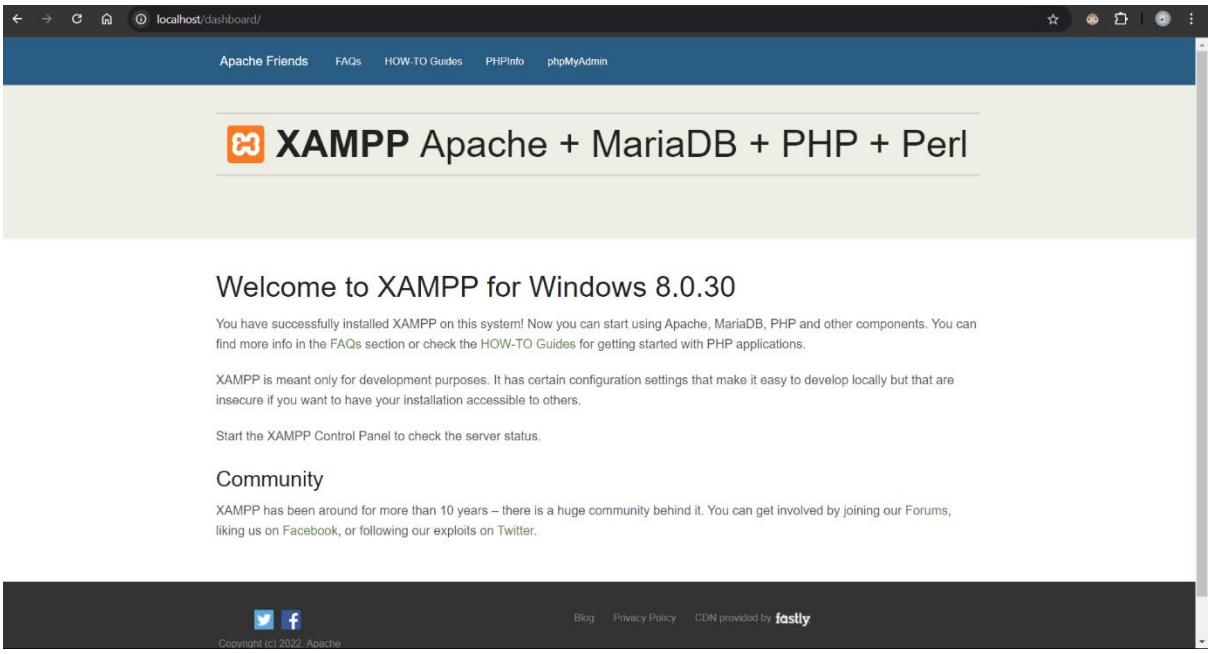
SHREYASH DUBEY
22MEI10038

ASSIGNMENT

Technical Implementation-Based Assignment: HTTP Security

Task 1: Setup HTTPS on a Web Server

1. **Install a Web Server:**



2. **SSL/TLS Certificate Verification:**
   Steps I Took to Configure HTTP Security Headers in Apache

Accessed Apache Configuration: I opened the Apache configuration file by navigating to:

bash

Copy code

cd C:\xampp\apache\conf\extra

Then, I edited httpd.conf using VS Code:

bash

Copy code

code httpd.conf

Added Security Headers: I added the following lines at the end of the file:

apache

SHREYASH DUBEY
22MEI10038

Copy code

Header always set Strict-Transport-Security "max-age=31536000; includeSubDomains"

Header set Content-Security-Policy "default-src 'self';"

Header set X-Frame-Options "SAMEORIGIN"

Header set X-Content-Type-Options "nosniff"

Header set Referrer-Policy "no-referrer"

