

14.08.2023

Курс:

Практическая работа к уроку № Lesson_2

--

Задание_1:

Найдите XSS на странице XSS(Stored) проекта DVWA на простом (Low) уровне сложности и определите ее тип. Составьте отчет о найденной уязвимости. В отчете укажите, в каком контексте присутствует XSS.

- http://192.168.56.11/dvwa/vulnerabilities/xss_s/

192.168.56.11/dvwa/vulnerabilities/xss_s/

Kali Tools Kali Docs Kali Forums Kali NetHunter Exploit-DB Google Hacking DB OffSec vagr

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)

DVWA Security
PHP Info
About

Logout

Username: admin
Security Level: low
PHPIDS: disabled

Vulnerability: Stored Cross Site Scripting (XSS)

Name *

Message *

Sign Guestbook Clear Guestbook

Name: test
Message: This is a test comment.

More Information

- [https://www.owasp.org/index.php/Cross-site_Scripting_\(XSS\)](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/>

View Source

```
# Stored XSS Source
<?php      if( isset( $_POST[ 'btnSign' ] ) ) {
// Get input
```

```

$message = trim( $_POST[ 'mtxMessage' ] );
$name     = trim( $_POST[ 'txtName' ] );
// Sanitize message input
$message = stripslashes( $message );
$message = ((isset($GLOBALS["__mysqli_ston"]) &&
is_object($GLOBALS["__mysqli_ston"])) ?
mysqli_real_escape_string($GLOBALS["__mysqli_ston"], $message ) :
(trigger_error("[MySQLConverterToo] Fix the mysql_escape_string() call! This code
does not work.", E_USER_ERROR)) ? "" : "");
// Sanitize name input
$name = ((isset($GLOBALS["__mysqli_ston"]) &&
is_object($GLOBALS["__mysqli_ston"])) ?
mysqli_real_escape_string($GLOBALS["__mysqli_ston"], $name ) : ((trigger_error("[
MySQLConverterToo] Fix the mysql_escape_string() call! This code does not work.",
E_USER_ERROR)) ? "" : "");
// Update database
$query = "INSERT INTO guestbook ( comment, name ) VALUES ( '$message', '$name' );";
$result = mysqli_query($GLOBALS["__mysqli_ston"], $query ) or die( '

```

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)

DVWA Security

Vulnerability: Stored Cross Site Scripting (XSS)

Name *

Message *

Sign Guestbook

Clear Guestbook

Name: test

Message: This is a test comment.

Name: 1111111

Message: 1111111111

More Information

- [https://www.owasp.org/index.php/Cross-site_Scripting_\(XSS\)](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/>

http://192.168.56.11	POST	/dvwa/vulnerabilities/xss...	✓	200	6184	HTML	Vulnerability: Stored Cross...
Request							
Pretty	Raw	Hex					
1 POST /dvwa/vulnerabilities/xss_s/ HTTP/1.1							
2 Host: 192.168.56.11							
3 User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:109.0) Gecko/20100101 Firefox/115.0							
4 Accept:							
text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,*/*;q=0.8							
5 Accept-Language: en-US,en;q=0.5							
6 Accept-Encoding: gzip, deflate							
7 Content-Type: application/x-www-form-urlencoded							
8 Content-Length: 61							
9 Origin: http://192.168.56.11							
Connection: close							
Referer:							
http://192.168.56.11/dvwa/vulnerabilities/xss_s/							
Cookie: security_level=0; _ym_uid=1690358991320892522; _ym_d=1690358991; _utma=11332212.1798629775.1690359071.1690359071.1690359071.1; _utmz=11332212.1690359071.1.1.utmcsrc=(direct) utmccn=(direct) utmcmd=(none); PHPSESSID=h16rr0g21vf906o2jdrjn3g9n1							
Upgrade-Insecure-Requests: 1							
txtName=lllllll&mtxMessage=lllllllllll&btnSign=Sign+Guestbook							
Response							
Pretty	Raw	Hex	Render				
1 HTTP/1.1 200 OK							
2 Date: Mon, 14 Aug 2023 16:17:45 GMT							
3 Server: Apache							
4 X-Powered-By: PHP/5.5.9-lubuntu4.29							
5 Expires: Tue, 23 Jun 2009 12:00:00 GMT							
6 Cache-Control: no-cache, must-revalidate							
7 Pragma: no-cache							
8 Vary: Accept-Encoding							
9 Content-Length: 5872							
Connection: close							
Content-Type: text/html; charset=utf-8							
<!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>							
<meta http-equiv="Content-Type" content="text/html; charset=UTF-8" />							
<title>							
Vulnerability: Stored Cross Site Scripting (XSS) :: Damn Vulnerable Web Application (DVWA) v1.10 *Development*							
</title>							
<link rel="stylesheet" type="text/css" href="../../dvwa/css/main.css" />							
<link rel="icon" type="image/ico" href="../../favicon.ico" />							
<script type="text/javascript" src="js/jquery.js" />							

Видим, что УЯ XSS присутствует в обоих передаваемых параметрах.

