CYBERSTRIKE User Manual

By P1AG4:

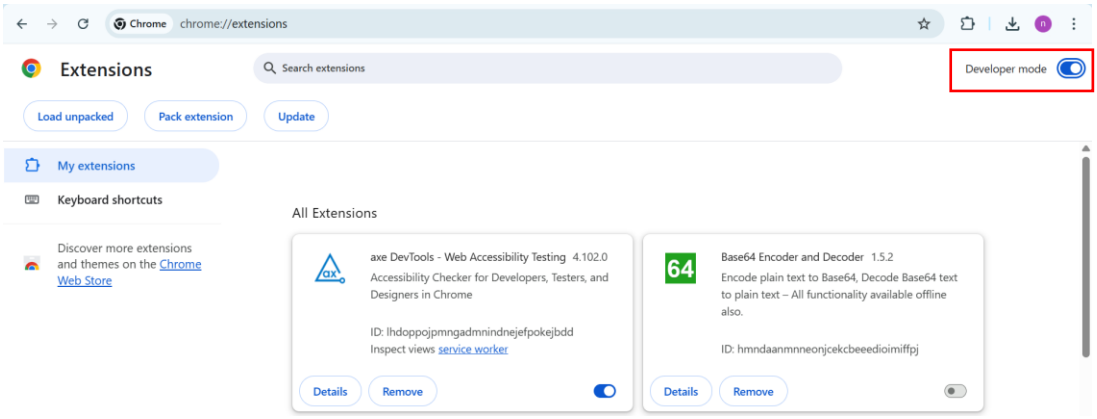
- Raul Gunawan S/O Iqbal Suppiah (2301895)
- Brandon Koh Lin Xi (2301902)
  Foo Zhan Yong (2301905)
- Qothrunnada Istiqamah Binte Muhammad Azhar (2301924)
- Muhammad Azzi Izzuan Bin Azahar (2301955)

Table of Contents

1. Setup

a) Visit chrome://extensions

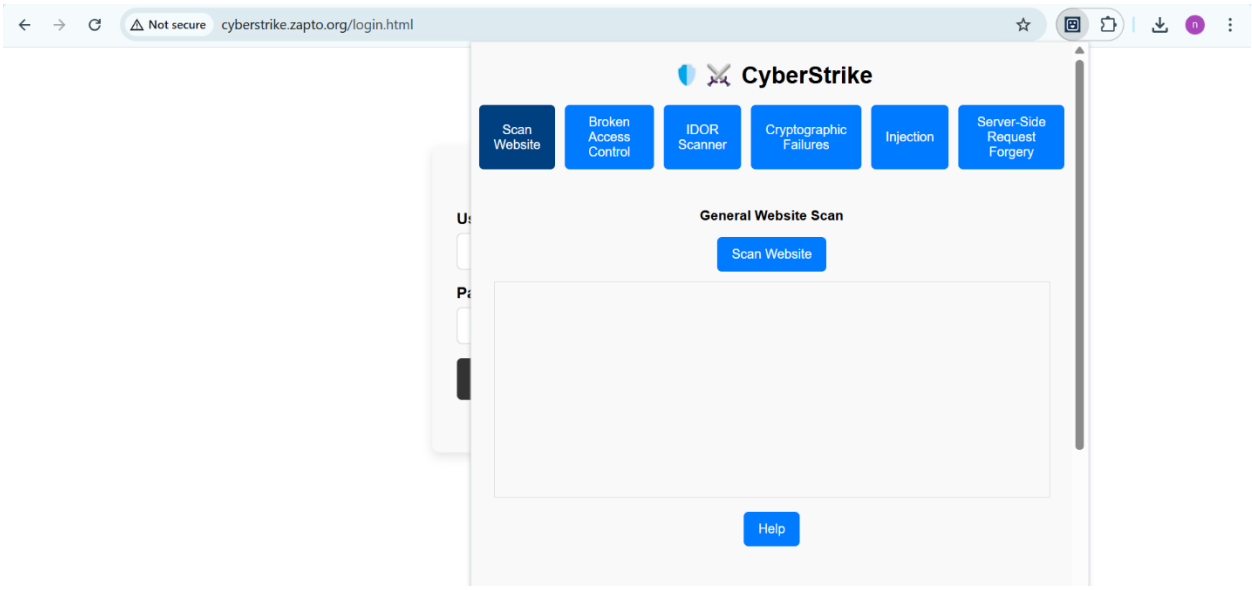
b) Toggle On “Developer Mode”