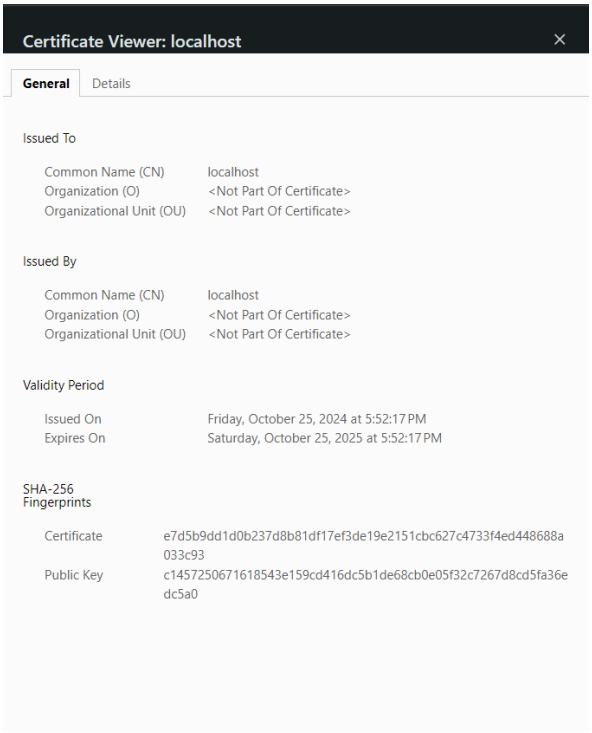
Enabled mod_headers: I ensured the mod_headers module was enabled by uncommenting the corresponding line in httpd.conf:

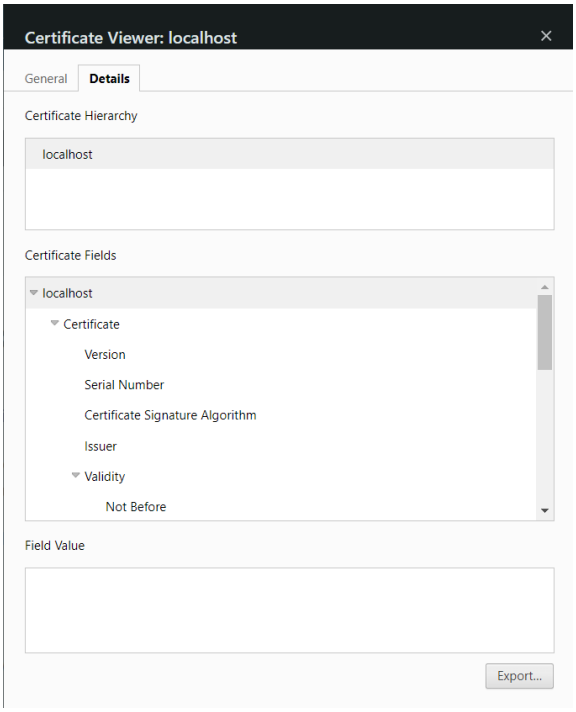
apache

Copy code

LoadModule headers_module modules/mod_headers.so

Restarted Apache: After saving my changes, I restarted Apache from the XAMPP Control Panel.