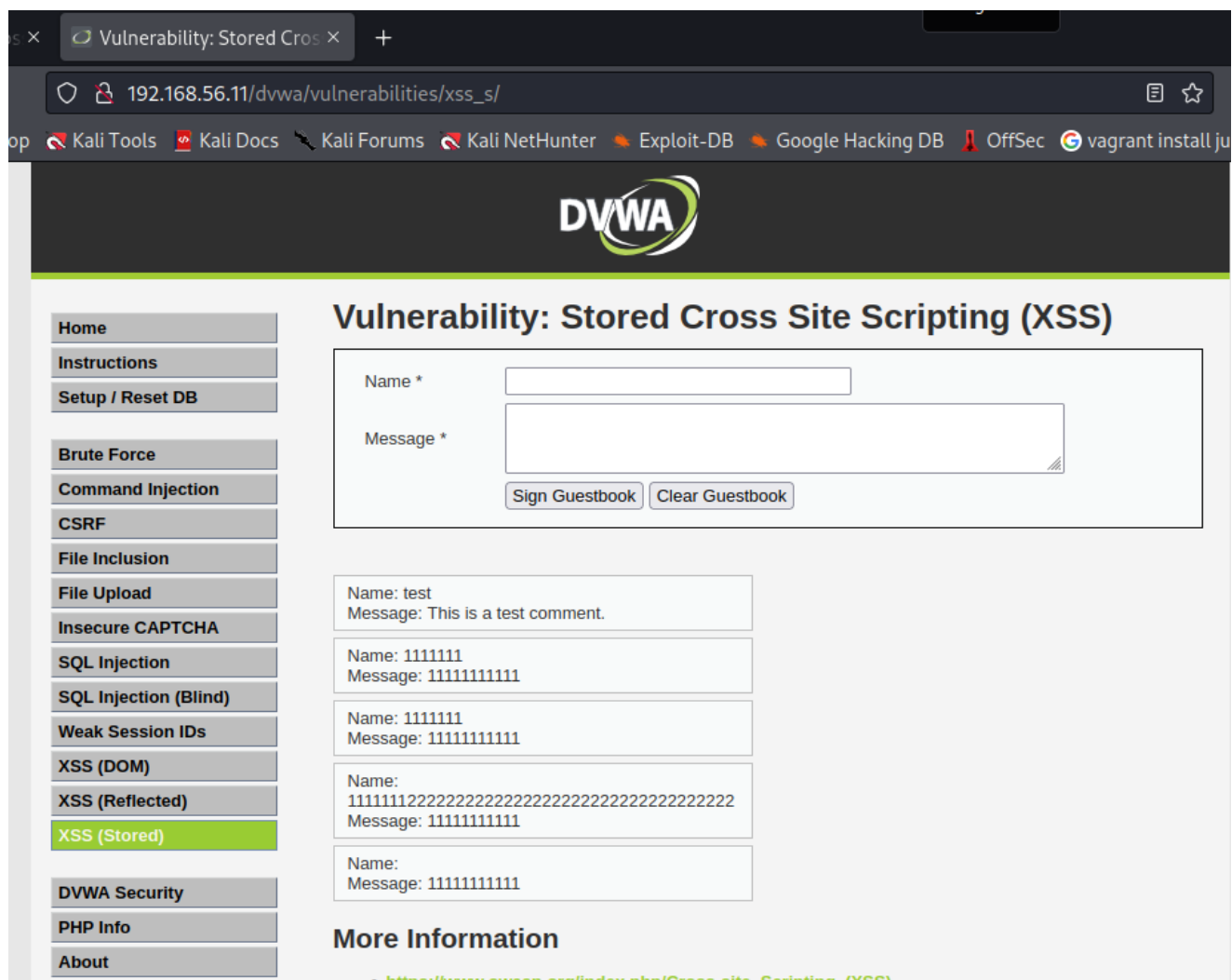
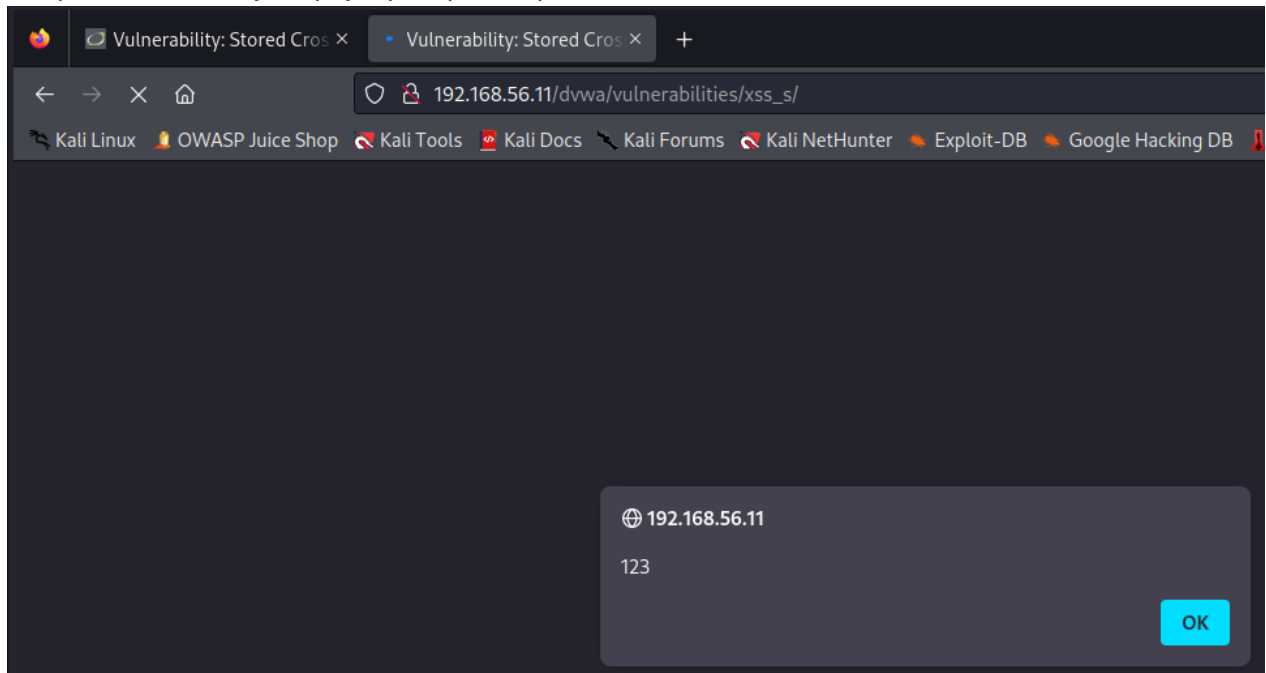
- Ограничений нет по вводу

