Future Interns - Cyber Security Task 1

Web Application Security Testing

Task:

- Conduct security testing on a sample web application to identify vulnerabilities like SQL injection, XSS, and authentication flaws

Key Features to Include

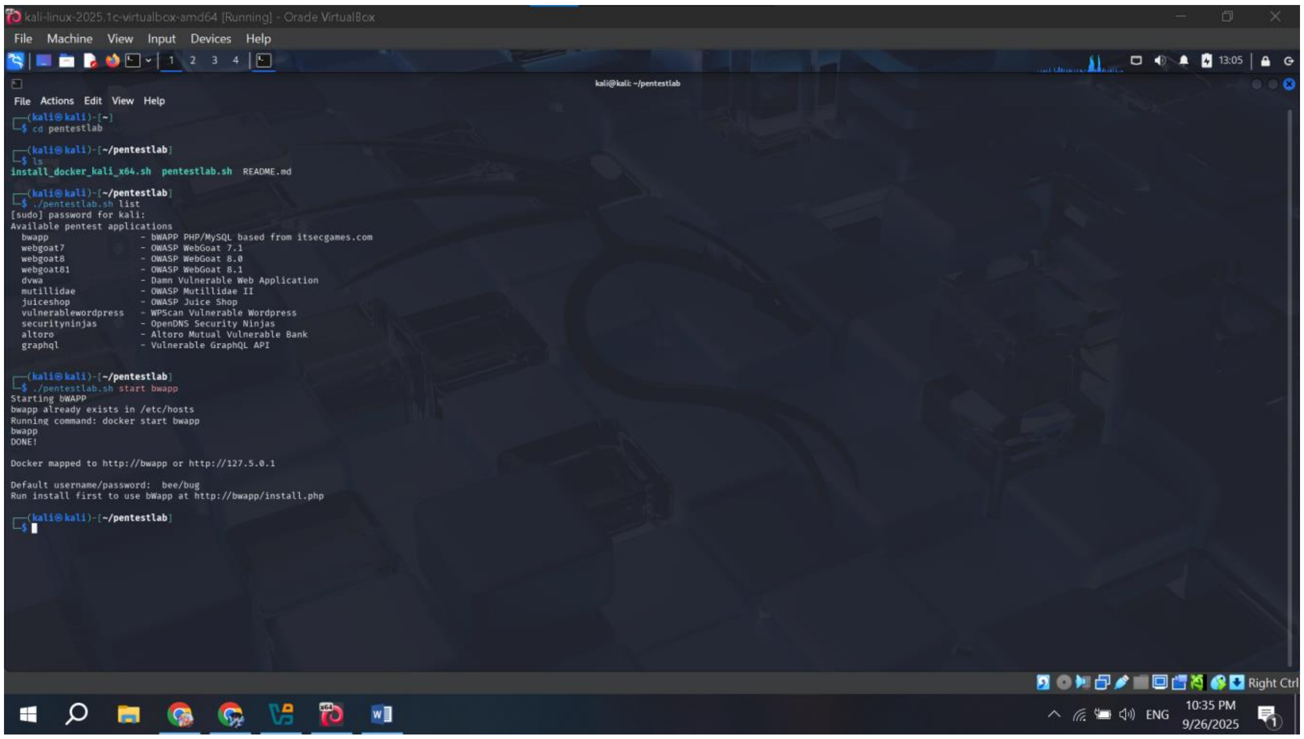
- At least 3–5 real vulnerabilities found and documented
- Screenshots of attack vectors and scanner outputs
- Mitigation steps for each vulnerability
- OWASP Top 10 Checklist mapping
- A polished Security Report (PDF) that simulates client work

1. Set up and explore a test web app (bWapp)

Target website: bWAPP, Link: http://www.itsecgames.com/

Setting up the target website.