SHREYASH DUBEY
22MEI10038

```
PS C:\xampp\apache\conf\ssl.crt> openssl s_client -connect localhost:443
Connecting to 127.0.0.1
CONNECTED(000001DC)
Can't use SSL_get_servername
depth=0 CN=localhost
verify error:num=18:self-signed certificate
verify return:1
depth=0 CN=localhost
verify return:1
---
Certificate chain
 0 s:CN=localhost
   i:CN=localhost
   a:PKEY: rsaEncryption, 2048 (bit); sigalg: RSA-SHA256
   v:NotBefore: Oct 25 12:22:17 2024 GMT; NotAfter: Oct 25 12:22:17 2025 GMT
---
Server certificate
-----BEGIN CERTIFICATE-----
MIIDCTCCAfGgAwIBAgIUGD9cgPWaDtvcvfudye04wsATSfswDQYJKoZIhvcNAQEL
BQAwFDESMBAGA1UEAwwJbG9jYWxob3N0MB4XDTI0MTAyNTEyMjIxN1oXDTI1MTAy
NTEyMjIxN1owFDESMBAGA1UEAwwJbG9jYWxob3N0MIIBIjANBgkqhkiG9w0BAQEF
AAOCAQ8AMIIBCgKCAQEApMI0TVxV04sQwUQN7EvqtbUr1mNc27oSL76rXU0kG/T6
H4HXij1SCt095hQ5imkYziqIwc8xedU3ArUWrt+P+OPAaSA7t3OH64yXqEn9wq2M
YChA4QDBOw0iULc7eVKEPx14myp8c1GA900UGwnM9lAv48dKxndvxtBFYtaiIhlC
WEtQq4R2CxcdTruhEMAaqP3RFKkK8ZXsXvqUlhJ/unu81WPJCV9AUd/f5l9hhI/2
iaaF9hDW8W2AxFOUJgRjz+SFMtYOKpldKsDai+TZfGZH03B4uwdUaG0bLf3p+vL2
vHCKwNl2SRzoNvmYXfZ3KidWwd4sM+3SpDLYuyu3hwIDAQABo1MwUTAdBgNVHQ4E
FgQU//CyMe/G/kgh3RBGIWn7Jv17gMcwHwYDVR0jBBgwFoAU//CyMe/G/kgh3RBG
IWn7Jv17gMcwDwYDVR0TAQH/BAUwAwEB/zANBgkqhkiG9w0BAQsFAAOCAQEABa+v
3kP0ct1k8tZ0k4Jpsz82MRAela36Mb1MKYBz9VEHSSM8q43Y+41WptxAD1r7bHyd
g3KhCQozNCTXucbE1+AsXyhIxBdMQtRYcB/ETECOSUGRlGDsDP1cBSVA1pjMzU/s
vCznSTJUrtRGWo/Yyb0IxK5pUSZJ0VBZKNtOwuyqm/HKrZzJf8O1nkJEJd23AiV8
98QWR+EGS51pUaTkJoHSw/MWEWD/vqPbJqya21l+xV8vXywBgRQTygQke7I/7AFD
aymO8k8MNy2sDkD+OZa2NG4Jpf5Y6IUy7+XMy/NkAsW0iLt0bUTVF+NY1/+sfg8q
xX4PAupChzpSFvVKyg==
-----END CERTIFICATE-----
subject=CN=localhost
issuer=CN=localhost
---
No client certificate CA names sent
```

```
    Start Time: 1729859322
    Timeout   : 7200 (sec)
    Verify return code: 18 (self-signed certificate)
    Extended master secret: no
    Max Early Data: 0
--
ead R BLOCK
```

SHREYASH DUBEY
22MEI10038

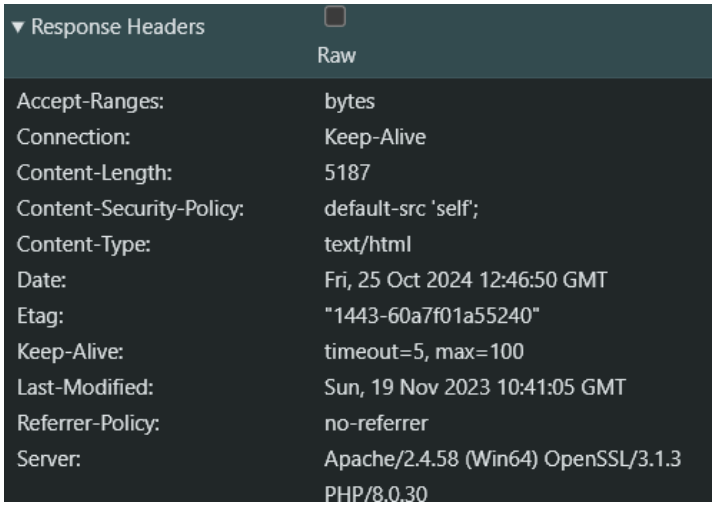
Task 2: Configure HTTP Security Headers:

The file C:\xampp\apache\conf\httpd.conf consists the security headers by:

```
<VirtualHost *:443>
    # Your existing configuration, e.g., ServerName, DocumentRoot, etc.

    # Enforce Strict-Transport-Security (HSTS)
    Header always set Strict-Transport-Security "max-age=31536000; includeSubDomains"

    # Content Security Policy (CSP)
    Header set Content-Security-Policy "default-src 'self';"

    # Prevent clickjacking
    Header set X-Frame-Options "SAMEORIGIN"

    # Avoid MIME-type sniffing
    Header set X-Content-Type-Options "nosniff"

    # Control how referrer information is sent
    Header set Referrer-Policy "no-referrer"

    # Other existing configurations...
</VirtualHost>
```

