REZV.ME

10 DECEMBER, 2025

VAPT REPORT

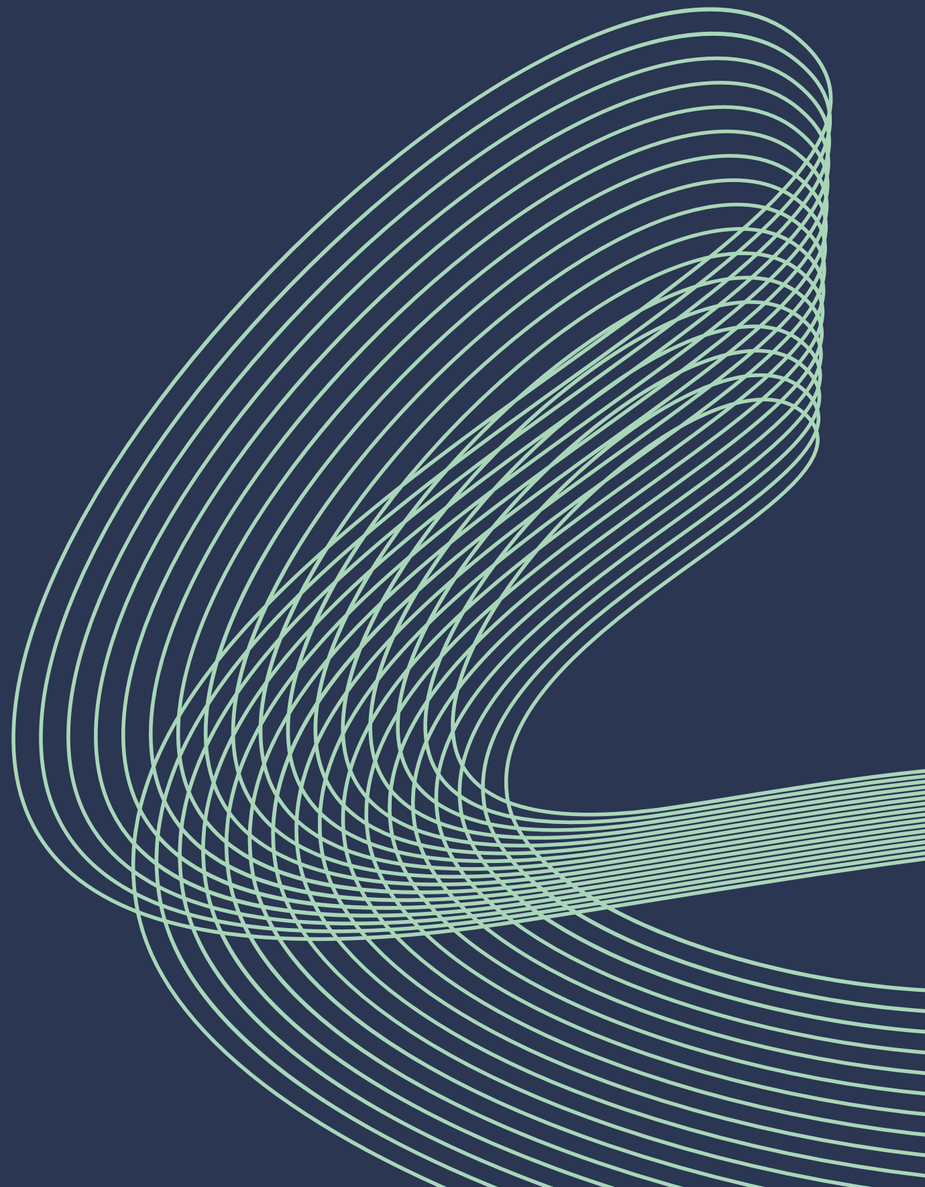
bWapp

PRESENTED TO

Md Mehedi Hasan Babu

PRESENTED BY

Md Rezvee Parvez



AGENDA

EXECUTIVE SUMMARY

This Vulnerability Assessment and Penetration Testing (VAPT) engagement was conducted for Bwapp with the objective of evaluating the security posture of its critical systems, applications, and supporting infrastructure. The assessment was designed to identify security weaknesses, determine the feasibility of potential attack vectors, and provide actionable recommendations to strengthen the organization’s overall cybersecurity resilience.

BACK TO AGENDA

CLIENT INFO

Client name : bWapp
Date : 09 December, 2025
Assesment Type : Web Application VAPT