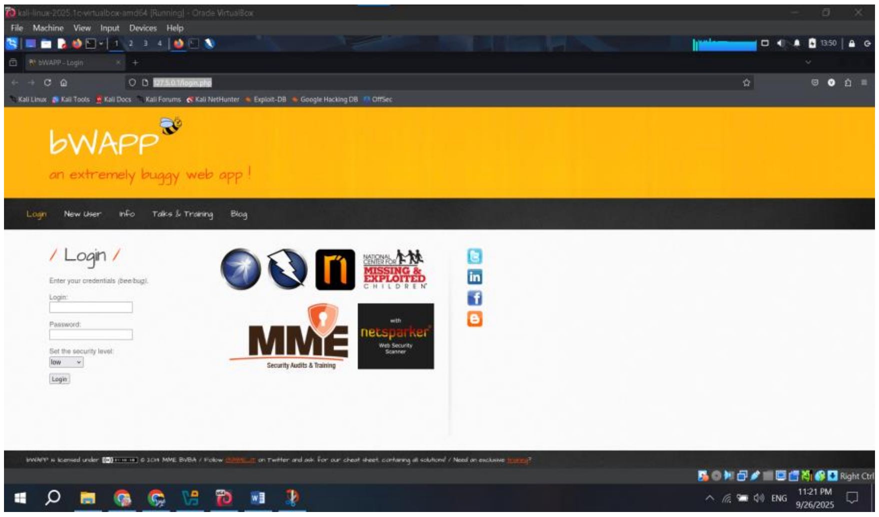
- Copied the https code from github https://github.com/eystsen/pentestlab.git
- Cloned this into my terminal
- Now this belongs to my directory. (/Desktop/pentestlab)
- Before starting anything, I installed docker.io



```
┌──(kali㉿kali)-[~]
└─$ cd pentestlab

┌──(kali㉿kali)-[~/pentestlab]
└─$ ls
install_docker_kali_x64.sh  pentestlab.sh  README.md

┌──(kali㉿kali)-[~/pentestlab]
└─$ ./pentestlab.sh list
[sudo] password for kali:
Available pentest applications
  bwapp                  - bWAPP PHP/MySQL based from itsecgames.com
  webgoat7               - OWASP WebGoat 7.1
  webgoat8               - OWASP WebGoat 8.0
  webgoat81              - OWASP WebGoat 8.1
  dvwa                   - Damn Vulnerable Web Application
  mutillidae             - OWASP Mutillidae II
  juiceshop              - OWASP Juice Shop
  vulnerablewordpress    - WPScan Vulnerable Wordpress
  securityninjas         - OpenDNS Security Ninjas
  altoro                 - Altoro Mutual Vulnerable Bank
  graphql                - Vulnerable GraphQL API


┌──(kali㉿kali)-[~/pentestlab]
└─$ ./pentestlab.sh start bwapp
Starting bWAPP
bwapp already exists in /etc/hosts
Running command: docker start bwapp
bwapp
DONE!

Docker mapped to http://bwapp or http://127.5.0.1

Default username/password:  bee/bug
Run install first to use bWapp at http://bwapp/install.php

┌──(kali㉿kali)-[~/pentestlab]
└─$ 
```

After logging in with the default credentials (bee/bug), I explored the application before starting Task 2 (running automated scans).

- Authentication: The application requires login; session and cookie handling will be important areas to test.
- Bug selection: The dashboard lets you pick different vulnerabilities from a dropdown (e.g., SQL injection, XSS, CSRF). There's also a Security Level control (Low / Medium / High) that adjusts how easy or difficult the flaws are to exploit.
- Navigation & functionality: The UI exposes several functional pages — profile pages, message boards, input forms, and file upload features — all useful test surfaces.
- Tech clues: The app sometimes displays PHP errors, indicating a PHP + MySQL backend.



127.5.0.1/login.php

/ Login /

Enter your credentials (bee/bug).

Login:

Password:

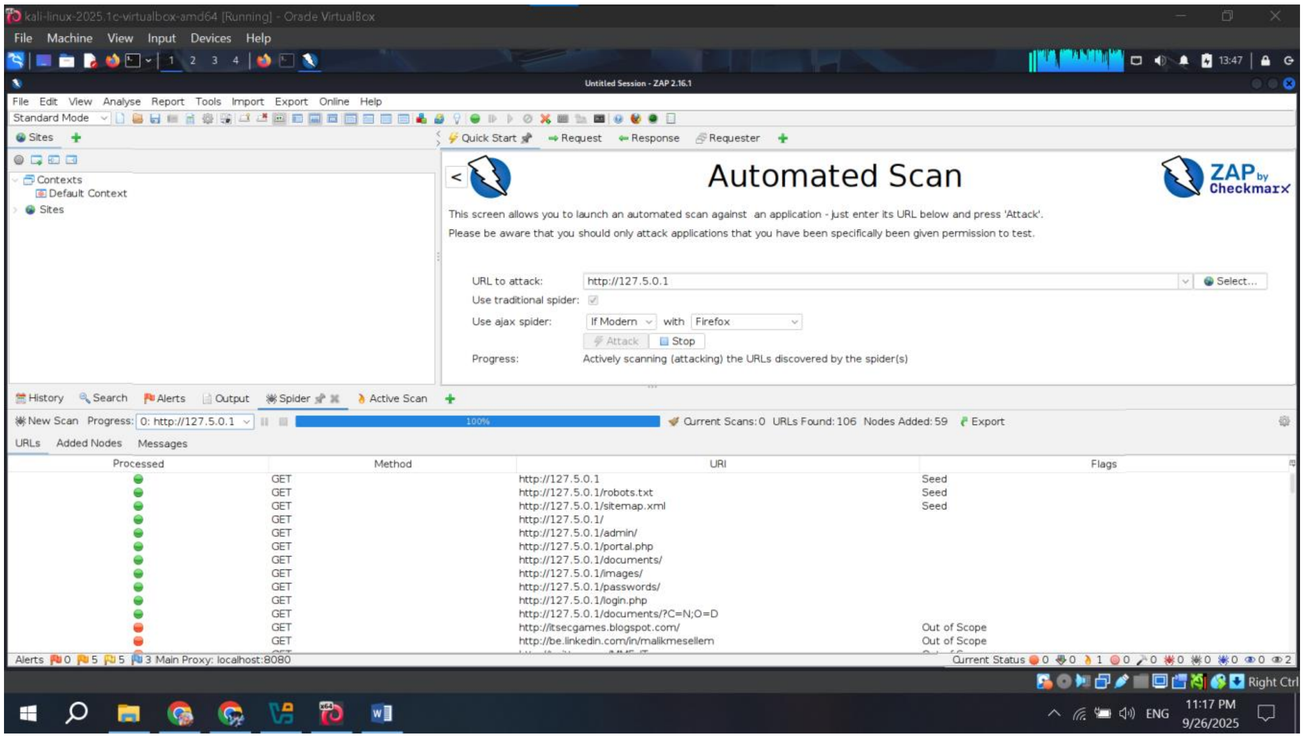
Set the security level: low

- Potential attack vectors:
    - The login form (possible SQL injection)
    - User-controlled input fields (search, messages)
    - File upload functionality
    - Cookie and session handling (possible manipulation or fixation)

