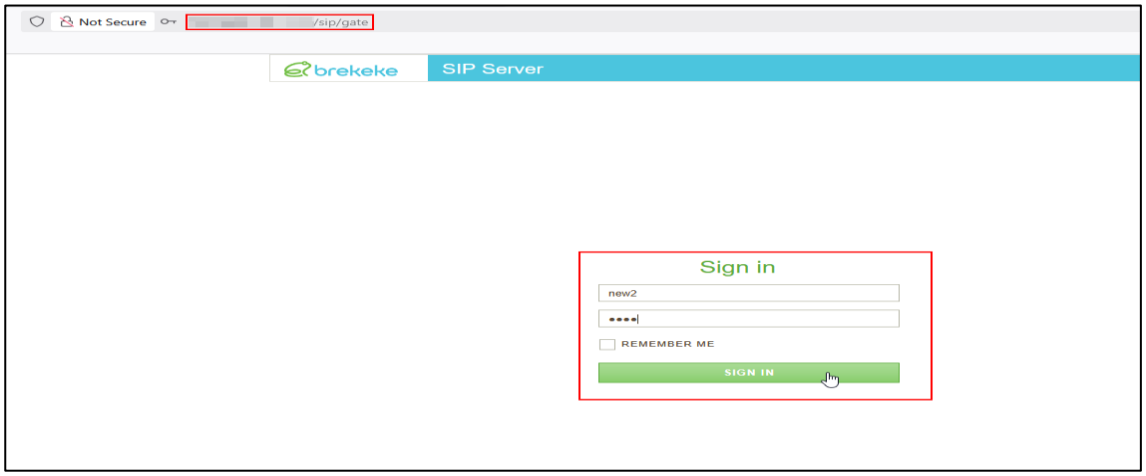
Step 3: The attacker sends this CSRF link to an authenticated admin user. The admin clicks the link while logged in, unknowingly executing the request

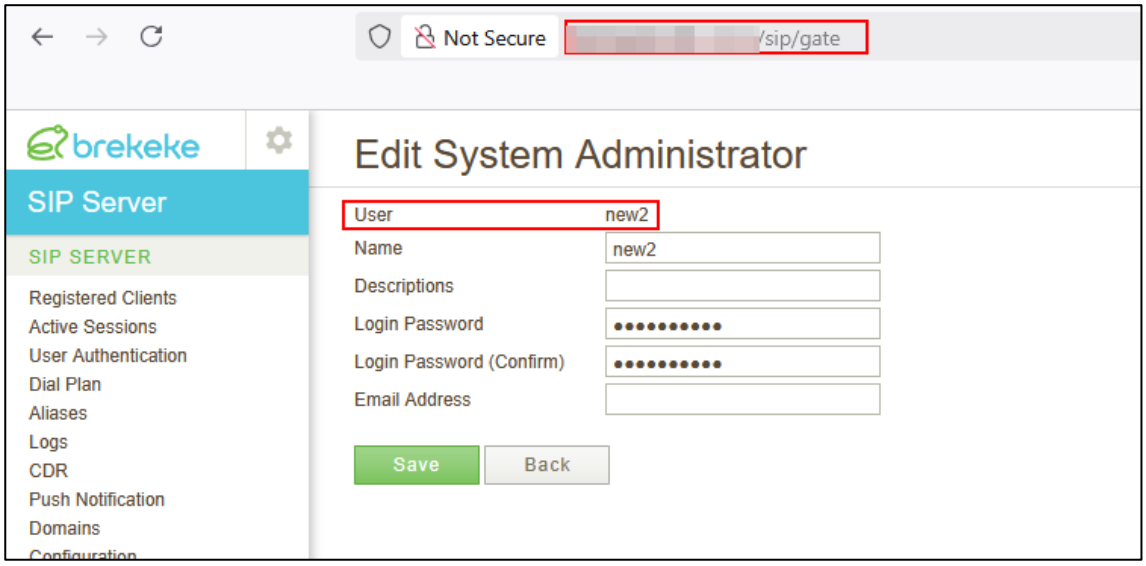


Step 4: The application processes the request and creates a new admin user without proper authorization checks.



