

1.

Vulnerability Title: Login Credential Exposure

Severity: Critical

Description:

The application exposes or mishandles login credentials, specifically the password used for user authentication. During testing, it was observed that login passwords are either transmitted insecurely (e.g., via HTTP instead of HTTPS) or can be easily accessed or predicted due to weak authentication mechanisms.

Impact:

An attacker could exploit this vulnerability to gain unauthorized access to user accounts, leading to **complete account takeover**. If administrative credentials are compromised, it could result in full system compromise.

Steps to Reproduce (Example):

1. Navigate to the login page at <http://zero.webappsecurity.com/login.html>
2. Intercept the login request using Burp Suite.
3. Observe that the credentials are sent over an unencrypted HTTP connection.
4. An attacker on the same network (e.g., via a Man-in-the-Middle attack) can capture these credentials in plaintext.

Remediation:

- Implement HTTPS across the entire site to ensure secure transmission of sensitive data.
- Enforce strong password policies and hashing mechanisms (e.g., bcrypt or Argon2).
- Enable multi-factor authentication (MFA) to add an extra layer of security.
- Avoid exposing passwords or sensitive tokens in URLs, logs, or HTML comments.

The image displays a web browser window on the right and a network traffic capture in Burp Suite on the left. The browser window shows the 'Log in to ZeroBank' page with input fields for 'Login' and 'Password', a 'Keep me signed in' checkbox, and a 'Log in' button. The Burp Suite window shows a captured HTTP POST request to 'http://zero.webappsecurity.com/login.html'. The request body, visible in the 'Raw' tab, contains the following data: 'user_login=username&user_password=password&submit=Sign+in&user_token=f9011740-2260-46a2-a250-240b051c9677'. This demonstrates that the password is being transmitted in plaintext over an unencrypted HTTP connection.

.2

Vulnerability Title: Unrestricted Open Ports on Server

Severity: Critical

Description:

During a port scan of the target server, multiple open ports were identified that are not required for the application's core functionality. These ports may be exposing services that can be exploited by attackers to gain unauthorized access, escalate privileges, or extract sensitive information.

Impact:

Open and unused ports increase the attack surface of the server. Attackers can:

- Identify and exploit known vulnerabilities in the exposed services.
- Perform brute-force or DoS attacks on specific services (e.g., SSH, FTP).
- Gain unauthorized access to internal services that were not meant to be public-facing.
- Pivot into the internal network, especially if internal management ports (e.g., port 22, 23, 3306) are exposed..

Remediation:

- Close all unnecessary ports using a host-based firewall (e.g., iptables, ufw) or cloud security group.
- Restrict access to required ports using IP whitelisting or VPN-only access.
- Regularly perform port scans to monitor the exposure status.
- Implement network segmentation and ensure sensitive services are only accessible from trusted IPs.
- Ensure all exposed services are patched and configured securely.

```
└─# nmap -sV -p 53,8080,443,80 zero.webappsecurity.com -Pn
Starting Nmap 7.94SVN ( https://nmap.org ) at 2025-04-13 03:03 EDT
Nmap scan report for zero.webappsecurity.com (54.82.22.214)
Host is up (0.24s latency).
rDNS record for 54.82.22.214: ec2-54-82-22-214.compute-1.amazonaws.com

PORT      STATE  SERVICE  VERSION
53/tcp    filtered domain
80/tcp    open   http     Apache Tomcat/Coyote JSP engine 1.1
443/tcp   open   ssl/http Apache httpd 2.2.6 ((Win32) mod_ssl/2.2.6 Open
SSL/0.9.8e mod_jk/1.2.40)
8080/tcp  open   http     Apache Tomcat/Coyote JSP engine 1.1

Service detection performed. Please report any incorrect results at https
://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 35.02 seconds
```

3.

Vulnerability:- Application is running on insecure channel

Severity: High

🚩 Impact:

Data transmitted can be intercepted or modified, leading to data breaches, session hijacking, or man-in-the-middle (MITM) attacks.