c) Click “Load unpacked” and select “CyberStrike” File

<to insert image of content of finalized submission folder>

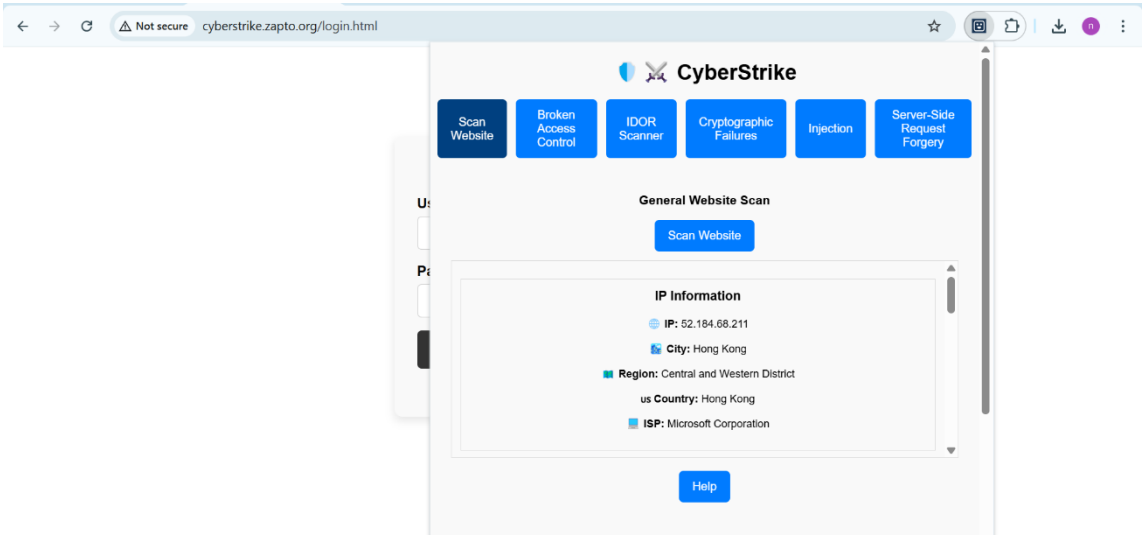
d) Click on top right to activate CyberStrike web extension