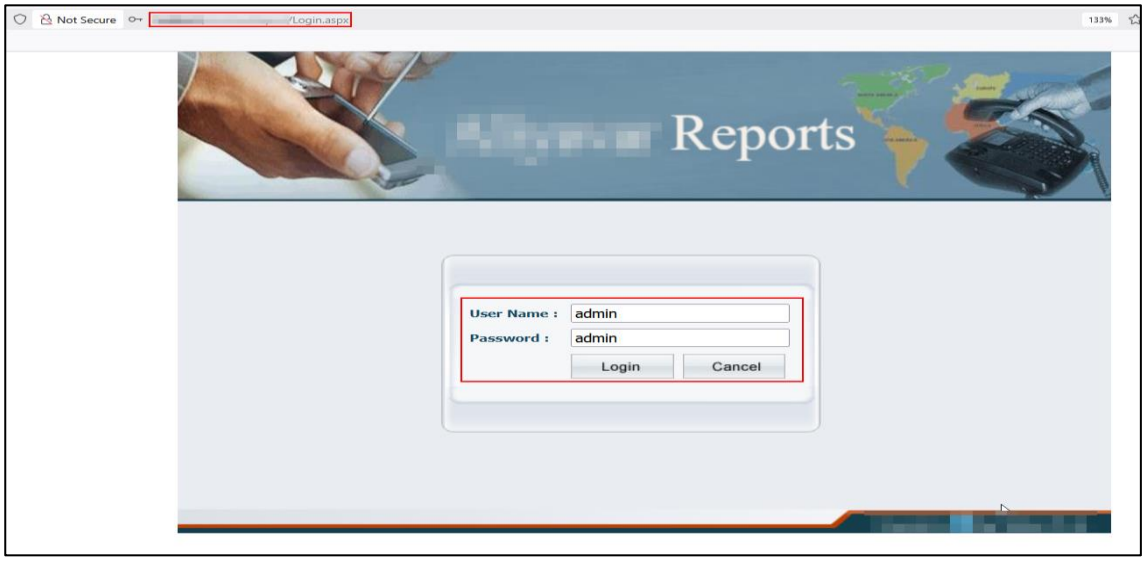
Step 5: The attacker then logs in using the newly created admin credentials and gains unauthorized access.





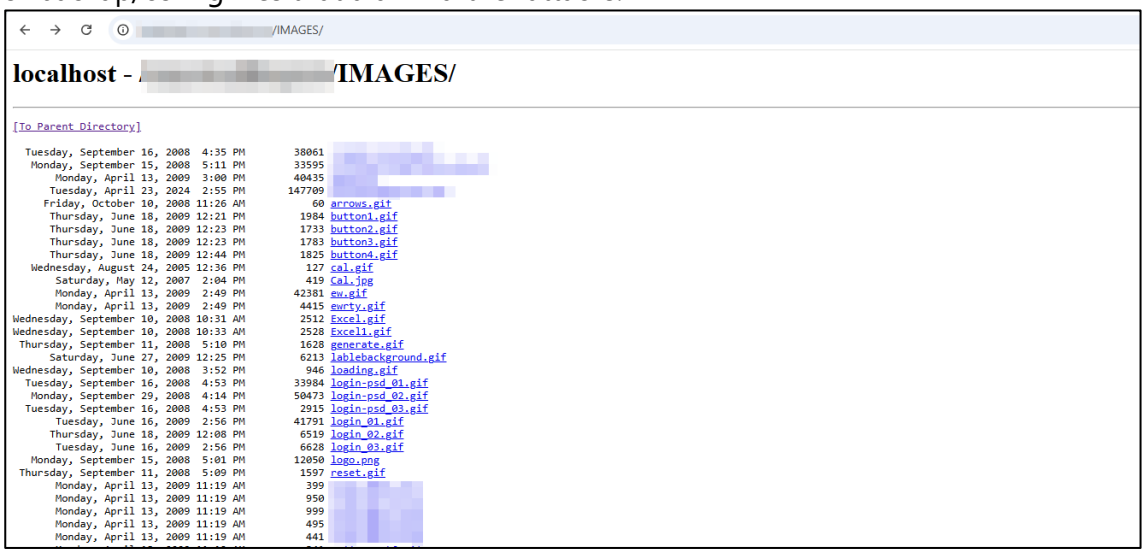
18. Weak Password Policy

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.



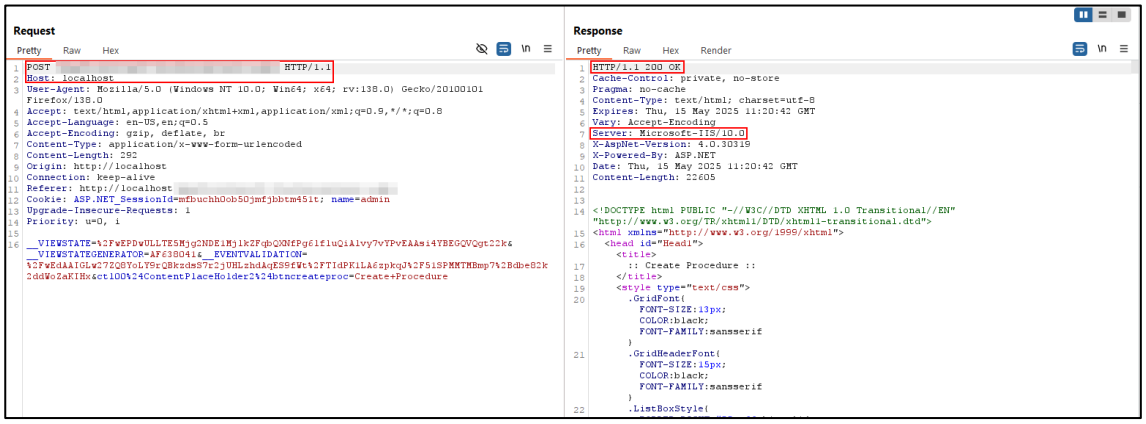
19. Application is vulnerable to Directory Traversal Attack

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.