Client : bWapp

BACK TO AGENDA

SCOPE OF ASSESSMENT

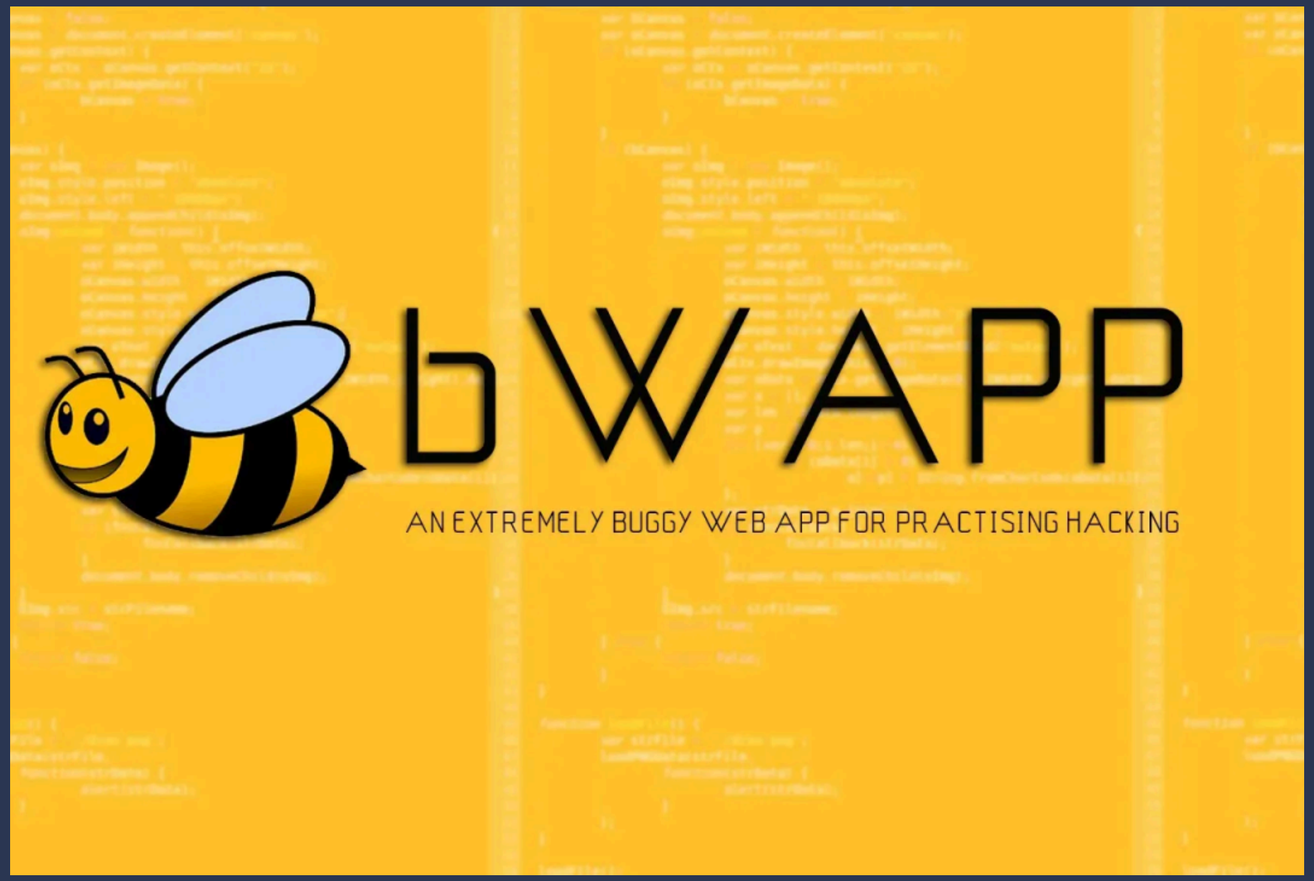
The assessment was limited to the following vulnerabilities:

- Php Code Injection
- Captcha Bypass
- IDOR-Secrect Change
- Directory Traversal
- Remote and Local file inclusion

BACK TO AGENDA

METHODOLOGY

- Nikto Vulnerablity Scan for the web Server
- Burpsuite for request Interception
- Manual exploitation
- Black-Box approach
- All findings are validated and doccumented



Client : bWapp

BACK TO AGENDA

NIKTO SCAN REPORT

Key Findings:

- Directory Indexing Enabled (High)

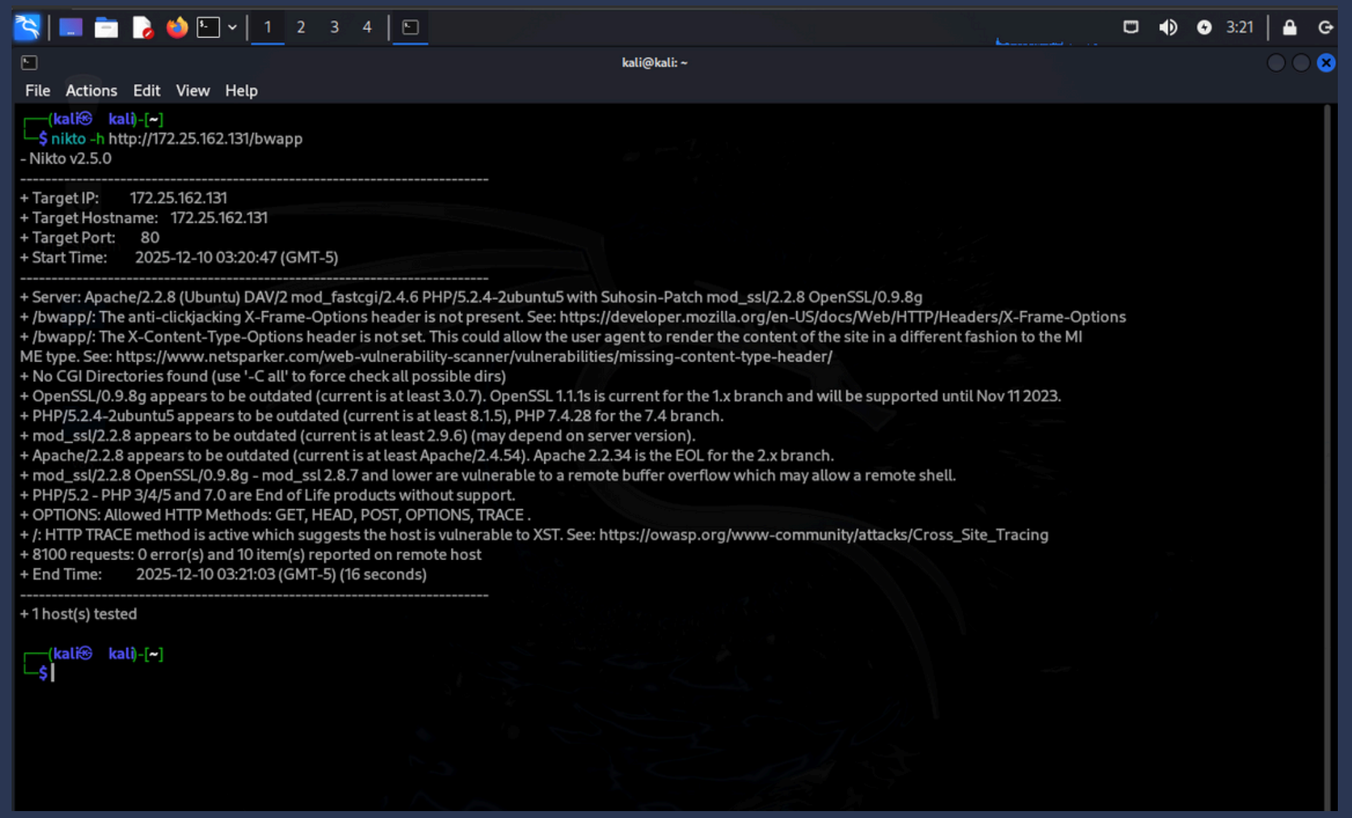
open directories: /passwords/logs/doccuments
Risk: Sensative files can be viewed
Fix: Disable directory listing