2. Test for common vulnerabilities like SQL injection, XSS, and CSRF

How I Used OWASP ZAP for Automated Scanning

I used OWASP ZAP (version 2.16.1) on Kalin Linux to scan the bWAPP application by entering its local URL into the "URL to attack" field and running the automated scan by clicking the "attack" button after I entered the target website URL. Zap first crawled the site to discover available pages and then actively tested those pages for vulnerabilities, showing the output in the alert tab with certain ratings. I confirmed the findings by copying the suspicious URLs (such as /phpinfo.php) into my browser and even checking the replaying request in ZAP's Requester tab to see if the issue could be exploited.


kali-linux-2025.1c-virtualbox-amd64 [Running] - Oracle VirtualBox
File Machine View Input Devices Help
Untitled Session - ZAP 2.16.1
File Edit View Analyse Report Tools Import Export Online Help
Standard Mode
Sites
Contexts
Default Context
Sites
Quick Start Request Response Requester
History Search Alerts Output Spider Active Scan
URLs Added Nodes Messages
Right Ctrl
11:17 PM
9/26/2025

Automated Scan

This screen allows you to launch an automated scan against an application - just enter its URL below and press 'Attack'.
Please be aware that you should only attack applications that you have been specifically been given permission to test.

URL to attack: http://127.5.0.1
Use traditional spider: ✓
Use ajax spider: If Modern with Firefox
Attack Stop
Progress: Actively scanning (attacking) the URLs discovered by the spider(s)

New Scan Progress: 0: http://127.5.0.1 100% Current Scans: 0 URLs Found: 106 Nodes Added: 59 Export

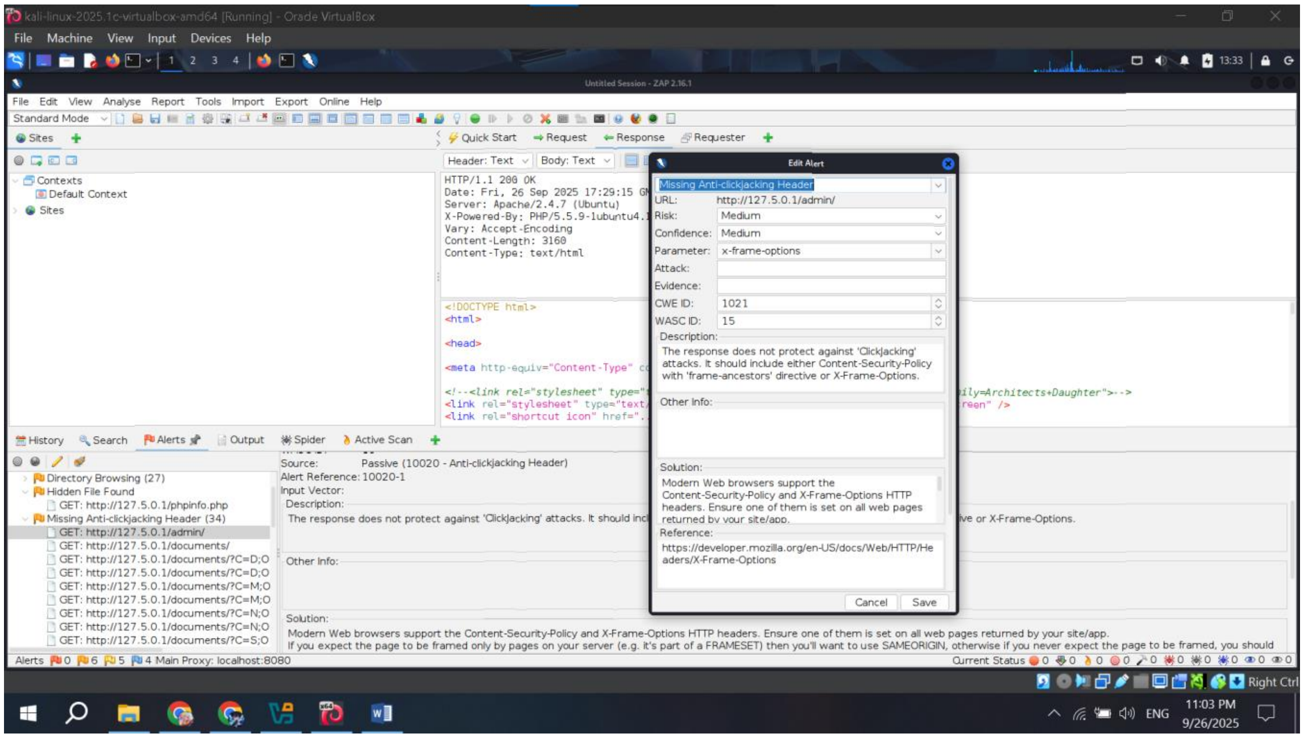
Processed	Method	URI	Flags
●	GET	http://127.5.0.1	Seed
●	GET	http://127.5.0.1/robots.txt	Seed
●	GET	http://127.5.0.1/sitemap.xml	Seed
●	GET	http://127.5.0.1/	
●	GET	http://127.5.0.1/admin/	
●	GET	http://127.5.0.1/portal.php	
●	GET	http://127.5.0.1/documents/	
●	GET	http://127.5.0.1/images/	
●	GET	http://127.5.0.1/passwords/	
●	GET	http://127.5.0.1/login.php	
●	GET	http://127.5.0.1/documents/?C=N;O=D	
●	GET	http://itsecgames.blogspot.com/	Out of Scope
●	GET	http://be.linkedin.com/in/malikmesellem	Out of Scope