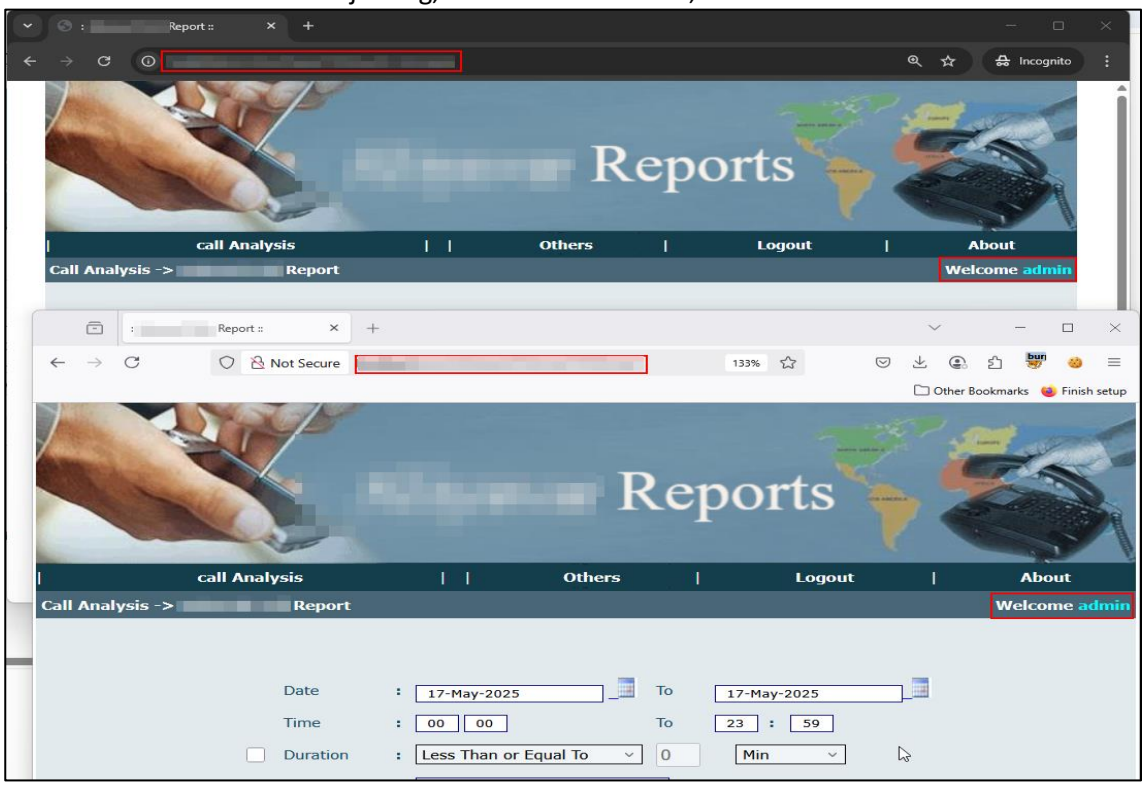
20. Application displays web server banner

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.