- Вектор атаки:
 - `<script>alert(123)</script>`

```
PHPSESSID=h16rr0g2lvf906o2jdrjn3g9n1  
Upgrade-Insecure-Requests: 1  
  
txtName=<script>alert(128)</script>&mtxMessage=lllllllllll&btnSign=  
Sign+Guestbook
```

```
<div id="guestbook_comments">
  Name: <script>
    alert(123)
  </script>
  <br />
  Message: 11111111111<br />
</div>
```

- Отправляем ссылку в браузер из репитера



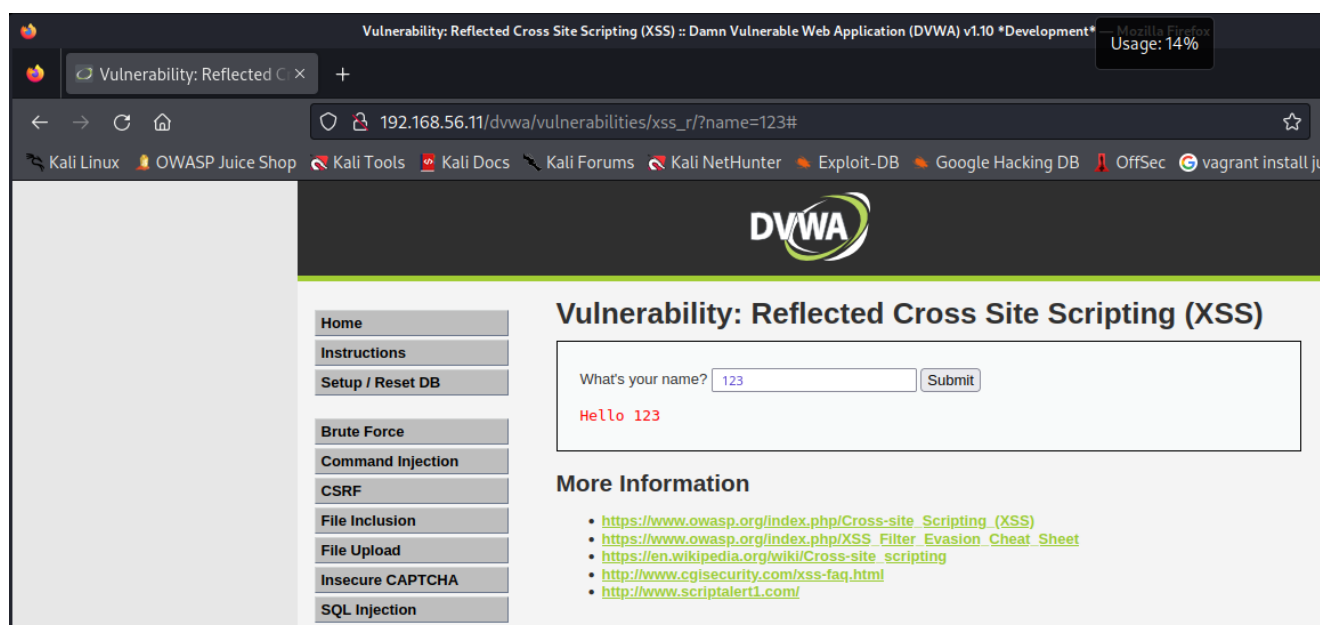
Срабатывает. Вектор на странице не остался, из этого следует, что это отраженный XSS.

Задание_2:

Найдите XSS на странице XSS(Reflected) проекта DVWA на простом (Low) уровне сложности и определите ее тип. Составьте отчет о найденной уязвимости. В отчете укажите, в каком контексте присутствует XSS.

- http://192.168.56.11/dvwa/vulnerabilities/xss_r/
- Burp Suite

```
<?php
header ("X-XSS-Protection: 0");
// Is there any input?
if( array_key_exists( "name", $_GET ) && $_GET[ 'name' ] != NULL ) {
    // Feedback for end user
    echo '<pre>Hello ' . $_GET[ 'name' ] . '</pre>';
}
?>
```



- Burp Suite (FoxitProxy: ON ; Intruder: OFF)

Данные не фильтруются, отображаются 1:1 в коде.