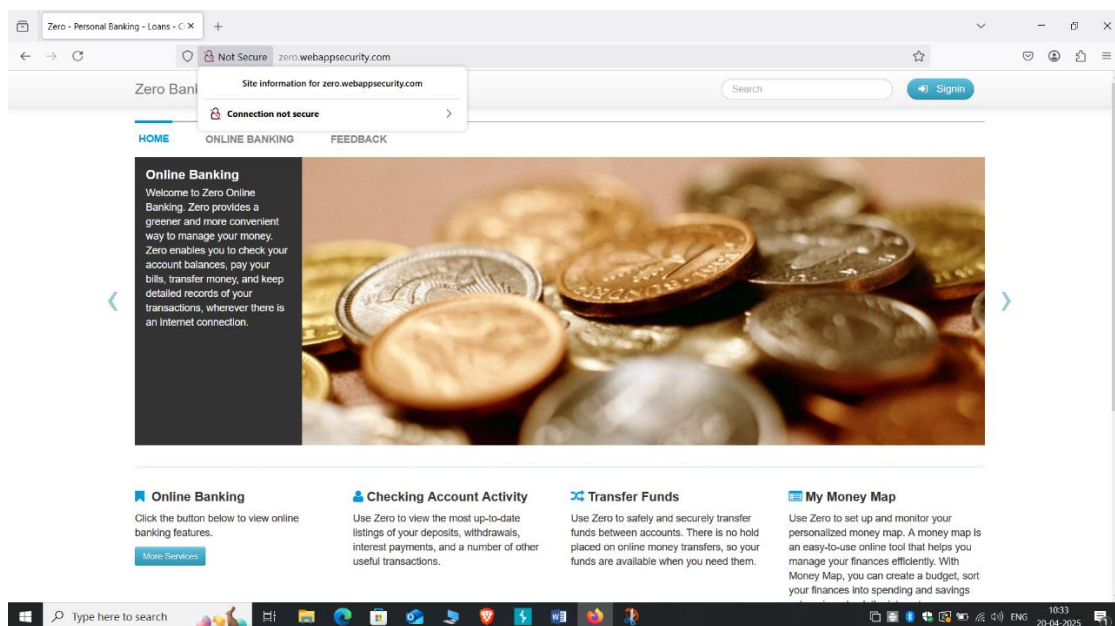
🔧 Description:

The application is using an unencrypted communication protocol (e.g., HTTP) instead of a secure one (e.g., HTTPS). This exposes all transmitted data to potential attackers, especially over public or untrusted networks.

🔧 Remediation:

- Configure the application to use **HTTPS** by installing a valid **SSL/TLS certificate**.
- Redirect all HTTP traffic to HTTPS.

Regularly test the application for secure communication compliance



4.

Vulnerability :- Outdated Components Being Used

Severity: High

Impact:

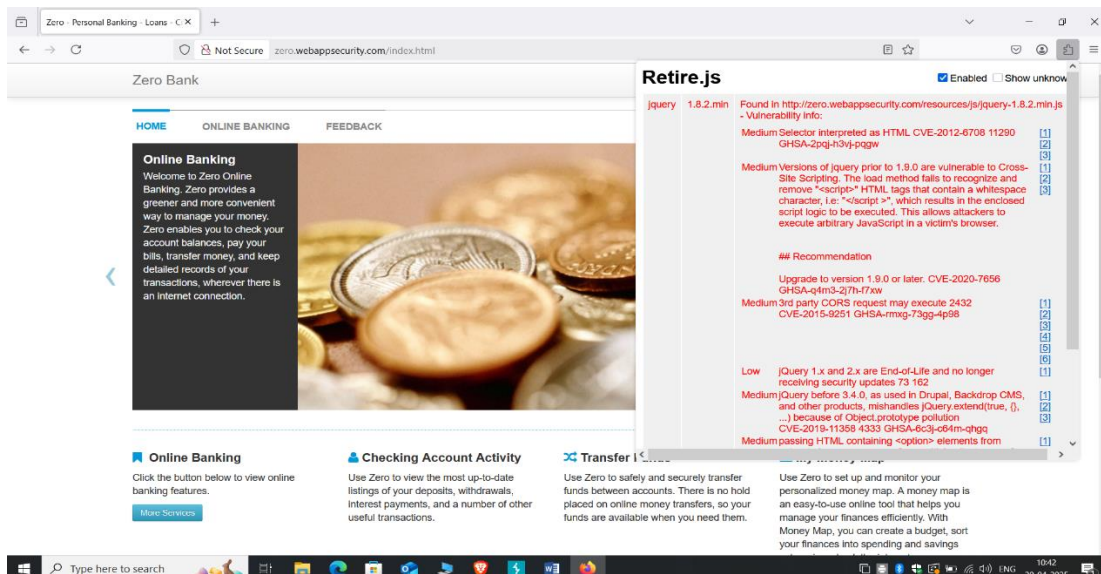
Outdated components may contain known vulnerabilities that can be exploited by attackers to compromise the application or underlying system.