- Sensative files accessable (High)

config.inc - contains user and password
phpinfo.php- Gets the php information
Risk:Information disclosure
Fix: Remove or restrict these files

- Outdated Software

php5.2.4
Apache 2.2.8
OpenSSL 0.9.8
Risk: Contains known vulnerablities
Fix: Upgrade to latest version

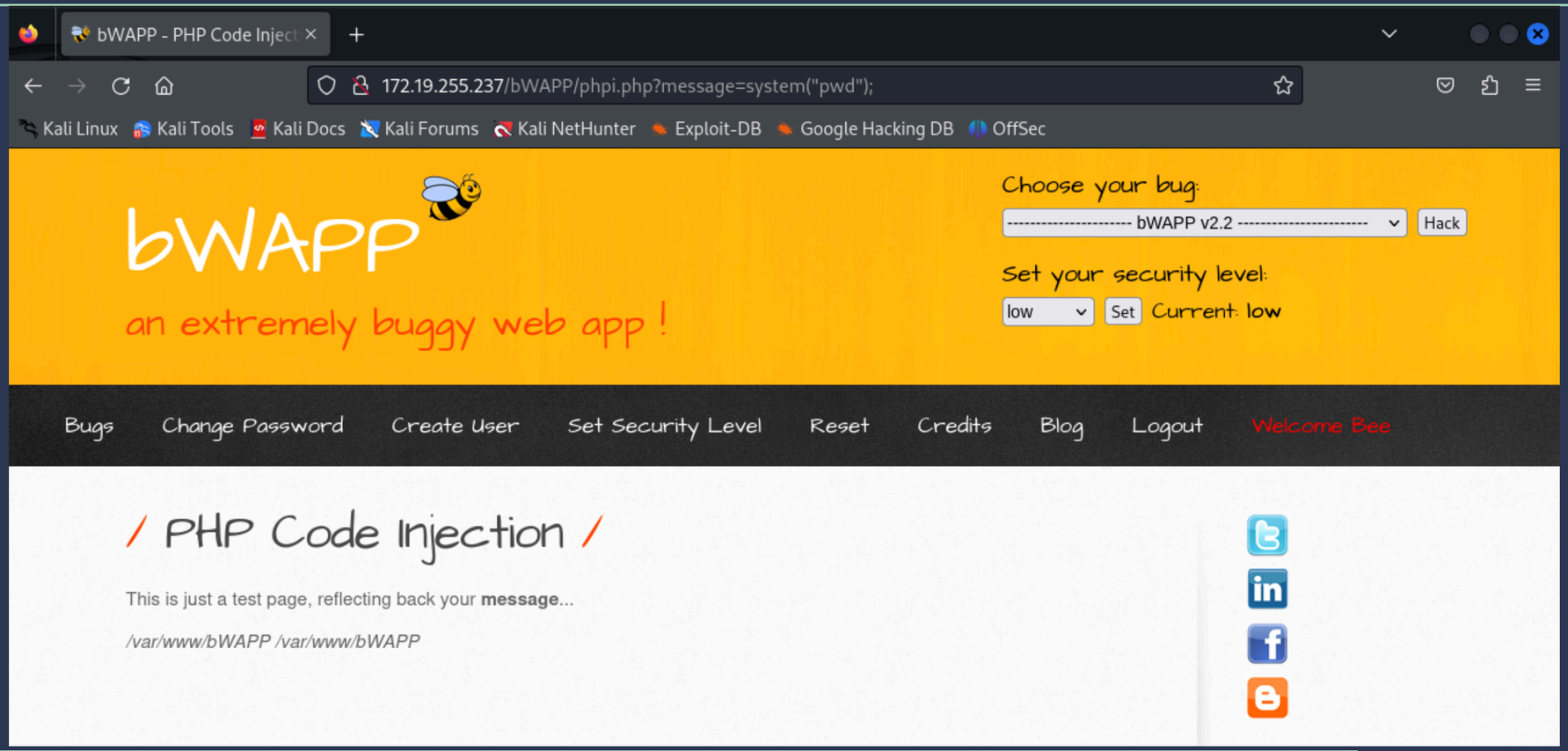


Target Information

- Target IP: 172.25.162.131
- Application : bWapp

PHP CODE INJECTION

PHP code injection is a security vulnerability where an attacker is able to insert and execute malicious PHP code on a server. This happens when a web application takes user-supplied input and processes it as PHP without proper validation or sanitization.