- Запускаем вектор атаки в Repeater:

Usage: 0%

Request

```
1 GET /dvwa/vulnerabilities/xss_r/?name=123 HTTP/1.1
2 Host: 192.168.56.11
3 User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:109.0) Gecko/20100101 Firefox/115.0
4 Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,*/*;q=0.8
5 Accept-Language: en-US,en;q=0.5
6 Accept-Encoding: gzip, deflate
7 Connection: close
8 Referer: http://192.168.56.11/dvwa/vulnerabilities/xss_r/
9 Cookie: security=low; security_level=0; _ym_uid=1690358991320892522; _ym_d=1690358991; __utma=11332212.1798629775.1690359071.1690359071.1690359071.1; __utmz=11332212.1690359071.1.1.utmcsr=(direct)|utmccn=(direct)|utmcid=(none); PHPSESSID=h16rr0g2lvf906o2jdrjn3g9n1
10 Upgrade-Insecure-Requests: 1
11
12
```

Response

```
1 HTTP/1.1 200 OK
2 Date: Mon, 14 Aug 2023 15:53:49 GMT
3 Server: Apache
4 X-Powered-By: PHP/5.5.9-1ubuntu4.29
5 Expires: Tue, 23 Jun 2009 12:00:00 GMT
6 Cache-Control: no-cache, must-revalidate
7 Pragma: no-cache
8 X-XSS-Protection: 0
9 Vary: Accept-Encoding
10 Content-Length: 5157
11 Connection: close
12 Content-Type: text/html; charset=utf-8
13
14
15 <!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
16 "http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
17
18 <html xmlns="http://www.w3.org/1999/xhtml">
19
20 <head>
21 <meta http-equiv="Content-Type" content="text/html; charset=UTF-8" />
22
23 <title>
24 Vulnerability: Reflected Cross Site Scripting (XSS) :: Damn Vulnerable
25 Web Application (DVWA) v1.10 *Development*
26 </title>
27
28 <link rel="stylesheet" type="text/css" href="../../dvwa/css/main.css" />
29
30 <link rel="icon" type="image/ico" href="../../favicon.ico" />
31
32 <script type="text/javascript" src="../../dvwa/js/dvwaPage.js">
33 </script>
34
35 </head>
36
37 <body class="home">
```

- Проверяем фильтрацию данных
- Вектор атаки:
 - 1. `canary<>123`
 - 2. `<script>alert(123)</script>`

Usage: 4%

Request

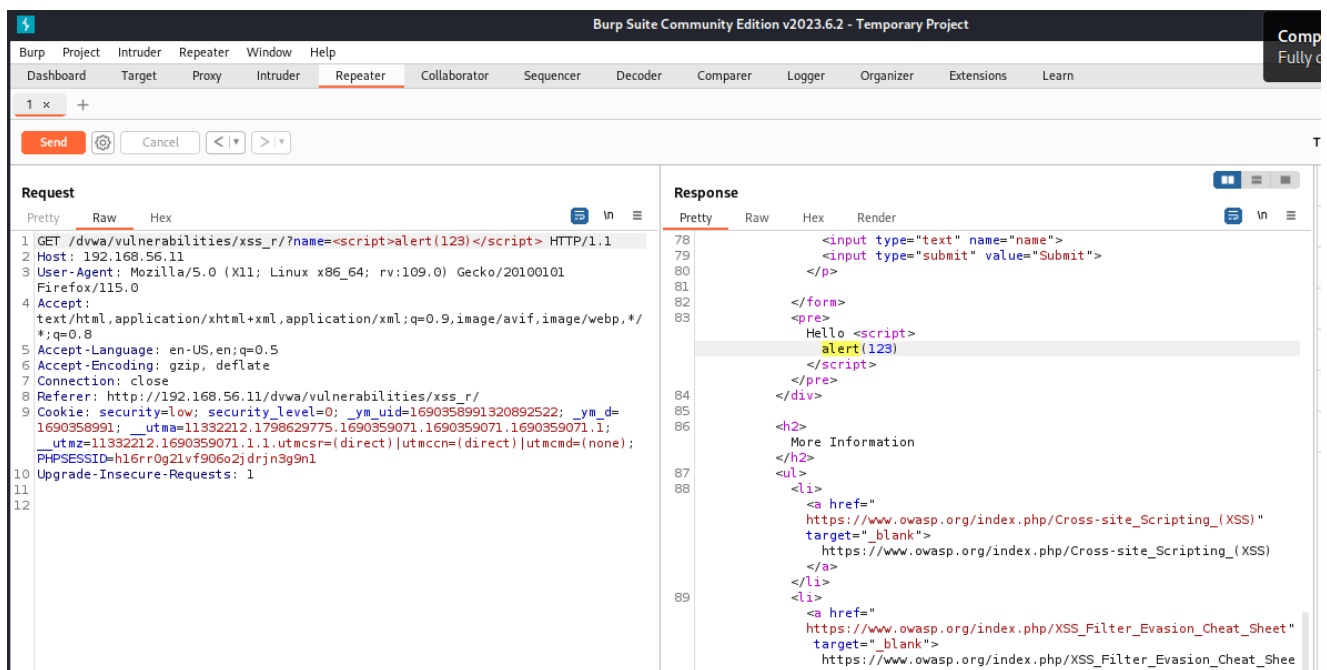
```
1 GET /dvwa/vulnerabilities/xss_r/?name=canary<>123 HTTP/1.1
2 Host: 192.168.56.11
3 User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:109.0) Gecko/20100101 Firefox/115.0
4 Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,*/*;q=0.8
5 Accept-Language: en-US,en;q=0.5
6 Accept-Encoding: gzip, deflate
7 Connection: close
8 Referer: http://192.168.56.11/dvwa/vulnerabilities/xss_r/
9 Cookie: security=low; security_level=0; _ym_uid=1690358991320892522; _ym_d=1690358991; __utma=11332212.1798629775.1690359071.1690359071.1690359071.1; __utmz=11332212.1690359071.1.1.utmcsr=(direct)|utmccn=(direct)|utmcid=(none); PHPSESSID=h16rr0g2lvf906o2jdrjn3g9n1
10 Upgrade-Insecure-Requests: 1
11
12
```