Description:

The application is using third-party libraries, frameworks, or dependencies with known security flaws or bugs. These components may not receive security patches, making the system vulnerable.

Remediation:

- Regularly update all components and dependencies to their latest stable versions.
- Use tools like **OWASP Dependency-Check**, **npm audit**, or **pip-audit** to identify vulnerable packages.
- Monitor security advisories related to the used technologies.



5.

Vulnerability :- Vulnerable to click jacking

 **Severity:** Medium

Impact:

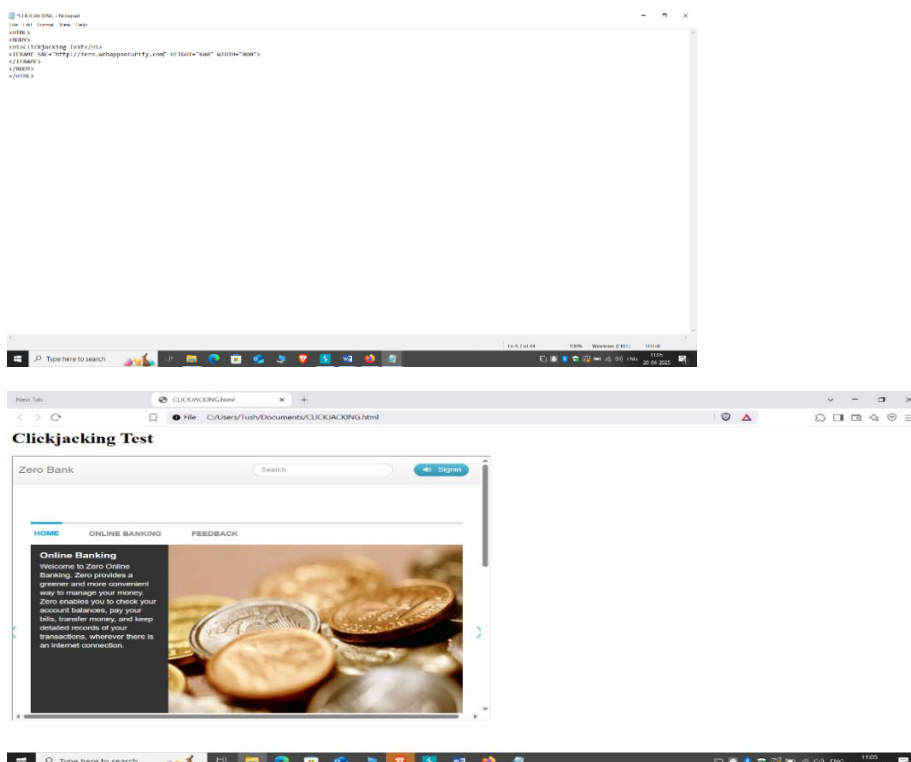
Attackers can trick users into clicking on invisible or disguised elements, leading to unintended actions such as account changes, data exposure, or malicious transactions.

Description:

The application is susceptible to **Clickjacking**, where a malicious website can embed the application within an invisible frame, misleading the user into interacting with elements they cannot see, resulting in security risks.


Remediation:

- Implement the **X-Frame-Options** HTTP header to prevent the application from being embedded in an iframe.
- Use **Content Security Policy (CSP)** with the `frame-ancestors` directive to control which sites can embed your pages.
- Enable **frame busting** JavaScript techniques as an additional defense
-



6.