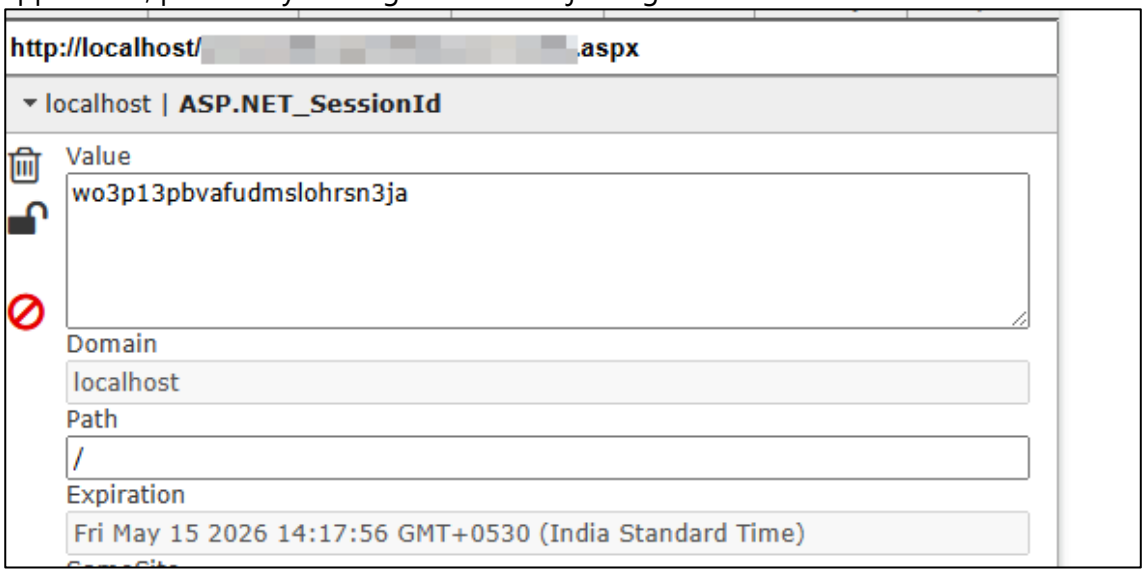
21. Application is vulnerable to simultaneous login

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.



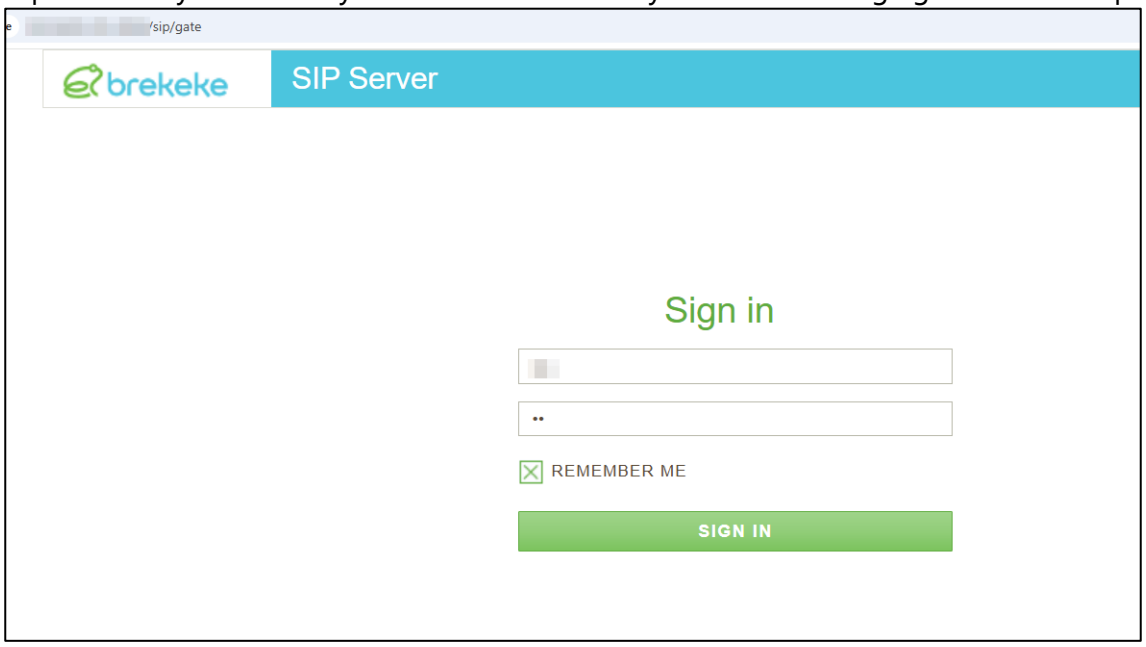
22. Path attribute not set in session cookie

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.