Response

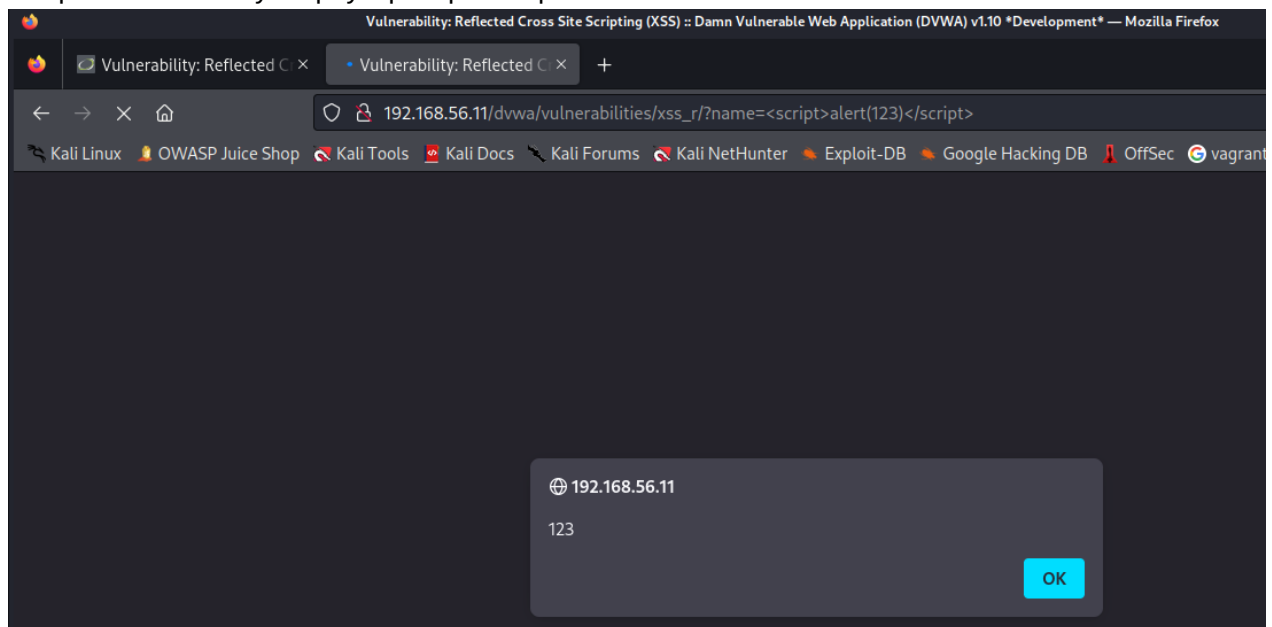
```
78
79 <input type="text" name="name">
80 <input type="submit" value="Submit">
81 </p>
82 </form>
83 <pre>
84 Hello canary<>
85 123
86 </pre>
87 </div>
88
89 <h2>
90 More Information
91 </h2>
92 <ul>
93 <li>
94 <a href="
95 https://www.owasp.org/index.php/Cross-site_Scripting_(XSS)"
96 target="_blank">
97 https://www.owasp.org/index.php/Cross-site_Scripting_(XSS)
98
```

Inspector

- Request attributes
- Request query parameters
- Request body parameters
- Request cookies
- Request headers
- Response headers



- Отправляем ссылку в браузер из репитера



Срабатывает. Вектор на странице не остался, из этого следует, что это отраженный XSS.

Задание_3:

Найдите XSS на странице XSS(DOM) проекта DVWA на простом (Low) уровне сложности и определите ее тип. Составьте отчет о найденной уязвимости. В отчете укажите, в каком контексте присутствует XSS.

- http://192.168.56.11/dvwa/vulnerabilities/xss_d/

192.168.56.11/dvwa/vulnerabilities/xss_d/

Kali ToolsKali DocsKali ForumsKali NetHunterExploit-DBGoogle Hacking DBOffSecvagrant install

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)

DVWA Security

PHP Info

About

Logout

Vulnerability: DOM Based Cross Site Scripting (XSS)

Please choose a language:

EnglishSelect

More Information

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

Username: adminSecurity Level: low

View SourceView Help

```
<?php
```

```
# No protections, anything goes
```

```
?>
```

- Отправляем запрос (Select English), отлавливаем его в Burp Suite

8http://192.168.56.11GET/dvwa/vulnerabilities/xss_d/?default=E...2005919HTMLVulnerability: DOM Based...