Vulnerability :- . Server version disclosure

 **Severity:** Medium

Impact:

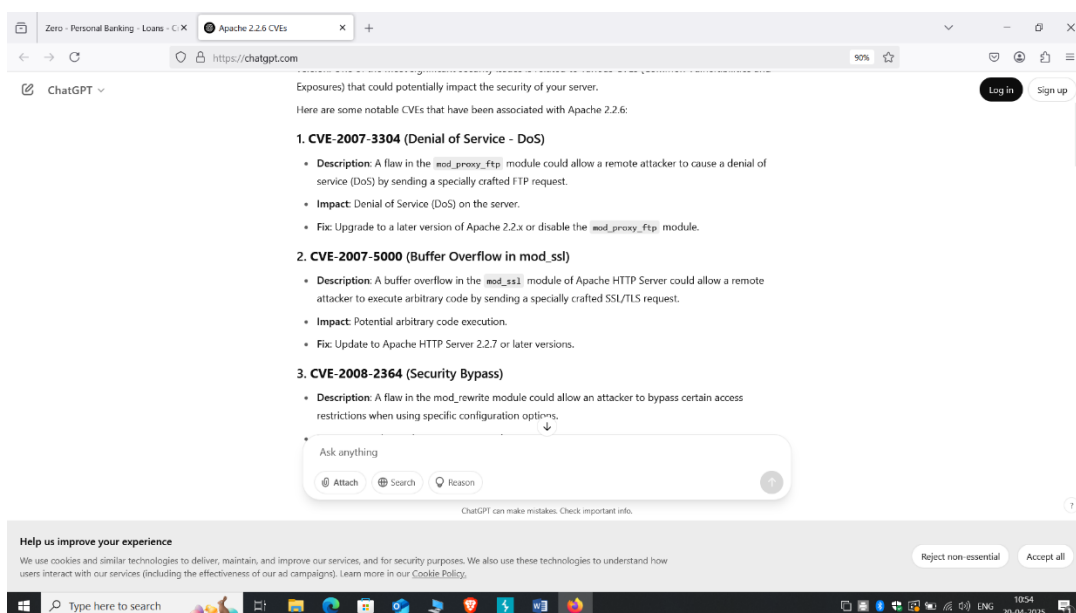
Disclosing the server version can provide attackers with detailed information about the system, allowing them to exploit known vulnerabilities specific to that version.

Description:

The application or server reveals information about its version (e.g., in HTTP headers, error messages, or metadata), which can help attackers identify potential exploits based on known vulnerabilities in that version.

Remediation:

- Disable version information in **HTTP response headers** (e.g., `Server` header).
- Configure the server to return generic error messages without disclosing version details.
- Regularly update the server and software to mitigate risks associated with known vulnerabilities



7.

● **Vulnerability:** Cross-Site Scripting (XSS)

🔒 **Severity:** High

📖 **Description:**

The application fails to properly sanitize user input before rendering it in the browser. This allows attackers to inject malicious scripts into web pages viewed by other users. The script executes in the context of the user's browser session.

☀️ **Impact:**

An attacker could steal session cookies, deface the website, redirect users to malicious sites, or perform actions on behalf of users without their consent.

✂️ **Remediation:**

- Properly **sanitize and encode** all user input before rendering it to the browser.
- Use secure frameworks that **auto-escape** output (e.g., React, Angular).
- Implement **Content Security Policy (CSP)** to restrict script execution.
- Validate input both on the **client and server side**.

Zero Bank

Search Signin

Add Currency

Home
Users
Currencies

ID

Country

Name

Add

Name	Password	SSN
Leeroy Jenkins	VIZ10AWT8VL	536-48-3769
Stephen Bowen	OTZ07BXM0BE	607-58-7435
Linus Moran	FKO04SXA7TI	247-54-1719
Nero Chan	TXJ77CQO5EI	578-13-3713
Kadeem Higgins	MFC50OQE7VO	449-20-3206
Quinn Bur		008-70-6738
Davis Tho		574-56-1932
Lester Kel		330-58-4012

zero.webappsecurity.com

1

OK