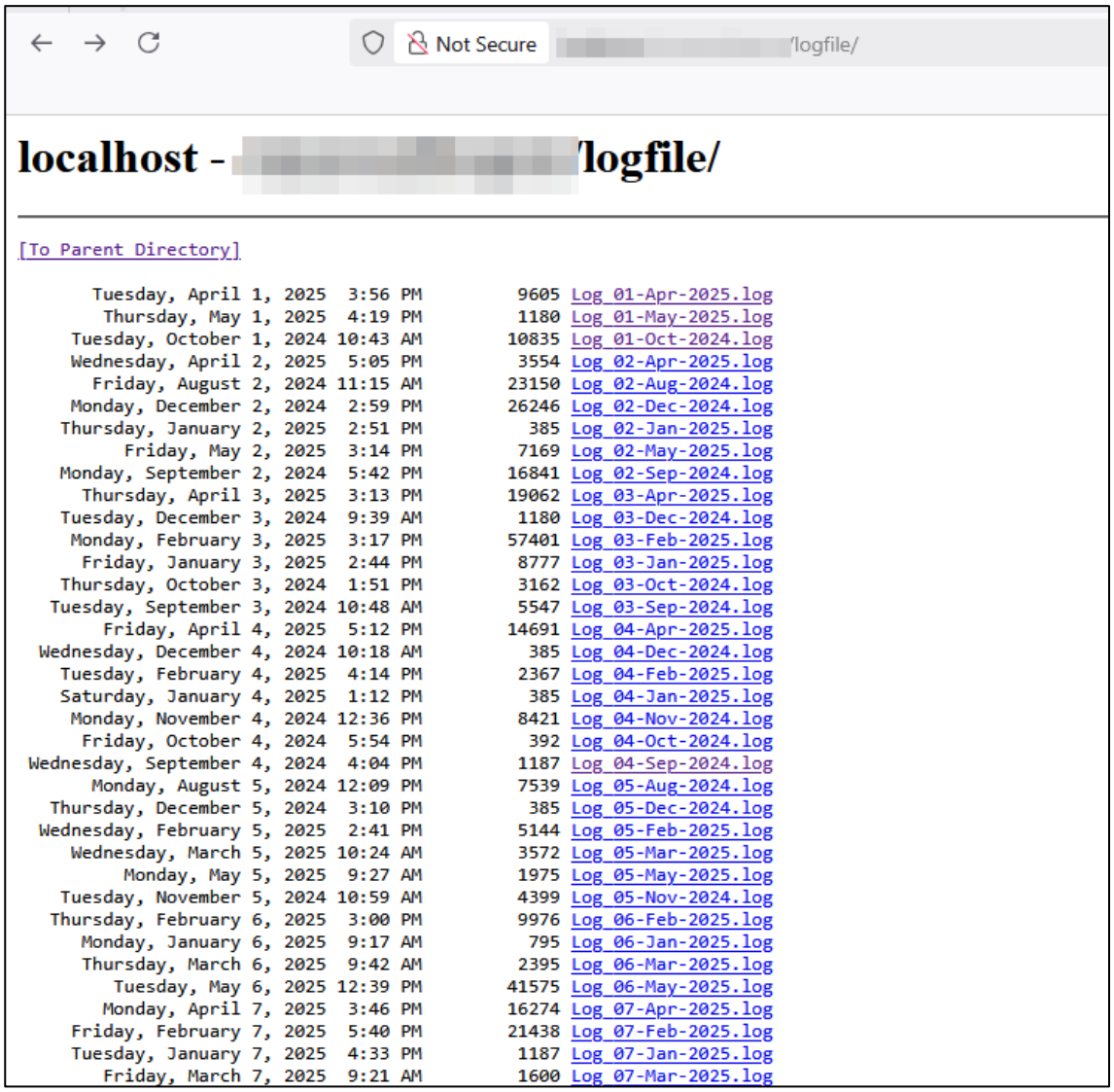
23. Default Passwords

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.



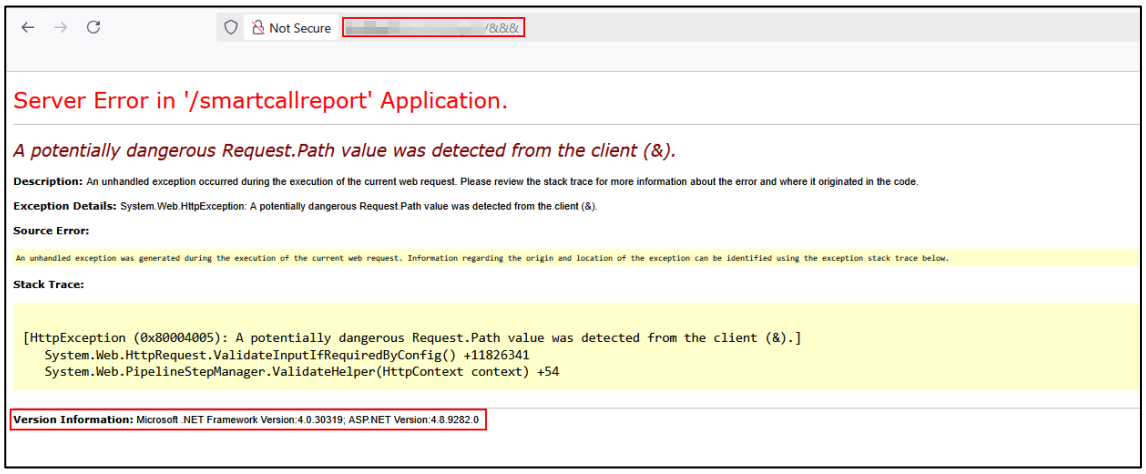
24. Exposed web server logs

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.