2. Vulnerability Scanning and Exploitation Features

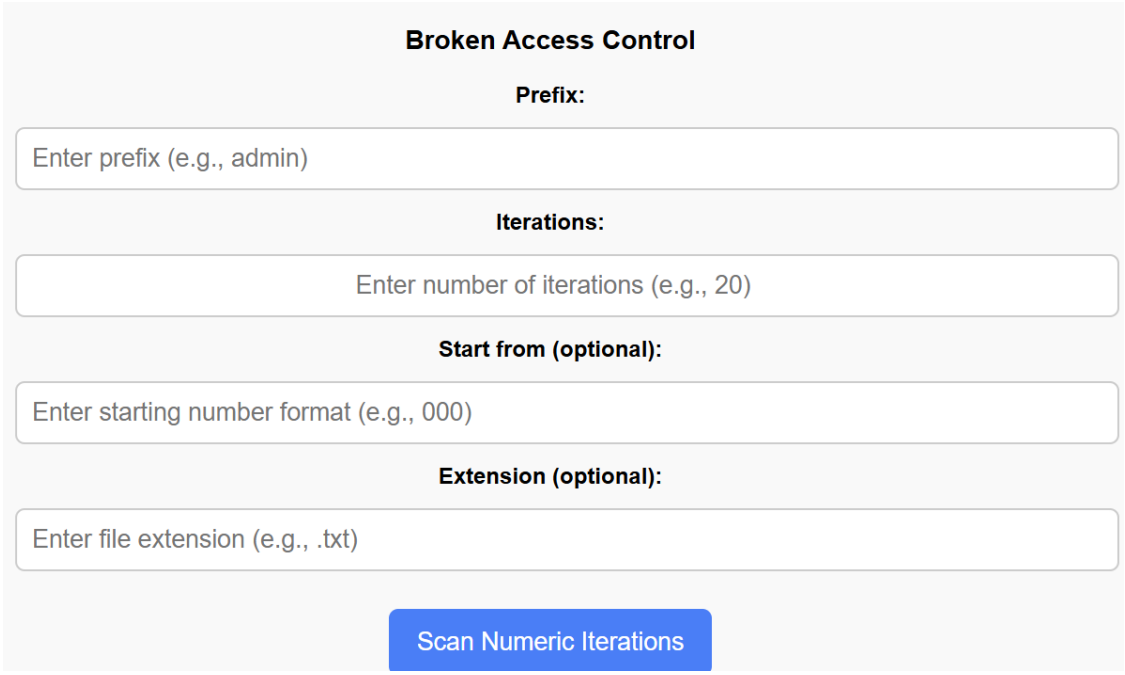
a) Website Scan

Selecting the “Scan Website” option would display information about the website such as its IP address, City, Country and Region the website is hosted at, and its Internet Service Provider (ISP).



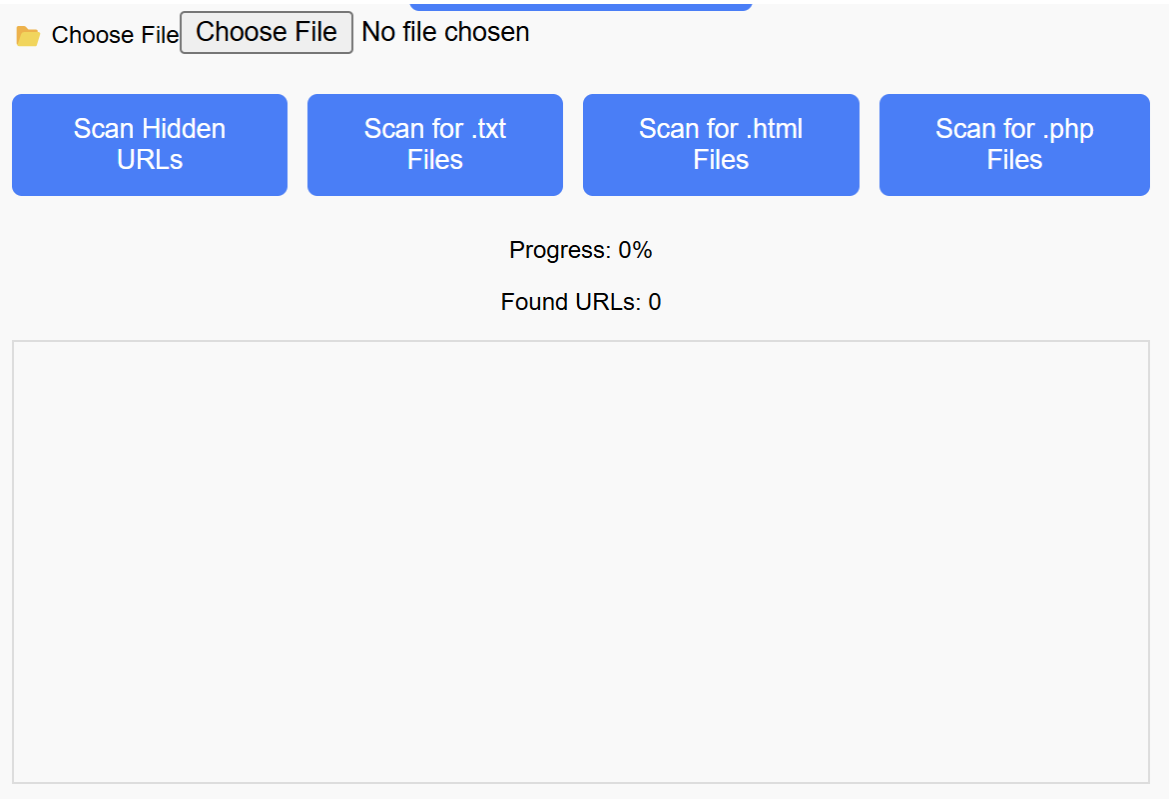
b) Broken Access Control

i. Numeric Iteration Scan



Use the **Numeric Iteration Scan** feature to test predictable URL patterns. Enter a **Prefix** (e.g., admin), the number of **Iterations** (e.g., 20), an optional **Start From** value (e.g., 001), and an optional **Extension** (e.g., .php). The tool will then generate URLs (e.g., admin1.php, admin2.php, etc.) and check each one in sequence. This helps identify unprotected or hidden files that follow numerical naming conventions.