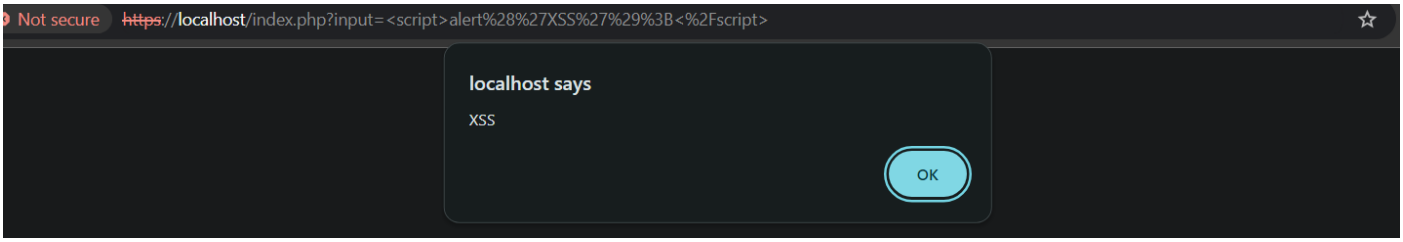
SHREYASH DUBEY
22MEI10038

Testing:



Task 3: Implement and Mitigate HTTP Vulnerabilities:

The C:\xampp\htdocs\index.php file consists of the simple form which demonstrates how **unescaped user input** can cause an XSS vulnerability



The C:\xampp\htdocs\index_new.php Implements **input validation and escaping techniques** to prevent XSS attacks.

by:

```
 if (preg_match('/^[\w\s\.\,\!\?]+$/', $input)) {
                // Escape output to prevent XSS
                echo "Raw input: " . htmlspecialchars($input) . "<br>";
                echo "Processed output: " . htmlspecialchars($input);
            } else {
                echo "Invalid input detected.";
            }
```

SHREYASH DUBEY
22MEI10038

Cross-Site Request Forgery (CSRF) Protection:

1. C:\xampp\htdocs\login.php file consists of the login form which ensures that form submissions are accepted only if the correct CSRF token is present.

   this is performed by:
   Session Start:
   - The session is started with session_start(). This is necessary to store and retrieve the CSRF token.
2. **CSRF Token Generation**:
   - A CSRF token is generated using bin2hex(random_bytes(32)) if it doesn't already exist in the session. This ensures that the token is unique and secure.
3. **Form Submission Handling**:
   - When the form is submitted (checked by $_SERVER['REQUEST_METHOD'] === 'POST'), the code checks whether the CSRF token from the form matches the one stored in the session using hash_equals(). This function is used to mitigate timing attacks.
4. **User Login Processing**:
   - If the CSRF token is valid, the username and password are checked (this is a dummy check for demonstration). In a real application, you would validate these credentials against a database.
5. **CSRF Token in Form**:
   - The CSRF token is included as a hidden input field in the form, ensuring it gets sent with the form submission.