Vulnerablity : Php Code Injection

Description: The web application does not properly check or filter user input. Because of this, an attacker can inject OS commands into a PHP function (like system()), allowing them to execute commands on the server. This leads to Remote Code Execution (RCE).

PHP CODE INJECTION

Affected Assets:

- URL: 172.19.255.237/bWapp/phpi.php
- Vulnerable Parameter : System

Severity: Critical (CVSS:9.8)

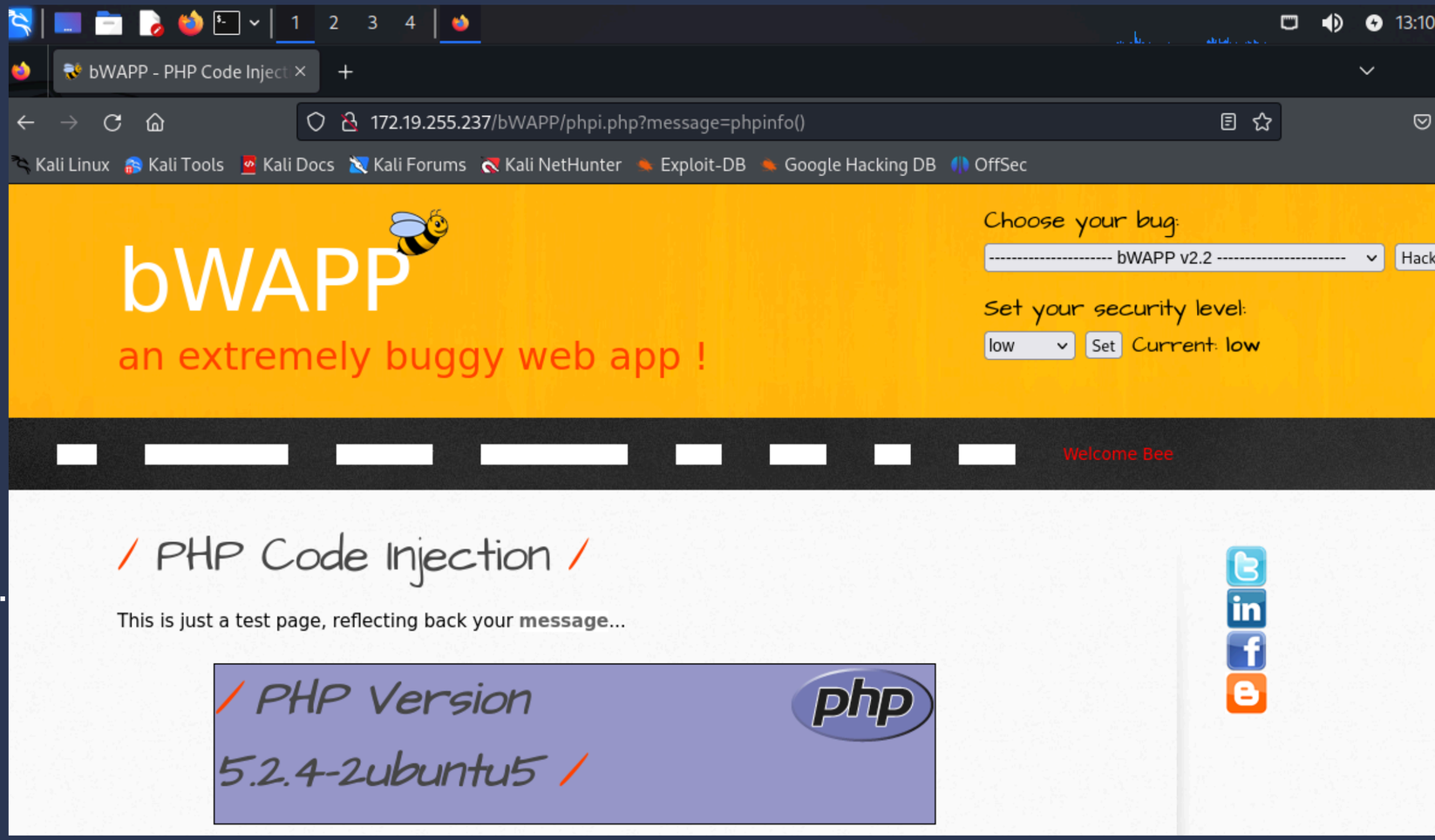
An attacker can gain full access to server

Steps to Reproduce (PoC)

- Open the vulnerable URL.