ii. Wordlist-Based Scans

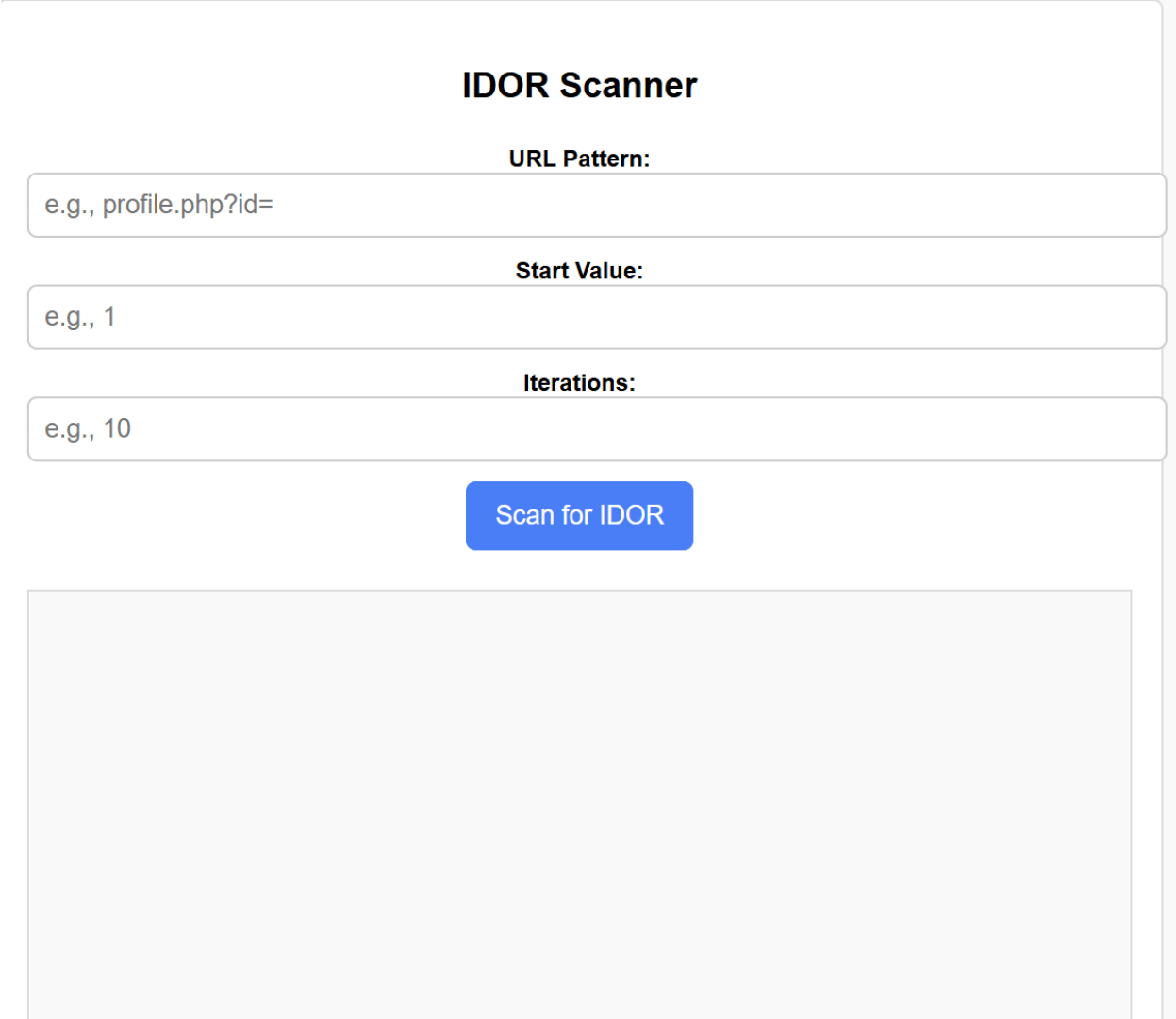


Upload a custom wordlist or use the built-in default wordlist to automate URL discovery:

- **Scan Hidden URLs**: Searches for directory or file paths using each entry in the wordlist.
- **Scan for .txt Files**: Appends .txt to each wordlist entry to find text-based endpoints.
- **Scan for .html Files**: Appends .html to locate accessible HTML files.
- **Scan for .php Files**: Appends .php for discovering PHP endpoints.