Alerts 0 5 5 3 Main Proxy: localhost:8080 Current Status 0 0 1 0 0 0 0 0 0 2

- OWASP ZAP is currently scanning the bWAPP site to detect potential weaknesses. The screenshot below shows the completed scan results. ZAP successfully flagged a number of issues, but not every finding is guaranteed to be exploitable. To verify each alert, open the Alerts tab and double-click the item to inspect the request, response, and evidence before classifying it as a true positive or false positive.

Vulnerability 1: Missing Anti-Clickjacking Header

- Risk Rating: Medium
- Description: The application responses lack security headers such as X-Frame-Options or the Content-Security-Policy (CSP) directive frame-ancestors. Without these, the site is vulnerable to clickjacking attacks, where attackers can embed the site in hidden frames and trick users into performing unintended actions.
- Evidence: Flagged by OWASP ZAP as *"Missing Anti-clickjacking Header"* for URL: http://127.5.0.1/admin/.



```
HTTP/1.1 200 OK
Date: Fri, 26 Sep 2025 17:29:15 GM
Server: Apache/2.4.7 (Ubuntu)
X-Powered-By: PHP/5.5.9-1ubuntu4.1
Vary: Accept-Encoding
Content-Length: 3160
Content-Type: text/html
```

Edit Alert

Missing Anti-clickjacking Header
URL: http://127.5.0.1/admin/
Risk: Medium
Confidence: Medium
Parameter: x-frame-options
Attack:
Evidence:
CWE ID: 1021
WASC ID: 15
Description: The response does not protect against 'ClickJacking' attacks. It should include either Content-Security-Policy with 'frame-ancestors' directive or X-Frame-Options.
Other Info:
Solution: Modern Web browsers support the Content-Security-Policy and X-Frame-Options HTTP headers. Ensure one of them is set on all web pages returned by your site/app.
Reference: https://developer.mozilla.org/en-US/docs/Web/HTTP/Headers/X-Frame-Options

```
Directory Browsing (27)
Hidden File Found
    GET: http://127.5.0.1/phpinfo.php
Missing Anti-clickjacking Header (34)
    GET: http://127.5.0.1/admin/
    GET: http://127.5.0.1/documents/
    GET: http://127.5.0.1/documents/?C=D;O
    GET: http://127.5.0.1/documents/?C=D;O
    GET: http://127.5.0.1/documents/?C=M;O
    GET: http://127.5.0.1/documents/?C=M;O
    GET: http://127.5.0.1/documents/?C=N;O
    GET: http://127.5.0.1/documents/?C=N;O
    GET: http://127.5.0.1/documents/?C=S;O
```

Source: Passive (10020 - Anti-clickjacking Header)
Alert Reference: 10020-1
Input Vector:
Description: The response does not protect against 'ClickJacking' attacks. It should inc… …ive or X-Frame-Options.
Other Info:
Solution: Modern Web browsers support the Content-Security-Policy and X-Frame-Options HTTP headers. Ensure one of them is set on all web pages returned by your site/app.
If you expect the page to be framed only by pages on your server (e.g. it's part of a FRAMESET) then you'll want to use SAMEORIGIN, otherwise if you never expect the page to be framed, you should