2. Inject a command into the system parameter.
3. Payload used: phpinfo().

The server’s php information pops up

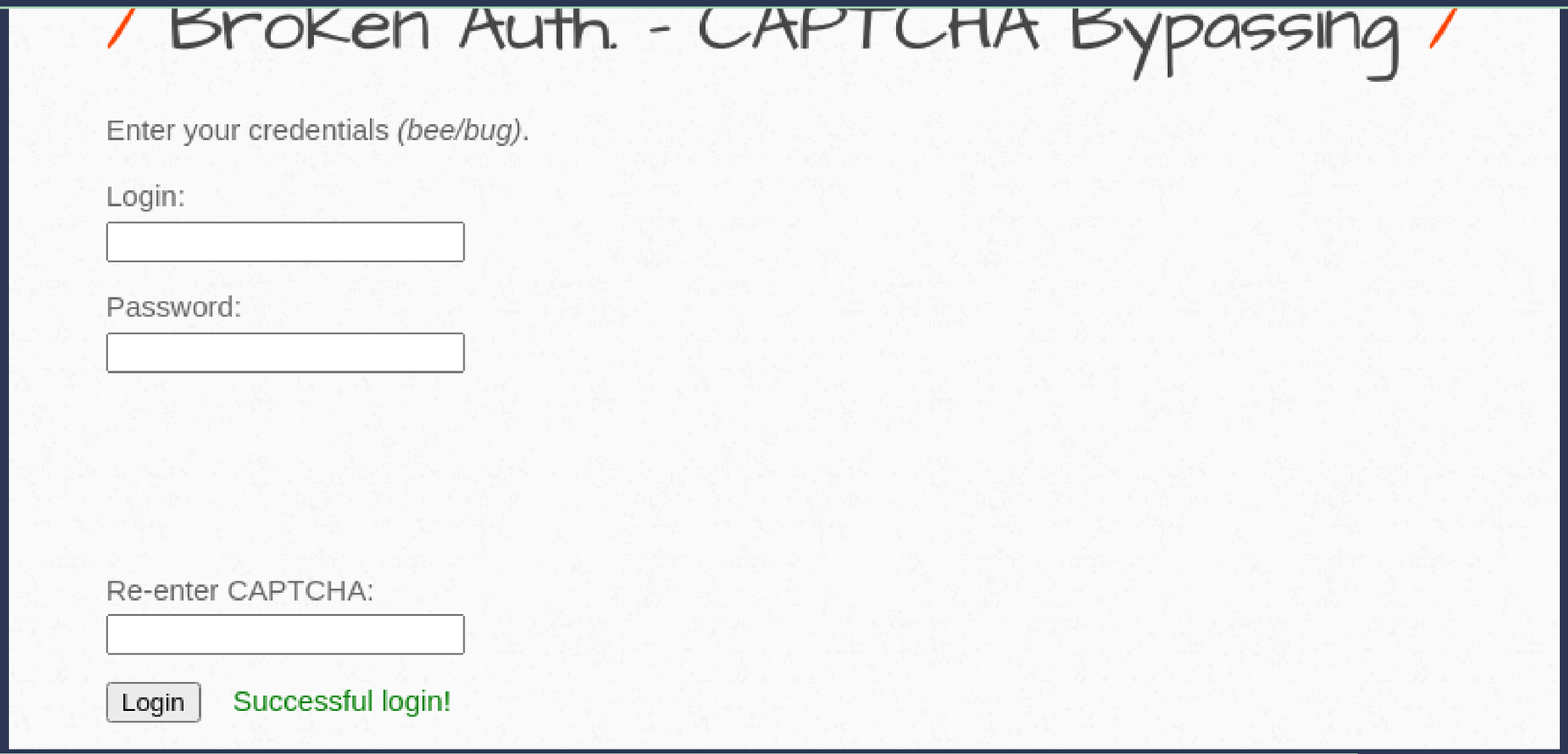


Remediation:

- Validate and sanitize all user inputs
- Avoid using dangerous PHP functions (system(), exec(), etc.)
- Disable risky functions in php.ini under disable_functions
- Use whitelisting when only specific commands or inputs are allowed

BROKEN AUTHENTICATION – CAPTCHA BYPASS

CAPTCHA bypass is when an attacker finds a way to defeat or avoid a CAPTCHA challenge that is meant to distinguish humans from bots.



Vulnerablity : Broken Authentication
Description: The loginpage uses a CAPTCHA for verification, but the server does not properly validate it. The CAPTCHA answer is not required during login, and the server accepts the request even if the captcha_user parameter is missing. This allows an attacker to log in without solving the CAPTCHA

BROKEN AUTHENTICATION – CAPTCHA BYPASS

CAPTCHA bypass is when an attacker finds a way to defeat or avoid a CAPTCHA challenge that is meant to distinguish humans from bots.