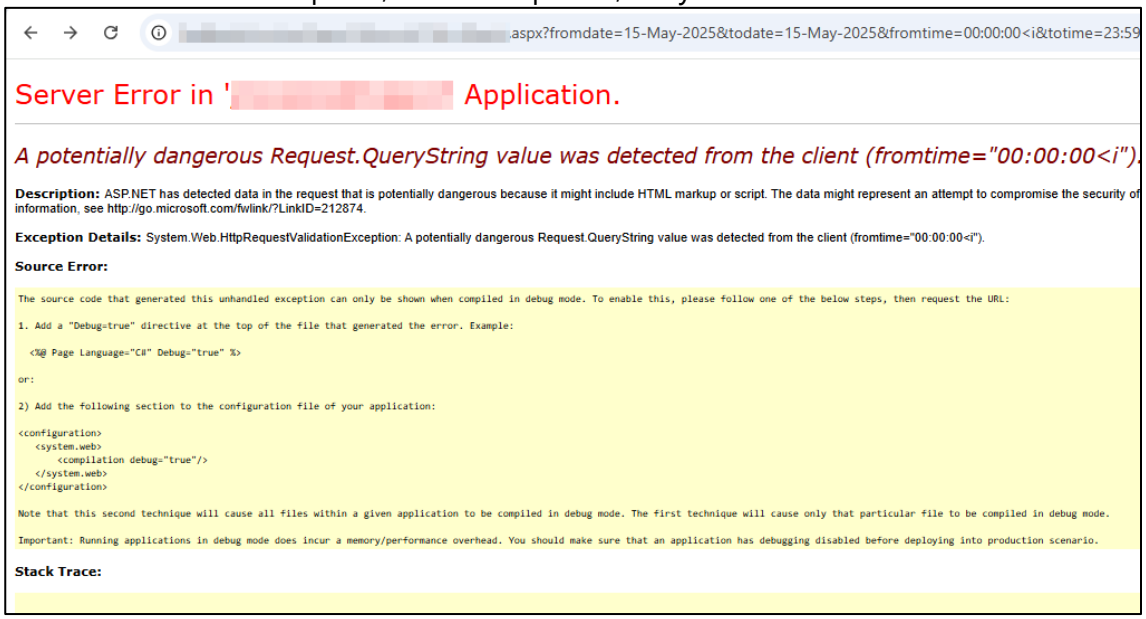
25. Framework Version Disclosure

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.



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.

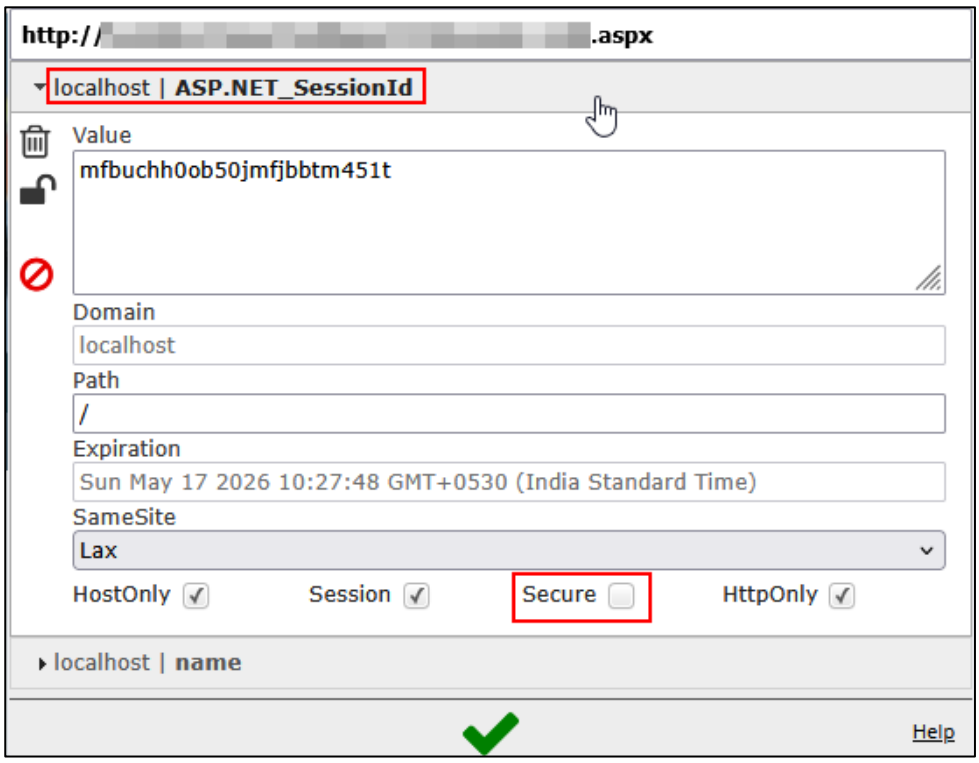


27.OPTIONS method enabled

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.