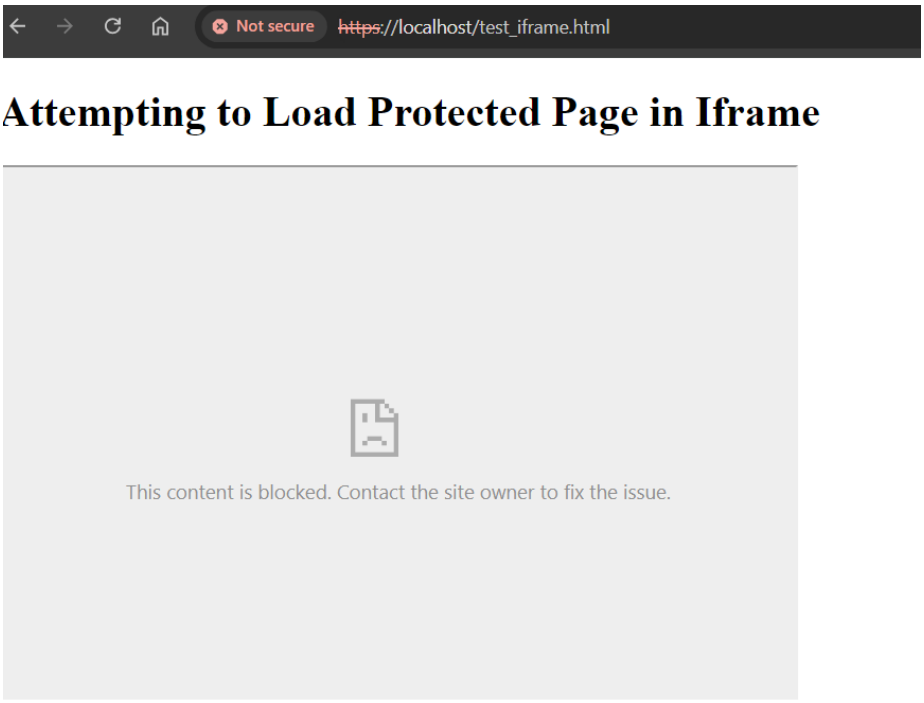
Testing the CSRF Protection

1. **Valid Submission**:

- Fill in the form with valid credentials (e.g., admin and password) and submit it. If the CSRF token is valid, you should see "Login successful!".

2. **Invalid Token**:
   - Try modifying the CSRF token in the form (for example, using browser developer tools to change the value) and submit the form. You should see "CSRF token validation failed."

Clickjacking Prevention:
The X-frame option heard: SAMEORIGIN ensures that Only the same origin can frame the content.
C:\xampp\htdocs\test_iframe.html demonstrates how the header prevents your site from being embedded in iframes on other websites



SHREYASH DUBEY
22MEI10038

TASK 5:

Findings Report: HTTP Security Testing

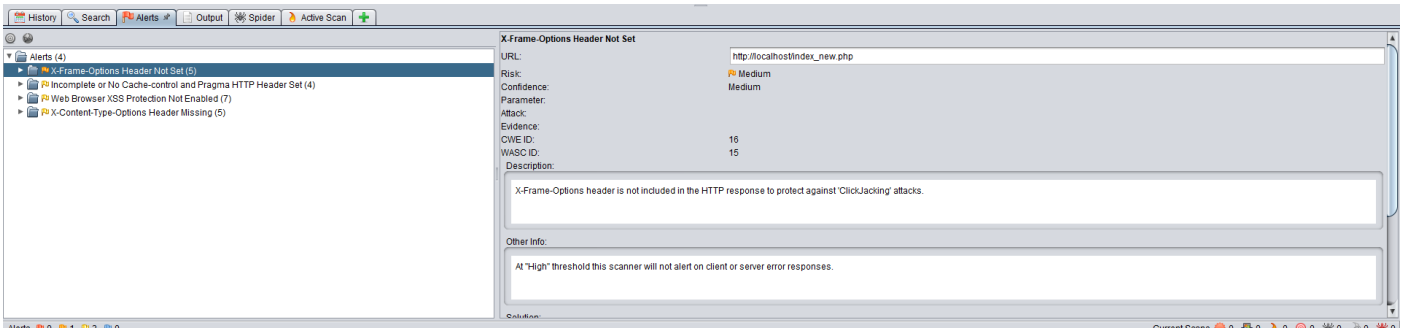
1. Introduction This report summarizes the findings from the HTTP security testing performed on the local application hosted on http://localhost.

2. Tools Used

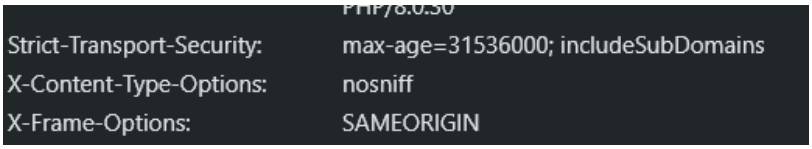
- OWASP ZAP
- Browser Developer Tools

3. Testing Results

- **OWASP ZAP**:
    - Identified Issues:



- **Browser Developer Tools**:
    - Security Headers Confirmed:
        - X-Frame-Options: SAMEORIGIN
        - Screenshot:



4. Identified Issues

- XSS vulnerability found in index_new.php. Resolved by implementing input validation and escaping.

5. Conclusion The application has a strong security posture with minor vulnerabilities that were promptly addressed. Further testing and regular updates are recommended.