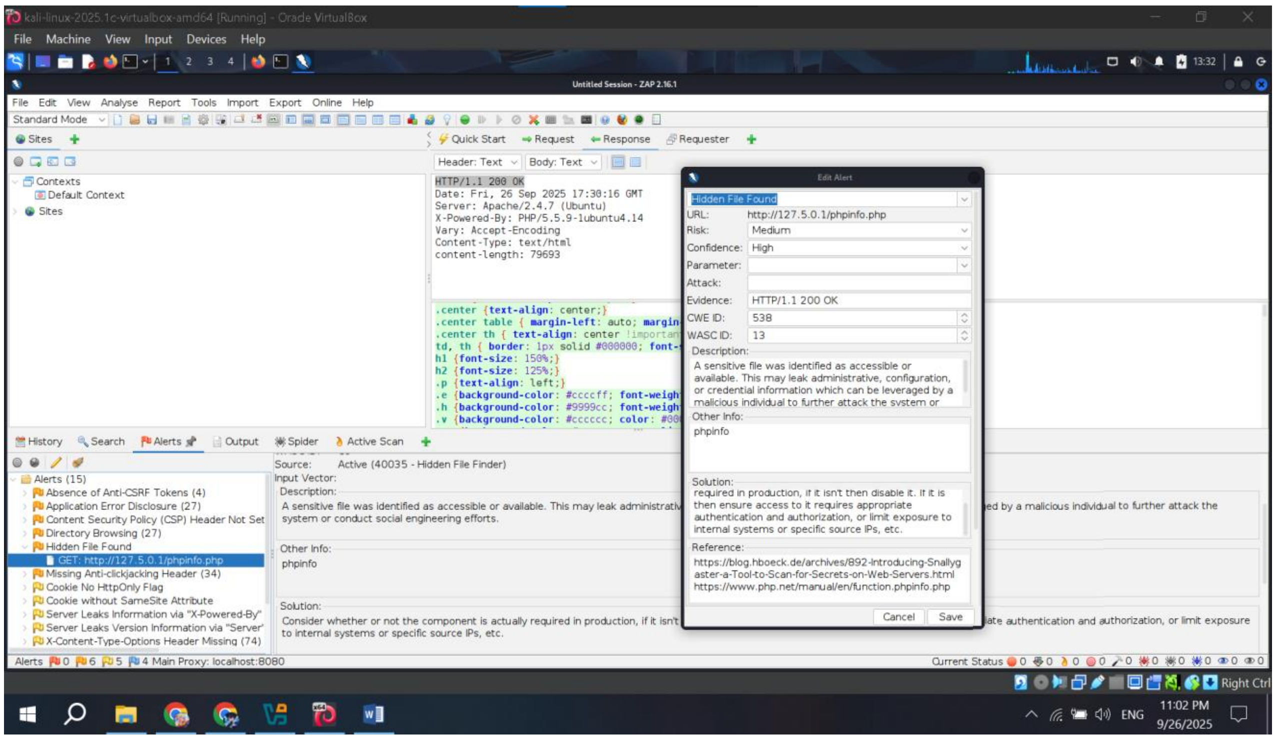
Alerts 0 6 5 4 Main Proxy: localhost:8080

- Impact: Attackers can load the site inside an invisible iframe and trick users into clicking hidden buttons (e.g., transferring money, changing passwords), resulting in unauthorized actions.
- Mitigation:
  - Implement one of the following headers:
    - X-Frame-Options: DENY or SAMEORIGIN
    - Content-Security-Policy: frame-ancestors 'none'
  - Apply headers consistently across all pages.

    - Validate deployment with tools like OWASP ZAP to ensure security headers are properly set.

Vulnerability 2: Hidden File Disclosure - /phpinfo.php

- Risk Rating: Medium
- Description: The phpinfo.php file is publicly accessible and reveals sensitive server details — PHP version, loaded modules, environment variables, and configuration settings. An attacker could use this information to discover exploitable software versions or other weaknesses, or to craft targeted social-engineering or technical attacks against the system.
- Evidence: A screenshot of the vulnerable page was captured during the scan and is included with this report .The image clearly displays the phpinfo() output — including the PHP version and configuration details — which corroborates the exposure of phpinfo.php.


kali-linux-2025.1c-virtualbox-amd64 [Running] - Oracle VirtualBox
File Machine View Input Devices Help
Untitled Session - ZAP 2.16.1
File Edit View Analyse Report Tools Import Export Online Help
Standard Mode
Sites
Quick Start Request Response Requester
Header: Text Body: Text
Contexts
Default Context
Sites
History Search Alerts Output Spider Active Scan
Edit Alert
Cancel Save
11:02 PM
9/26/2025

```
HTTP/1.1 200 OK
Date: Fri, 26 Sep 2025 17:30:16 GMT
Server: Apache/2.4.7 (Ubuntu)
X-Powered-By: PHP/5.5.9-1ubuntu4.14
Vary: Accept-Encoding
Content-Type: text/html
content-length: 79693
```

```
.center {text-align: center;}
.center table { margin-left: auto; margin
.center th { text-align: center !importan
td, th { border: 1px solid #000000; font-
h1 {font-size: 150%;}
h2 {font-size: 125%;}
.p {text-align: left;}
.e {background-color: #ccccff; font-weigh
.h {background-color: #9999cc; font-weigh
.v {background-color: #cccccc; color: #00
```

Alerts (15)
- Absence of Anti-CSRF Tokens (4)
- Application Error Disclosure (27)
- Content Security Policy (CSP) Header Not Set
- Directory Browsing (27)
- Hidden File Found
  - GET: http://127.5.0.1/phpinfo.php
- Missing Anti-clickjacking Header (34)
- Cookie No HttpOnly Flag
- Cookie without SameSite Attribute
- Server Leaks Information via "X-Powered-By"
- Server Leaks Version Information via "Server"
- X-Content-Type-Options Header Missing (74)

Source: Active (40035 - Hidden File Finder)
Input Vector:
Description:
A sensitive file was identified as accessible or available. This may leak administrativ ed by a malicious individual to further attack the system or conduct social engineering efforts.
Other Info:
phpinfo
Solution:
Consider whether or not the component is actually required in production, if it isn't ate authentication and authorization, or limit exposure to internal systems or specific source IPs, etc.