- **Impact**

Severity : High

CVSS Score : 9.4

- **Proof of Concept**

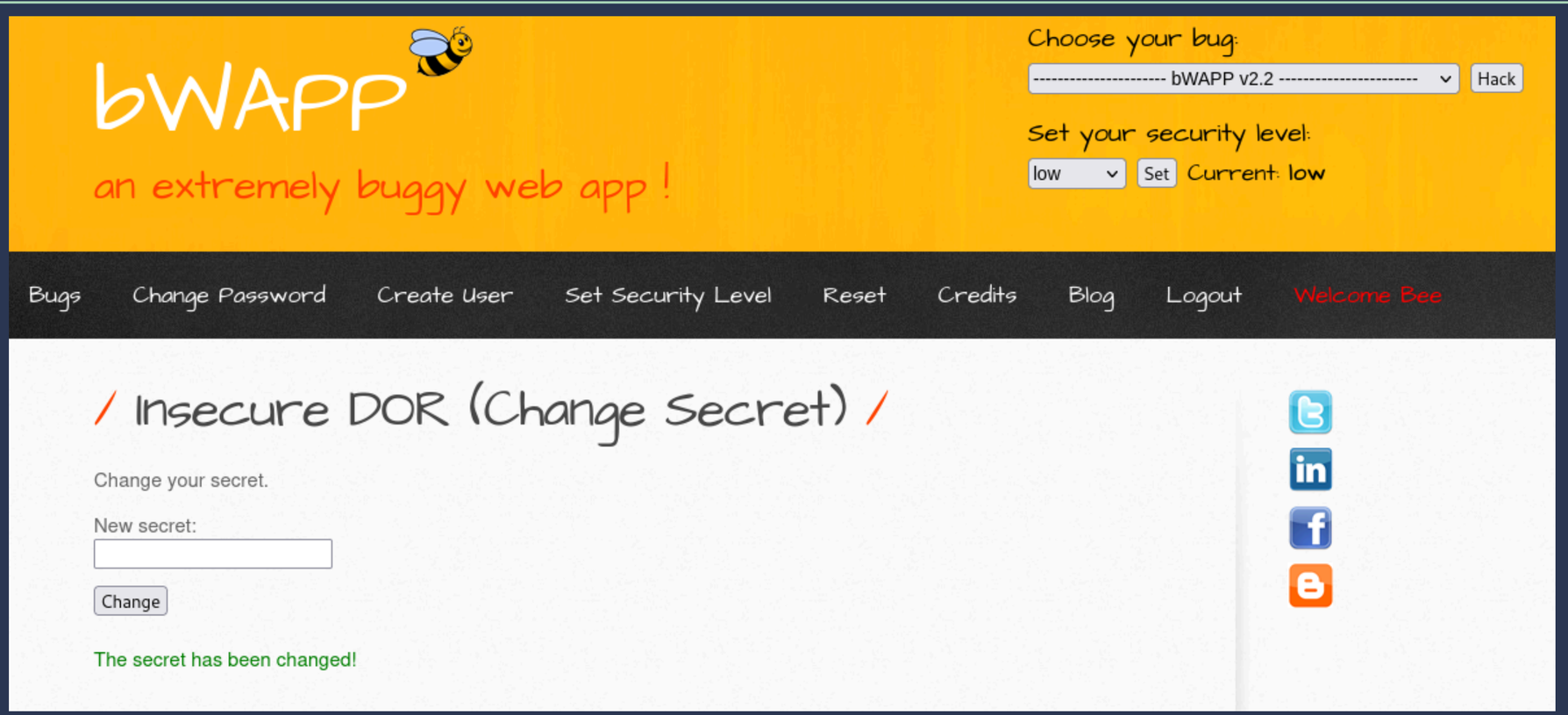
A login request with valid credentialsand a CAPTCHA value was captured.
Using Burp Suite, the CAPTCHA parameter was removed from the POST request:
log in= bee&password= bug&form=submit

- **Remediation**

Enforce server-side CAPTCHA validation. Reject login attempts if the CAPTCHA is missing or incorrect.
Store the CAPTCHA value in the session and validate it properly.
Validate all user input to prevent similar authentication bypass issues.

INSECURE DIRECT OBJECT REFERENCE (IDOR)

The application trusts user-controlled input for sensitive objects, so an attacker can change a parameter (like secret=123) and access or modify another user's data.



Vulnerablity : Insecure Direct Object Reference

Description: The application contains an IDOR vulnerability in the user secret-change functionality. When updating the secret phrase, the application relies directly on a user-controlled parameter login=bee in the POST request. The server does not verify whether the authenticated user is authorized to modify the secret of the user specified in this parameter.

INSECURE DIRECT OBJECT REFERENCE (IDOR)

The application trusts user-controlled input for sensitive objects, so an attacker can change a parameter (like secret=123) and access or modify another user's data.