As the scan progresses, the extension shows the **progress percentage** and **list of discovered URLs**. If no resources are found, it indicates no accessible endpoints matched the given patterns or file type.

c) IDOR Scanner



The **IDOR (Insecure Direct Object References) Scanner** tests whether a resource parameter (like id) can be incremented or decremented to access other users' data without proper authorisation. Enter a **URL Pattern** (e.g., profile.php?id=), specify a **Start Value** (e.g., 1), and the number of **Iterations** (e.g., 10). The scanner then requests *profile.php?id=1*, profile.php?id=2, and so on, checking which responses are accessible. Pages that return valid data might indicate a potential IDOR vulnerability if they reveal information about other user IDs.

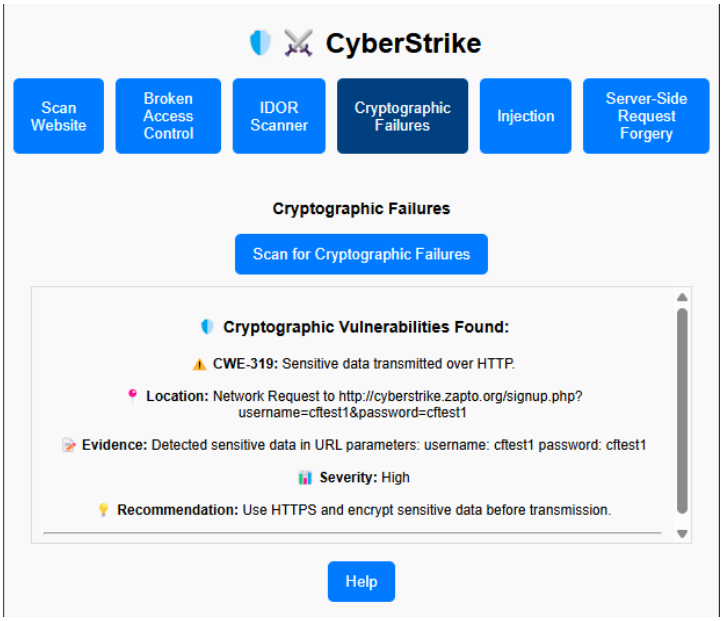
d) Cryptographic Failures

i. Scanning

After selecting **“Cryptographic Failures”** and clicking **“Scan for Cryptographic Failures”**, the tool passively monitors your current tab’s traffic and checks page scripts for insecure key usage. Any findings related to **CWE-319** (Cleartext Transmission), **CWE-321** (Hardcoded Key), or **CWE-523** (Unprotected Transport of Credentials) will be displayed in the popup. This includes:

- Credentials or secrets sent over plain HTTP.
- Hardcoded API keys or tokens within client-side JavaScript.
- Login credentials transmitted in cleartext.

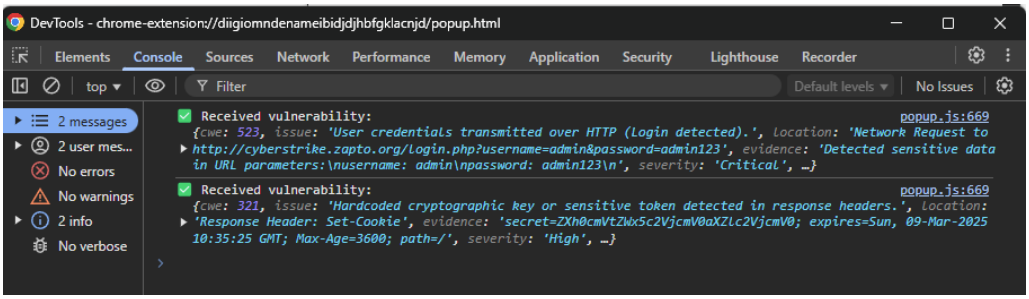
Each detected vulnerability entry highlights the **CWE number**, **severity**, **evidence**, and **remediation advice** to guide you in securing your application.



ii. Note

To prevent popup from closing, you may open the Chrome console to ensure the popup window is persistent as you navigate the site to detect vulnerabilities (Right click on extension popup > Inspect > Console)