Request

PrettyRawHex

1GET /dvwa/vulnerabilities/xss_d/?default=English HTTP/1.1

2Host: 192.168.56.11

3User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:109.0) Gecko/20100101 Firefox/115.0

4Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,*/*;q=0.8

5Accept-Language: en-US,en;q=0.5

6Accept-Encoding: gzip, deflate

7Connection: close

8Referer: http://192.168.56.11/dvwa/vulnerabilities/xss_d/?default=English

9Cookie: security=low; security_level=0; _ym_uid=1690358991320892522; _ym_d=1690358991; __utma=11332212.1798629775.1690359071.1690359071.1690359071.1; __utwz=11332212.1690359071.1.1.utmcsr=(direct)|utmccn=(direct)|utmcmd=(none); PHPSESSID=h16rr0g2lvf906o2jdrjn3g9n1

10Upgrade-Insecure-Requests: 1

11

12

Response

PrettyRawHexRender

1HTTP/1.1 200 OK

2Date: Mon, 14 Aug 2023 16:42:44 GMT

3Server: Apache

4X-Powered-By: PHP/5.5.9-1ubuntu4.29

5Expires: Tue, 23 Jun 2009 12:00:00 GMT

6Cache-Control: no-cache, must-revalidate

7Pragma: no-cache

8Vary: Accept-Encoding

9Content-Length: 5607

10Connection: close

11Content-Type: text/html; charset=utf-8

12

13

14<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">

15

16<html xmlns="http://www.w3.org/1999/xhtml">

17

18<head>

19<meta http-equiv="Content-Type" content="text/html; charset=UTF-8" />

20

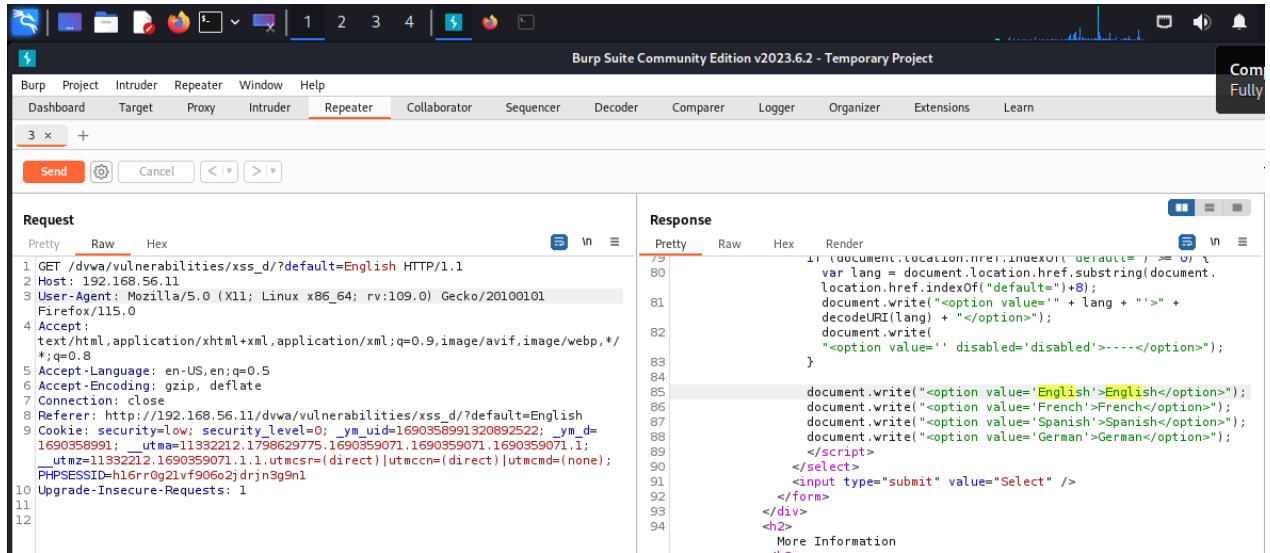
21<title>

22Vulnerability: DOM Based Cross Site Scripting (XSS) :: Damn Vulnerable Web Application (DVWA) v1.10 *Development*