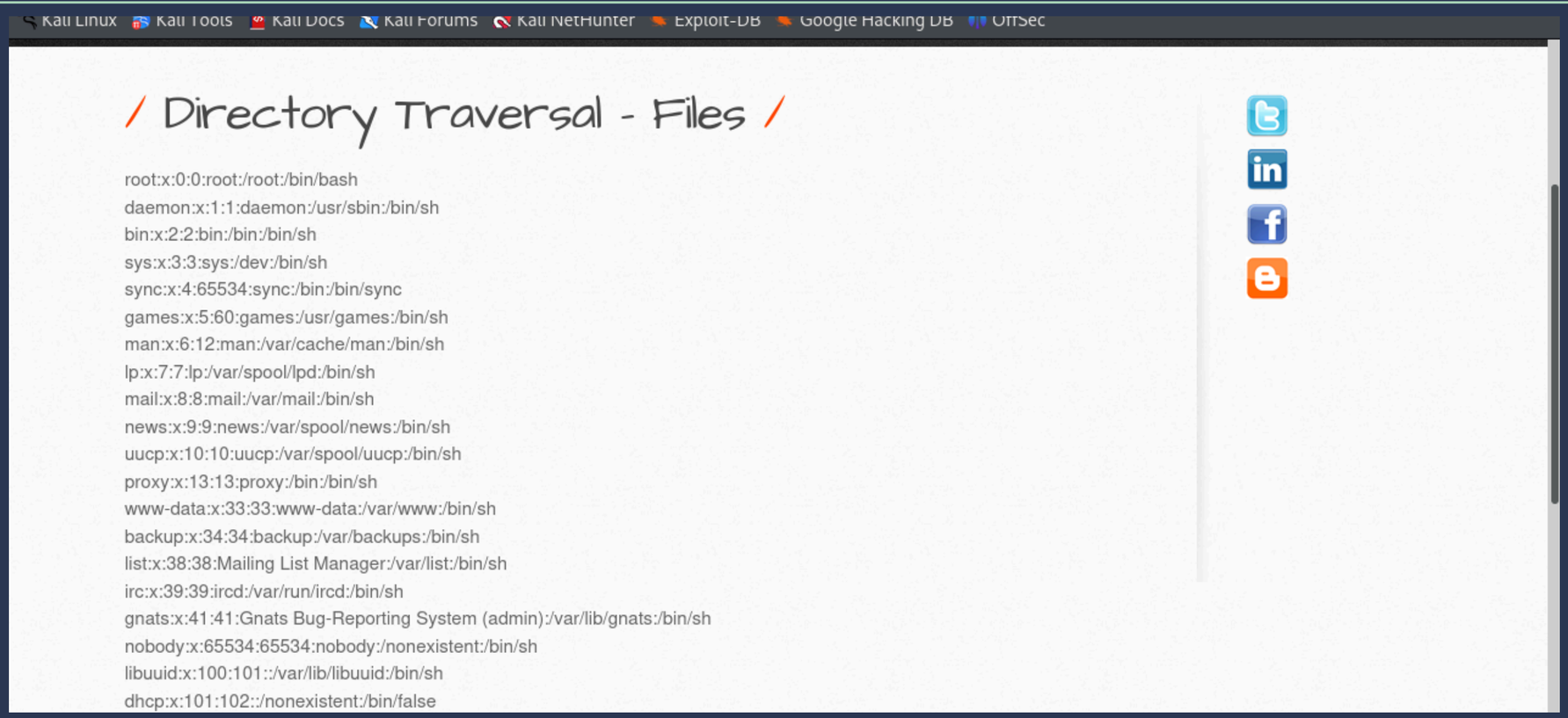
Impact
Severity : High
CVSS Score : 8.2
This vulnerability allows unauthorized modification of sensitive user information. An attacker can change another user's secret phrase, which may be used for account recovery or secondary authentication. This compromises both the integrity and confidentiality of user accounts.

Proof of Concept:
RequestCapture
Exploitation (IDOR)
Result

Remediation:
Enforce Server-Side Authorization Checks: Validate that the user ID in the request matches the authenticated user's ID from the session. Reject any mismatched or unauthorized requests.
Avoid Direct Object References:
Use indirect identifiers (e.g., mapped tokens) instead of exposing user IDs or login names directly in request parameters.
Apply Least Privilege:
Ensure user sessions only have permissions necessary for their own account and cannot modify data belonging to other users.

DIRECTORY TRAVERSAL

Directory traversal is a vulnerability where an attacker manipulates file-path input to access files and folders outside the intended directory.



Vulnerablity : Directory Traversal

Description: The applicationis vulnerable to a Directory Traversal attack. The page parameter is directly used to build the file path without proper validation. Because of this, an attacker can use ../ sequences to access files outside the web root directory.

DIRECTORY TRAVERSAL

Impact
Severity : High
CVSS Score : 7.5
A successful attack allows an attacker to read arbitrary system files, which can leads to Information Disclosure, System Enumeration, Exposure of sensitive configuration files such as database credentials

Proof of Concept:

- Exploitation

The attacker used atraversal path to access a sensitive system file

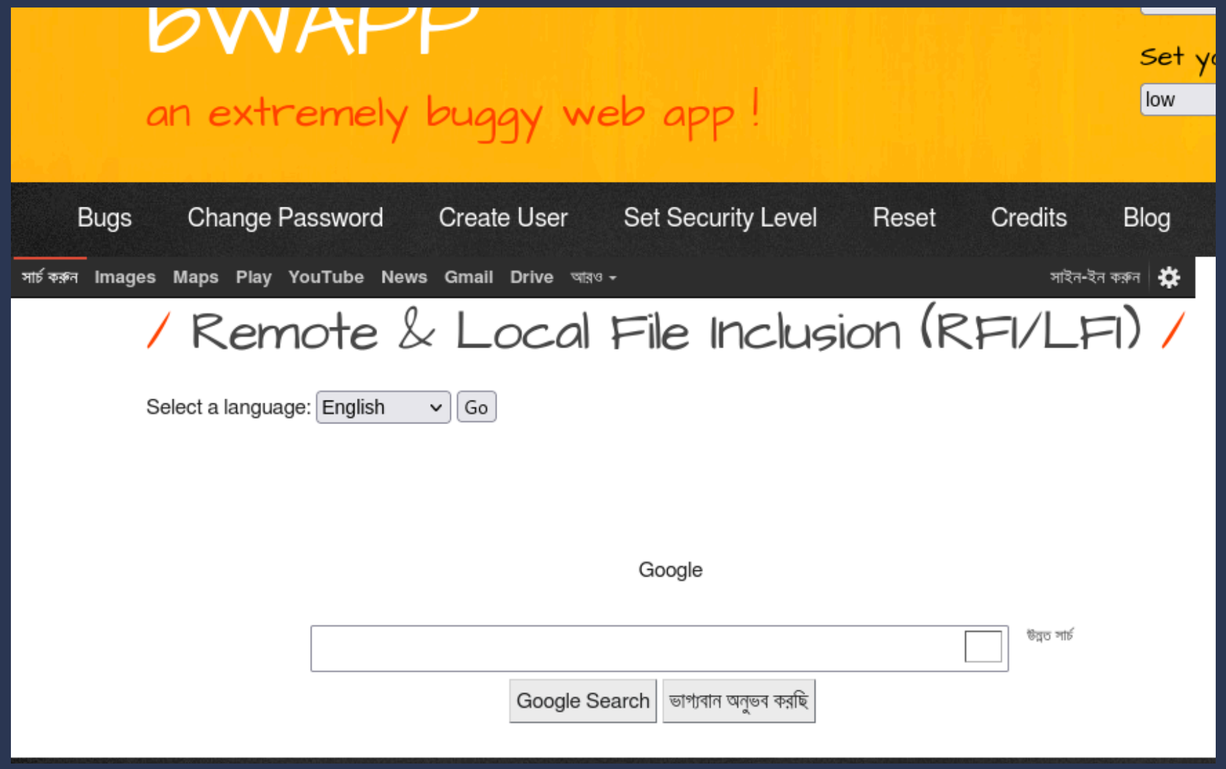
- Result

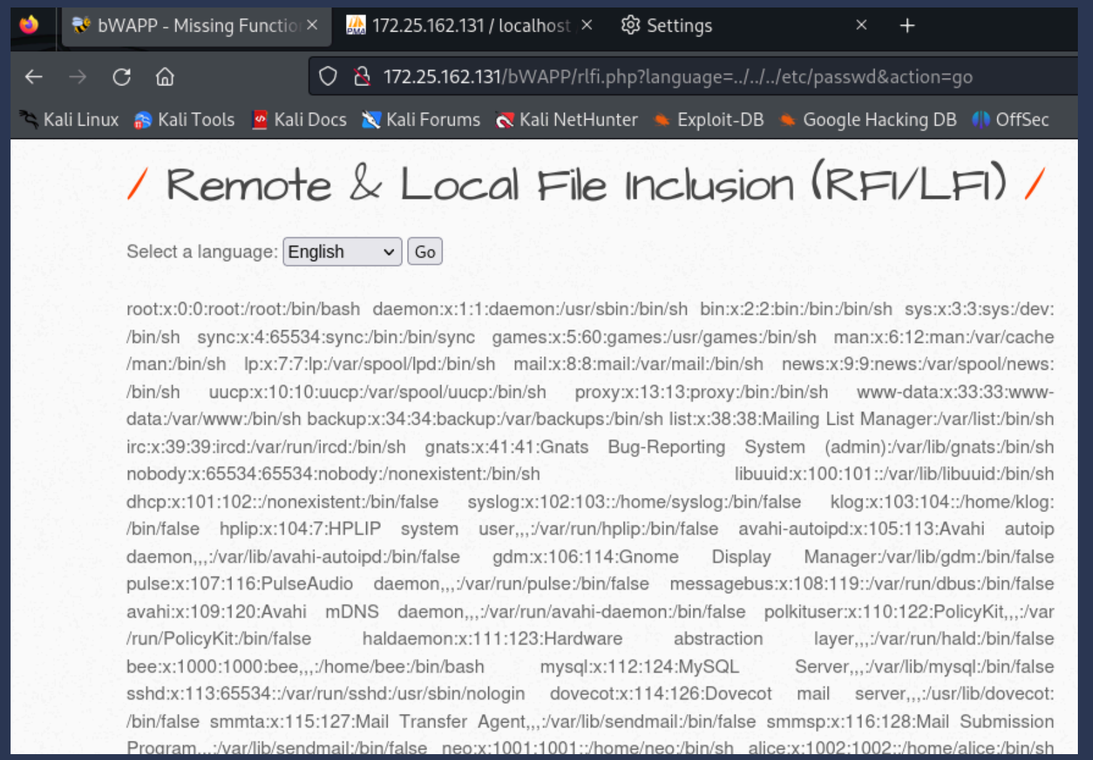
The server returned the contents

Remediation:
Validate Input
Canonicalize Paths
Use Whitelists
Least Priviledge

REMOTE AND LOCAL FILE INCLUSION

The attacker tricks the application into loading or executing files already on the server, such as logs or configuration files.





Vulnerablity : Directory Traversal

Description: The applicationis vulnerable to a Directory Traversal attack. The page parameter is directly used to build the file path without proper validation. Because of this, an attacker can use ../ sequences to access files outside the web root directory.

REMOTE AND LOCAL FILE INCLUSION

Impact
Severity : Critical
CVSS Score : 9.0
A successful attack allows an attacker to read arbitrary system files, which can leads to Information Disclosure, System Enumeration, Exposure of sensitive configuration files such as database credentials

Proof of Concept:

- RFI Example

The server returned Google.com

- LFI Example

The server returned the contents of the server

Remediation:
Sanitize and validate all input (remove ../, http://, etc.).
Implement strict allowlists for files and paths.
Disable allow_url_fopen and allow_url_include in PHP.
Use secure frameworks or templating systems instead of direct includes.
Set proper file permissions to limit accessible directories.
Keep server and PHP versions updated with security patches.

THANK YOU !

FOR INQUIRIES, CONTACT US.

Email

rezvx@proton.me

Social Media

www.linkedin.com/in/rezvx

www.rezv.me