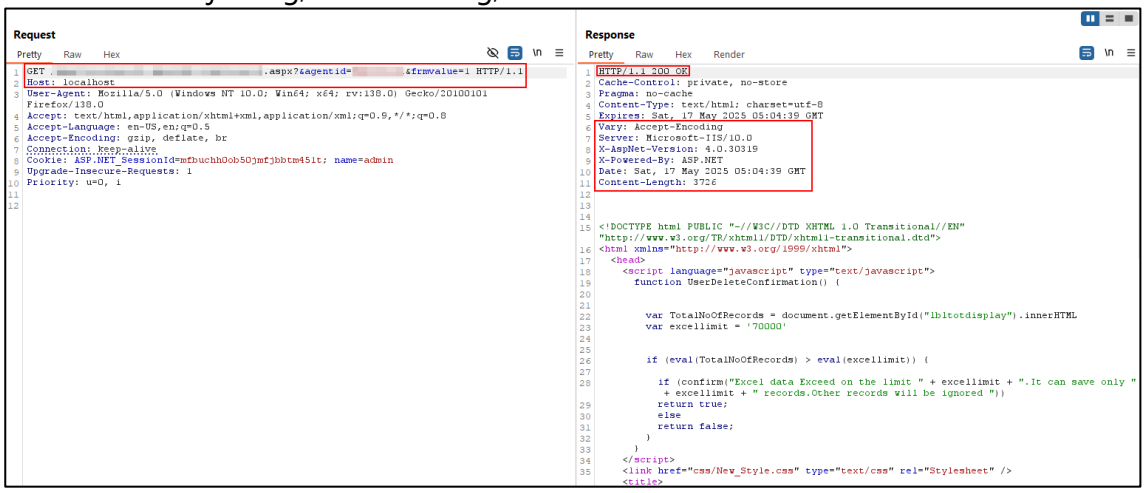
28.Secure flag is not set in session cookie.

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.



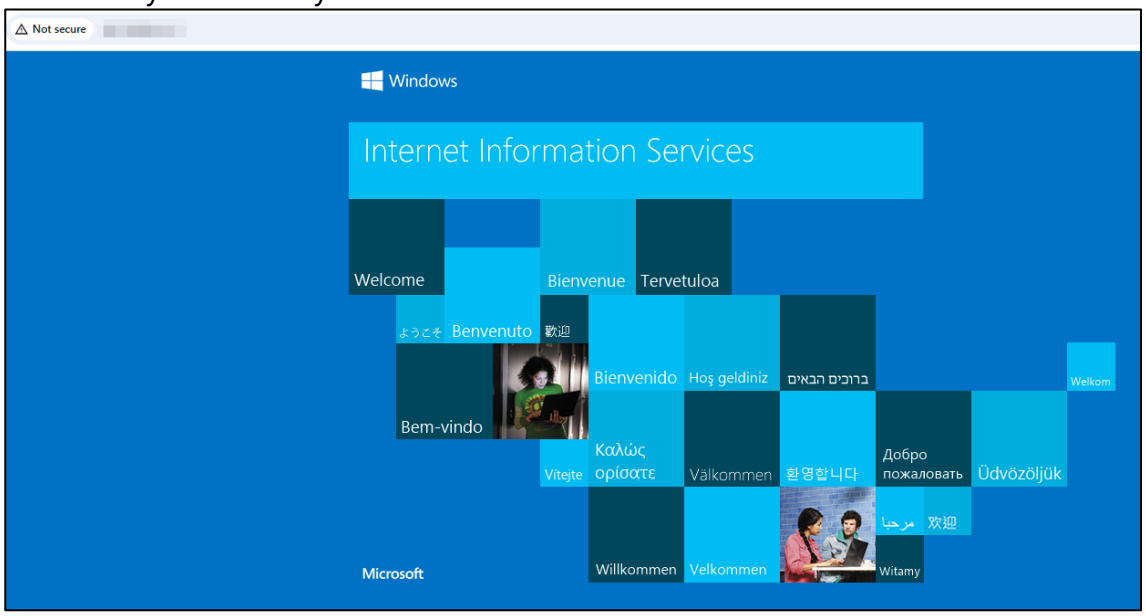
29. Security Headers are missing

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.