23</title>

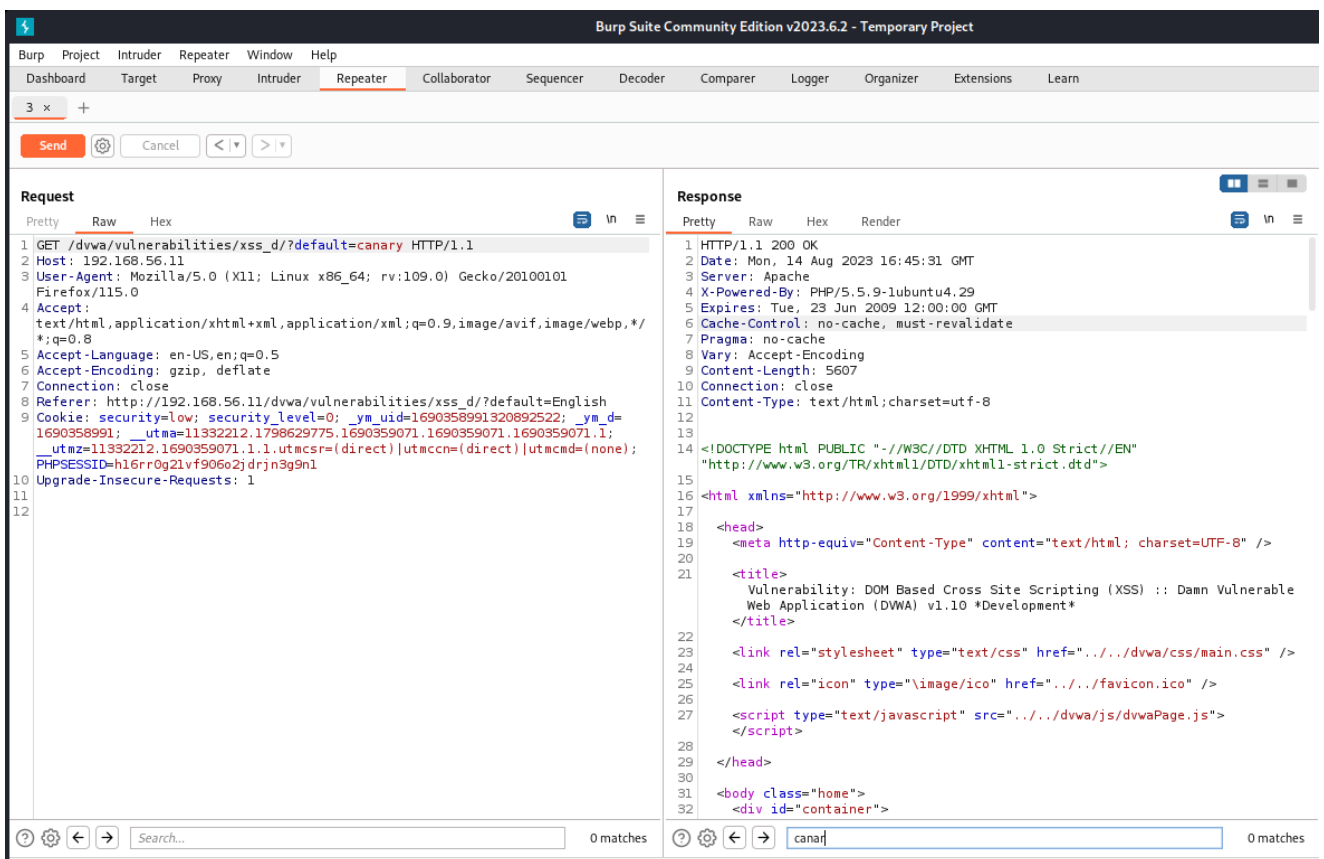
24<link rel="stylesheet" type="text/css" href="../../dvwa/css/main.css"