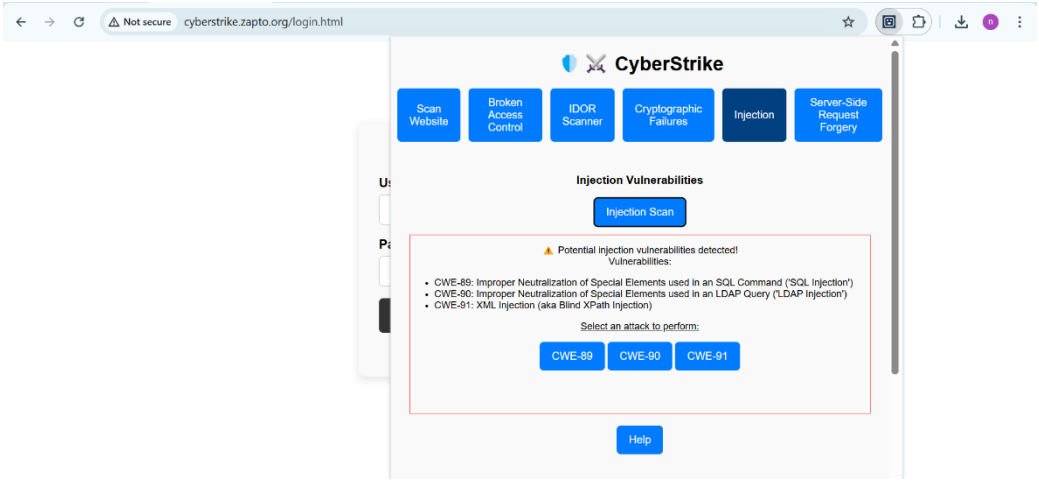
The console also logs the vulnerabilities detected.



e) Injection

i. Scanning

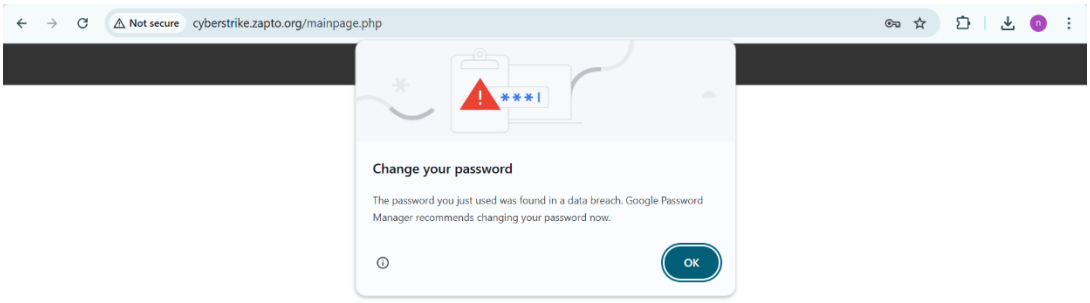
After selecting 'Injection' and pressing 'Scan Website,' the tool will scan the current page of the website and output the detected CWEs related to injection vulnerabilities.



ii. Exploitation

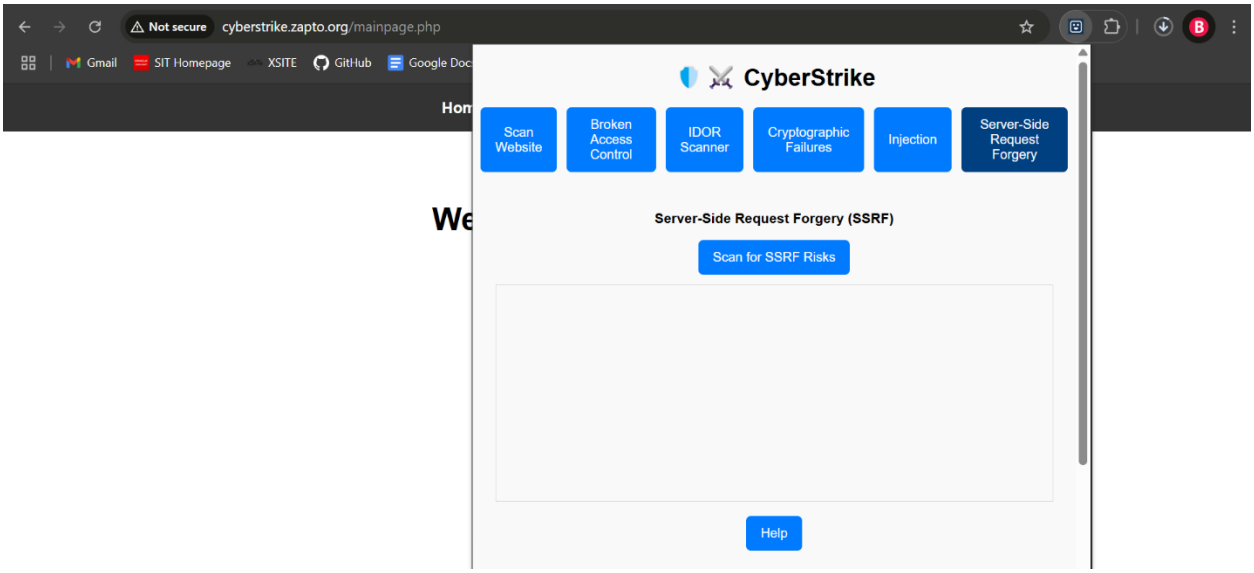
To perform exploitation on the website, the user can click on one of the listed CWEs.