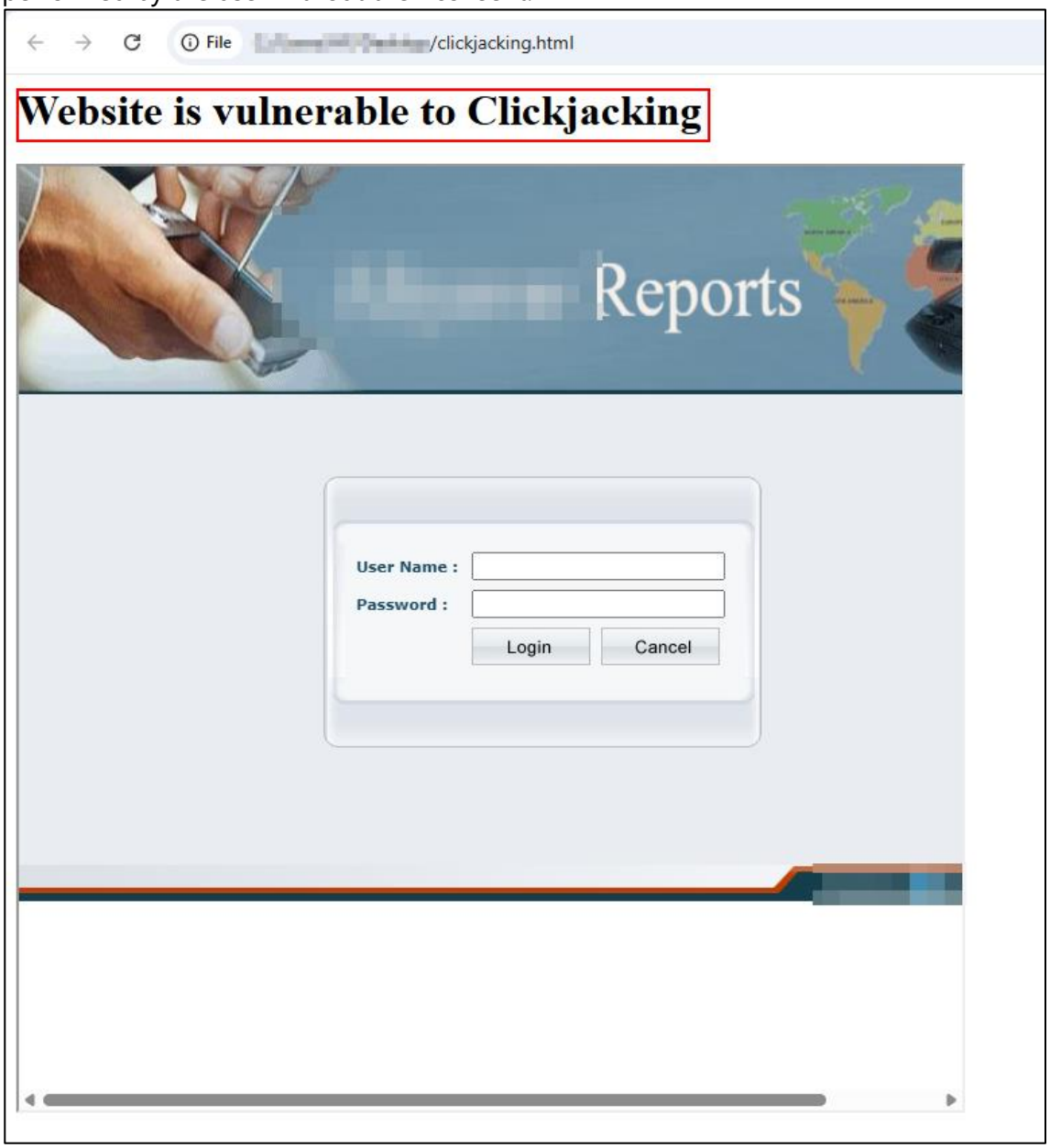
30. Default web page found on the application server

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.



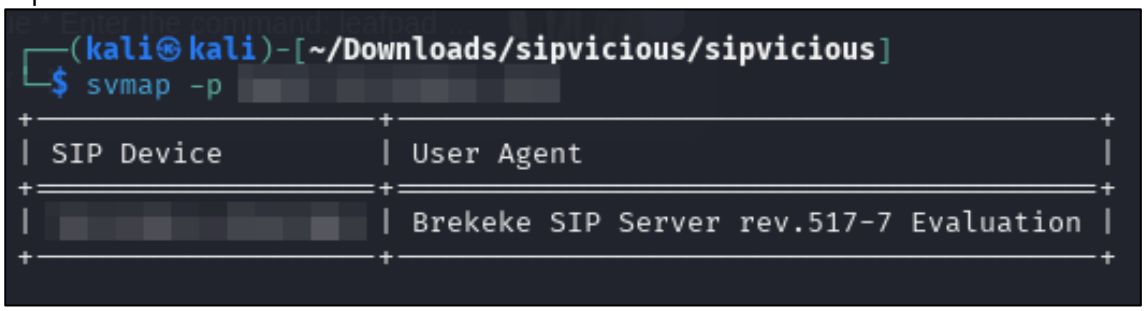
31. Application is vulnerable to ClickJacking attack.

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.



This page has been intentionally left blank.