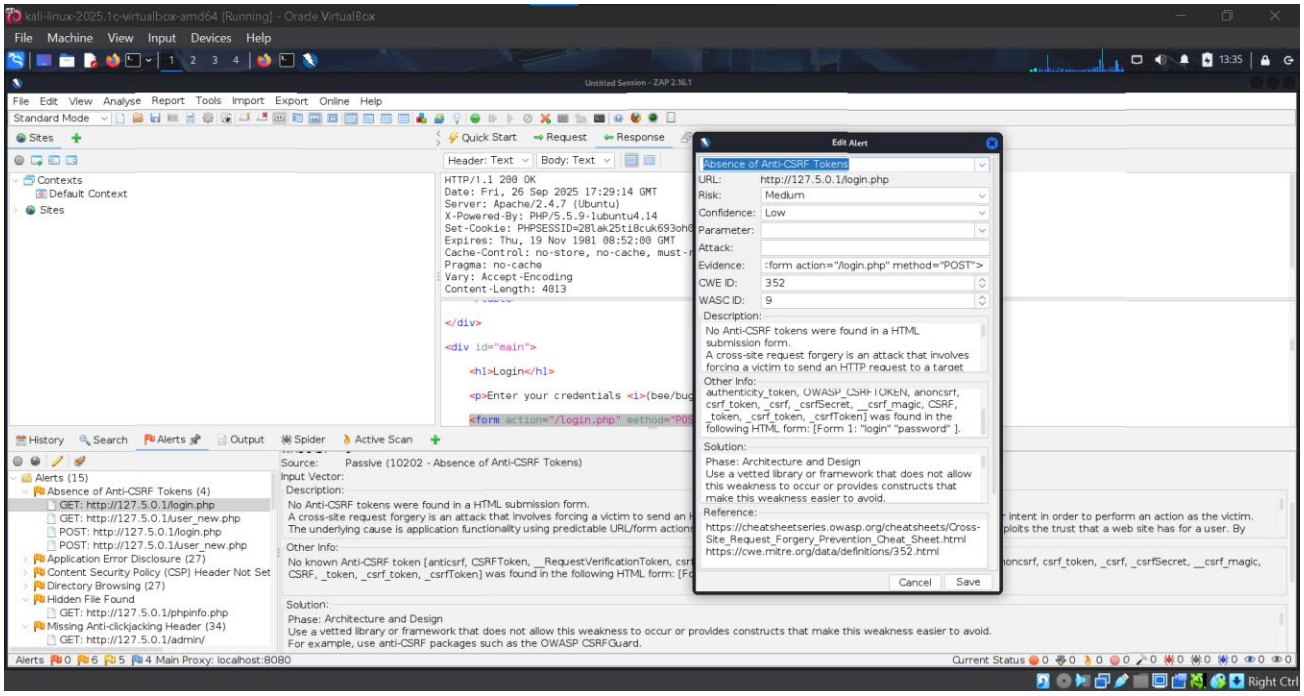
Alerts 0 6 5 4 Main Proxy: localhost:8080

Edit Alert:
Hidden File Found
URL: http://127.5.0.1/phpinfo.php
Risk: Medium
Confidence: High
Parameter:
Attack:
Evidence: HTTP/1.1 200 OK
CWE ID: 538
WASC ID: 13
Description:
A sensitive file was identified as accessible or available. This may leak administrative, configuration, or credential information which can be leveraged by a malicious individual to further attack the system or
Other Info:
phpinfo
Solution:
required in production, if it isn't then disable it. If it is then ensure access to it requires appropriate authentication and authorization, or limit exposure to internal systems or specific source IPs, etc.
Reference:
https://blog.hboeck.de/archives/892-Introducing-Snallygaster-a-Tool-to-Scan-for-Secrets-on-Web-Servers.html
https://www.php.net/manual/en/function.phpinfo.php

- Impact: Attackers can use this piece of information to identify any vulnerabilities in the PHP version system.
- Mitigation: Check if the component is actually required in the production, if it is not then make sure to disable it! If it is then make sure access to this php version requires authentication and authorization.

Vulnerability 3: Reflected Cross-Site Scripting (XSS)

- Risk: High
- Description: The application reflects user-supplied input directly into the web page without applying proper validation or output encoding. This allows attackers to inject malicious JavaScript code that is executed in the victim's browser when they visit the crafted link.
- Evidence: In the screenshot, the test payload <script>alert('XSS')</script> is returned in the application's response, proving the input is not sanitized.


kali-linux-2025.1c-virtualbox-amd64 [Running] - Oracle VirtualBox
File Machine View Input Devices Help
Untitled Session - ZAP 2.16.1
File Edit View Analyse Report Tools Import Export Online Help
Standard Mode
Sites
Contexts
Default Context
Sites
Quick Start Request Response
Header: Text Body: Text
History Search Alerts Output Spider Active Scan
Edit Alert
Cancel Save
Right Ctrl

```
HTTP/1.1 200 OK
Date: Fri, 26 Sep 2025 17:29:14 GMT
Server: Apache/2.4.7 (Ubuntu)
X-Powered-By: PHP/5.5.9-1ubuntu4.14
Set-Cookie: PHPSESSID=28lak25ti8cuk693oh0
Expires: Thu, 19 Nov 1981 08:52:00 GMT
Cache-Control: no-store, no-cache, must-r
Pragma: no-cache
Vary: Accept-Encoding
Content-Length: 4013
```

```
</div>

<div id="main">

    <h1>Login</h1>

    <p>Enter your credentials <i>(bee/bug

    <form action="/login.php" method="POS
```