- Отправляем в репитер пробуем применить вектор атаки

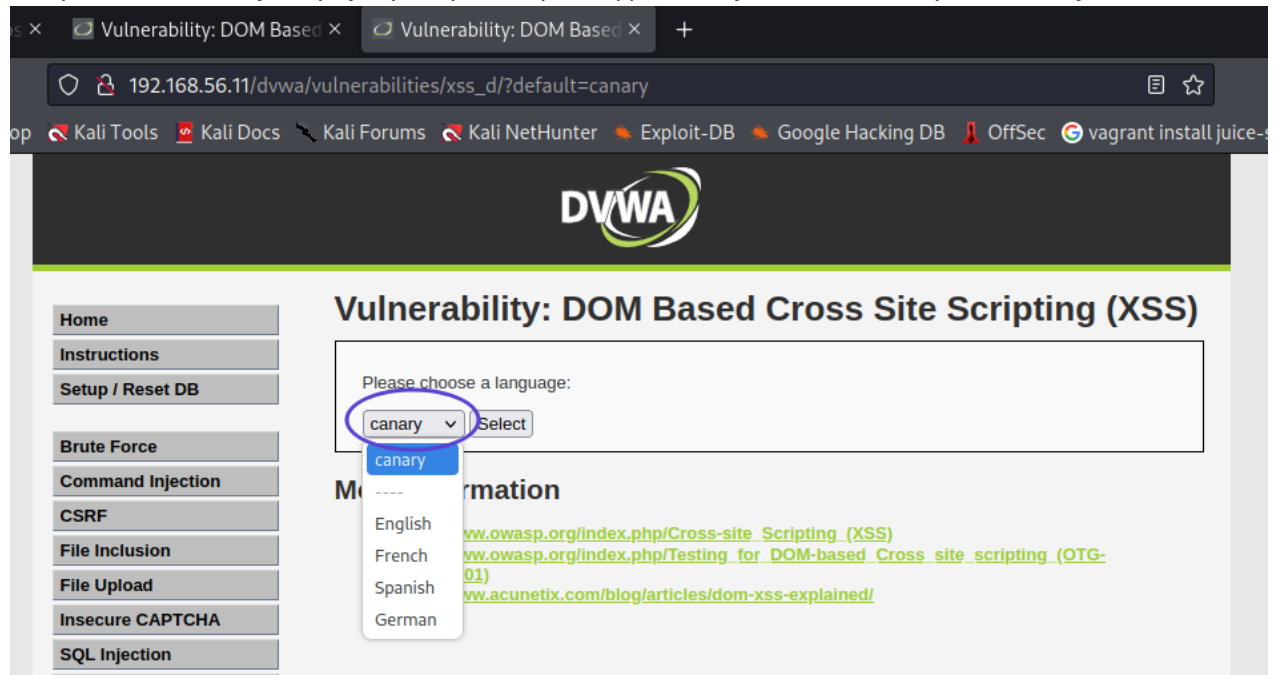


- Задаем canary

Видим, что в тексте не присутствует



- Отправляем ссылку в браузер из репитера, видим, что уже можно выбрать Canary



из чего следует, что прописывается между тегами Option

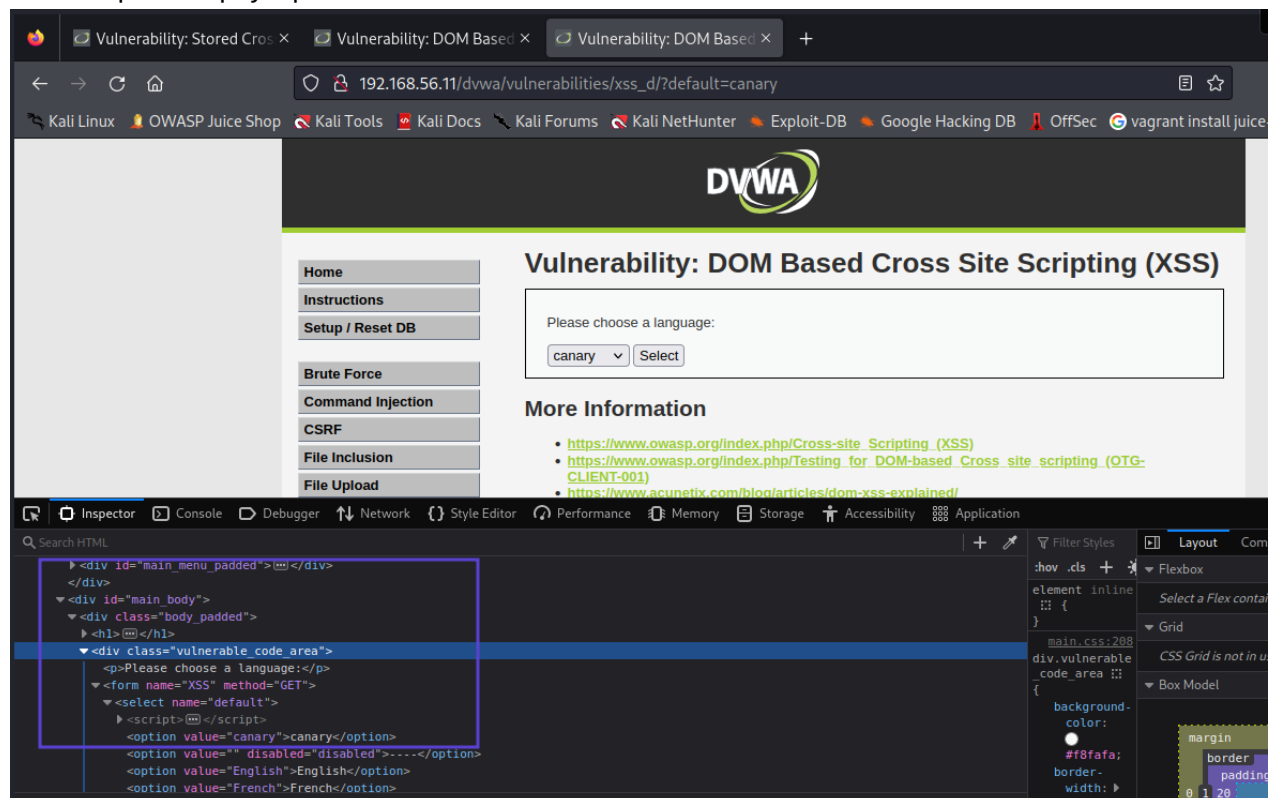
```

1 GET /dvwa/vulnerabilities/xss_d/?default=canary HTTP/1.1
2 Host: 192.168.56.11
3 User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:109.0) Gecko/20100101
4 Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,*/*;q=0.8
5 Accept-Language: en-US,en;q=0.5
6 Accept-Encoding: gzip, deflate
7 Connection: close
8 Referer: http://192.168.56.11/dvwa/vulnerabilities/xss_d/?default=English
9 Cookie: security=low; security_level=0; _ym_uid=1690358991.320892522; _ym_d=1690358991; _ym_h=11332212.1798629775.1690359071.1690359071.1690359071.1; _ym_v=11332212.1690359071.1.1.utmcsr=(direct)|utmccn=(direct)|utmcnd=(none); PHPSESSID=h16rr0g21vf906o2jdrjn3g9n1
10 Upgrade-Insecure-Requests: 1
11

78 <select name="default">
79 <script>
80 if (document.location.href.indexOf("default=") >= 0) {
81   var lang = document.location.href.substring(document.
82     location.href.indexOf("default=")+8);
83   document.write("<option value='" + lang + "'" +
84     decodeURI(lang) + "</option>");
85   document.write(
86     "<option value=' disabled=' disabled'>----</option>");
87 }
88 document.write("<option value='English'>English</option>");
89 document.write("<option value='French'>French</option>");
90 document.write("<option value='Spanish'>Spanish</option>");
91 document.write("<option value='German'>German</option>");
92 </script>
93 </select>

```

- См в Inspector браузера



Из чего следует, что это УЯ DOM XSS (base)

Задание_4:

(*) Выполните установку компонентов, необходимых для выполнения практических занятий, согласно инструкции из методички к уроку 2.

Задание_5:

(*) Найдите максимальное число XSS на странице Pen Test Tool Lookup (AJAX Version) в OWASP Mutillidae, определите их тип и составьте отчет о найденных уязвимостях.

Выводы:

- ...

Вся информация в данной работе представлена исключительно в ознакомительных целях!
Любое использование на практике без согласования тестирования подпадает под действие УК
РФ

- <https://gb.ru>

Выполнил: AndreiM