In this case, after clicking ‘CWE-89’, user was able to successfully bypass login using SQL Injection.

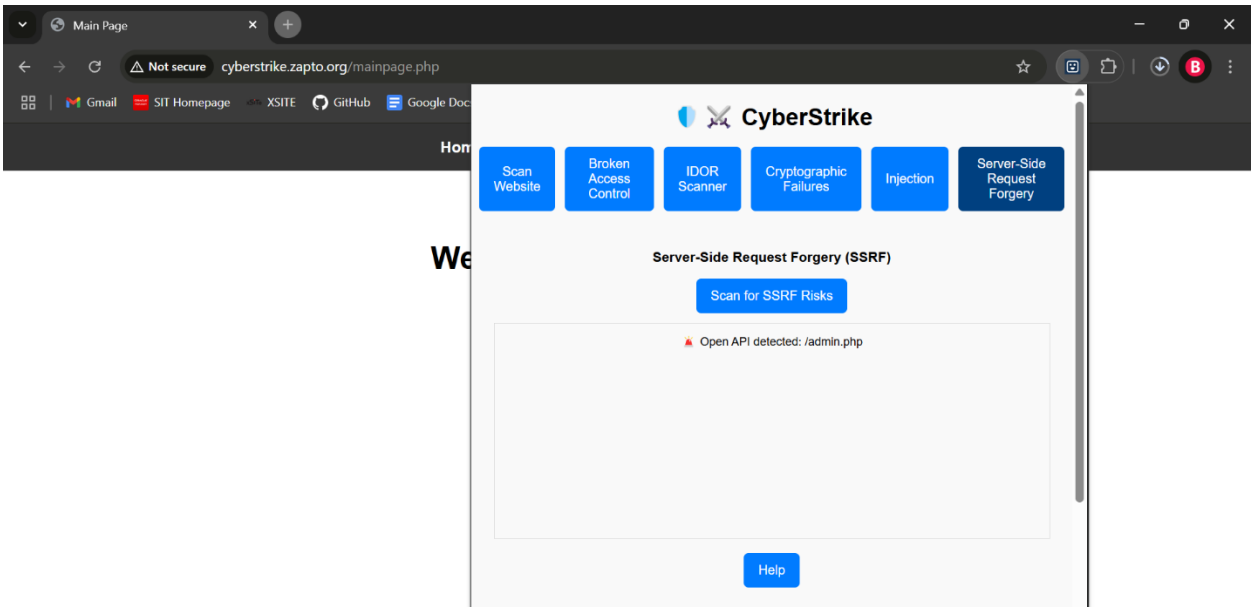


f) Server-Side Forgery

Upon clicking the “Scan for SSRF Risks” button, the scanner will check for open API endpoints that may be exposed and attempt to access internal services. E.g. localhost, internal networks and cloud metadata services.



After the scan is complete, vulnerable SSRFs will be listed below as seen in Figure xx.


Main Page
Not secure
cyberstrike.zapto.org/mainpage.php
Gmail
SIT Homepage
XSITE
GitHub
Google Doc
CyberStrike
Scan Website
Broken Access Control
IDOR Scanner
Cryptographic Failures
Injection
Server-Side Request Forgery
Server-Side Request Forgery (SSRF)
Scan for SSRF Risks
Open API detected: /admin.php
Help