Absence of Anti-CSRF Tokens
URL: http://127.5.0.1/login.php
Risk: Medium
Confidence: Low
Parameter:
Attack:
Evidence: :form action="/login.php" method="POST">
CWE ID: 352
WASC ID: 9
Description:
No Anti-CSRF tokens were found in a HTML submission form.
A cross-site request forgery is an attack that involves forcing a victim to send an HTTP request to a target
Other Info:
authenticity_token, OWASP_CSRFTOKEN, anoncsrf, csrf_token, _csrf, _csrfSecret, __csrf_magic, CSRF, _token, _csrf_token, _csrfToken] was found in the following HTML form: [Form 1: "login" "password"].
Solution:
Phase: Architecture and Design
Use a vetted library or framework that does not allow this weakness to occur or provides constructs that make this weakness easier to avoid.
Reference:
https://cheatsheetseries.owasp.org/cheatsheets/Cross-Site_Request_Forgery_Prevention_Cheat_Sheet.html
https://cwe.mitre.org/data/definitions/352.html

Alerts (15)
- Absence of Anti-CSRF Tokens (4)
  - GET: http://127.5.0.1/login.php
  - GET: http://127.5.0.1/user_new.php
  - POST: http://127.5.0.1/login.php
  - POST: http://127.5.0.1/user_new.php
- Application Error Disclosure (27)
- Content Security Policy (CSP) Header Not Set
- Directory Browsing (27)
- Hidden File Found
  - GET: http://127.5.0.1/phpinfo.php
- Missing Anti-clickjacking Header (34)
  - GET: http://127.5.0.1/admin/

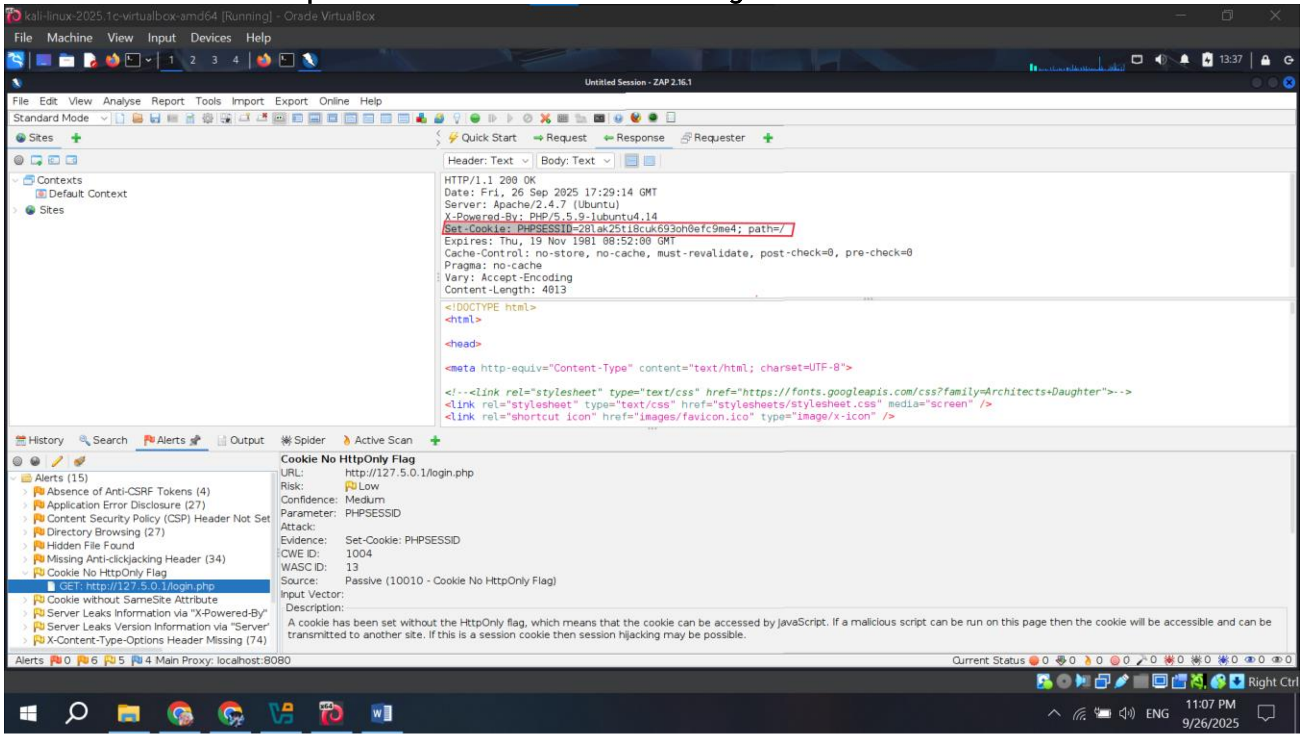
Source: Passive (10202 - Absence of Anti-CSRF Tokens)
Input Vector:
Description:
No Anti-CSRF tokens were found in a HTML submission form.
A cross-site request forgery is an attack that involves forcing a victim to send an H … r intent in order to perform an action as the victim.
The underlying cause is application functionality using predictable URL/form actions … ploits the trust that a web site has for a user. By
Other Info:
No known Anti-CSRF token [anticsrf, CSRFToken, __RequestVerificationToken, csr … oncsrf, csrf_token, _csrf, _csrfSecret, __csrf_magic, CSRF, _token, _csrf_token, _csrfToken] was found in the following HTML form: [Fo
Solution:
Phase: Architecture and Design
Use a vetted library or framework that does not allow this weakness to occur or provides constructs that make this weakness easier to avoid.
For example, use anti-CSRF packages such as the OWASP CSRFGuard.

Alerts 0 6 5 4 Main Proxy: localhost:8080
Current Status 0 0 0 0 0 0 0 0 0 0

- Impact: Successful exploitation can lead to theft of session cookies, redirection to malicious sites, defacement of the application, or full account takeover if combined with session hijacking. This vulnerability undermines user trust and poses a significant risk to data confidentiality and integrity.
- Mitigation: Implement strict input validation and contextual output encoding for all user-supplied data. Use secure frameworks that provide automatic escaping, enforce a strong Content Security Policy (CSP), and sanitize HTML/JavaScript inputs. Regularly test using automated scanners (e.g., OWASP ZAP) and manual verification.

Vulnerability 4: Cookie No HttpOnly Flag

- Risk rating: Medium
- Description: Some cookies are set without the `HttpOnly` flag, making them accessible via client-side JavaScript. If a malicious script executes on the page (e.g., through XSS), these cookies could be stolen and sent to an attacker. Session cookies without this protection may allow session hijacking.
- Evidence: ZAP flagged the cookies as accessible by JavaScript, and a screenshot of the alert has been captured to document the finding.


kali-linux-2025.1c-virtualbox-amd64 [Running] - Oracle VirtualBox
File Machine View Input Devices Help
Untitled Session - ZAP 2.16.1
File Edit View Analyse Report Tools Import Export Online Help
Standard Mode
Sites
Contexts
Default Context
Sites
Quick Start
Request
Response
Requester
Header: Text
Body: Text
History Search Alerts Output Spider Active Scan

```
HTTP/1.1 200 OK
Date: Fri, 26 Sep 2025 17:29:14 GMT
Server: Apache/2.4.7 (Ubuntu)
X-Powered-By: PHP/5.5.9-1ubuntu4.14
Set-Cookie: PHPSESSID=28lak25ti8cuk693oh0efc9me4; path=/
Expires: Thu, 19 Nov 1981 08:52:00 GMT
Cache-Control: no-store, no-cache, must-revalidate, post-check=0, pre-check=0
Pragma: no-cache
Vary: Accept-Encoding
Content-Length: 4013
```

```
<!DOCTYPE html>
<html>

<head>

<meta http-equiv="Content-Type" content="text/html; charset=UTF-8">

<!--<link rel="stylesheet" type="text/css" href="https://fonts.googleapis.com/css?family=Architects+Daughter">-->
<link rel="stylesheet" type="text/css" href="stylesheets/stylesheet.css" media="screen" />
<link rel="shortcut icon" href="images/favicon.ico" type="image/x-icon" />
```

Alerts (15)
- Absence of Anti-CSRF Tokens (4)
- Application Error Disclosure (27)
- Content Security Policy (CSP) Header Not Set
- Directory Browsing (27)
- Hidden File Found
- Missing Anti-clickjacking Header (34)
- Cookie No HttpOnly Flag
  - GET: http://127.5.0.1/login.php
- Cookie without SameSite Attribute
- Server Leaks Information via "X-Powered-By"
- Server Leaks Version Information via "Server"
- X-Content-Type-Options Header Missing (74)

Cookie No HttpOnly Flag

URL:	http://127.5.0.1/login.php
Risk:	Low
Confidence:	Medium
Parameter:	PHPSESSID
Attack:	
Evidence:	Set-Cookie: PHPSESSID
CWE ID:	1004
WASC ID:	13
Source:	Passive (10010 - Cookie No HttpOnly Flag)
Input Vector:	

Description:

A cookie has been set without the HttpOnly flag, which means that the cookie can be accessed by JavaScript. If a malicious script can be run on this page then the cookie will be accessible and can be transmitted to another site. If this is a session cookie then session hijacking may be possible.

Alerts 0 6 5 4 Main Proxy: localhost:8080 Current Status 0 0 0 0 0 0 0 0 0 0

11:07 PM 9/26/2025

- Impact: If the hacker exploits the XSS, they could steal cookies and hjack sessions.
- Mitigation: Make sure the HttpOnly Flag is set for all cookies

3. Map the vulnerabilities to OWASP Top 10 threats

Vulnerability	Impact Level	OWASP Top 10 Category
Hidden File Disclosure - /phpinfo.php	Medium	A05: Security Misconfiguration
Missing Anti-Clickjacking Header	Medium	A05: Security Misconfiguration
Reflected Cross-Site Scripting (XSS)	High	A03: Injection
Cookie No HttpOnly Flag	Medium	A07: Identification and Authentication Failures