TASK 5

YOU NEED TO COMPLETE 5 XSS (CROSS SITE SCRIPTING) CHALLENGES FROM THE FOLLOWING PLATFORM:

<https://xss-quiz.int21h.jp/>

YOU NEED TO PROVE THERE IS A XSS VULNERABILITY IN THAT PARTICULAR WEB PAGE

NOTE: YOU NEED TO COMPLETE THE CHALLENGES ONE AFTER ANOTHER. BECAUSE AFTER YOU COMPLETE THE FIRST CHALLENGE ONLY YOU WILL ENTER INTO SECOND CHALLENGE.

Summary:

A black and white background with orange text

Description automatically generated with low confidence

It is a website that is used to test XSS vulnerability through challenges. Each challenge is a stage and by completing one challenge we can access the next challenge.

XSS is a type of security vulnerability commonly found in web applications that allows attackers to inject malicious scripts into trusted websites viewed by other users.

By participating in the XSS quiz, we can improve their knowledge and skills in identifying and preventing XSS vulnerabilities. It serves as a valuable resource for developers, security professionals, and individuals interested in web application security.

Stage 1:

A screenshot of a computer error

Description automatically generated with low confidence

As we can in stage 1, we must find the XSS vulnerability, and we are given with the hint that it is very simple.

Now we will insert the following JavaScript command in the search box:

<script>alert(document.domain);</script>  


So, we get an alert box on the page:

A screenshot of a computer

Description automatically generated

So, we have passed Stage 1 and we can go to Stage 2.



Stage 2:

A screenshot of a computer

Description automatically generated with medium confidence

In this challenge we must close the current tag and then add the script tag.

So, we can use the following JavaScript command for this stage:

"><script>alert(document.domain);</script>



When we use this command, we get an alert box.

A screenshot of a computer

Description automatically generated with medium confidence

We have passed Stage 2, now we can go to Stage 3.

A screenshot of a computer

Description automatically generated with medium confidence

Stage 3:

A screenshot of a computer

Description automatically generated with low confidence

In this Stage we must give a JavaScript command in the search box, but in this stage the input is properly escaped. Hence the normal alert function wont work in the field as the > and < symbols are encoded.

So, we must pass the input in the other field i.e., the drop-down list in the form element, we must change the value of the drop-down list element and then anything given in the search field will work.

A screenshot of a computer program

Description automatically generated with low confidence

These are the elements of the drop-down list, and we must replace one of the values with the following command:

<script>alert(document.domain);</script>

A picture containing text, screenshot, software

Description automatically generated

Now we have changed the value and then we execute the command.



We get an alert box on the website:

A picture containing text, screenshot, font

Description automatically generated

We have passed the Stage 3 and now we can go to Stage 4

A screenshot of a computer

Description automatically generated with medium confidence

Stage 4:

A screenshot of a computer

Description automatically generated with low confidence

In Stage 4, we have been given a hint of the page having an invisible input field.

We can use that input field and pass the input there to execute the JavaScript command.

When we inspect the form element, we find an invisible input field named p3.



We must change its default value to our payload and execute the code.

However, trying to insert that into the input field's value attribute won't work, because of the " character at the beginning. Instead, insert into the value attribute with the encoded code of the special symbols.

&#34> <script>alert(document.domain);</script>



After updating the value attribute, when we search something, we get an alert box.

A screen shot of a computer

Description automatically generated with medium confidence

We have passed Stage 4 and can go to Stage 5.

A screenshot of a computer

Description automatically generated with medium confidence

Stage 5:

A screenshot of a computer

Description automatically generated with low confidence

In the Stage 5 page, the input text field is of limited length.



As we can see the length is fixed to 15, so we must increase the size of the text box by using the inspect action to insert our JavaScript command.



I have changed the size to 60 and the code can now be executed.



We get an alert box with the domain name of the website.

A picture containing text, screenshot, font

Description automatically generated

We have passed Stage 5 and we can now go to Stage 6.

Stage 6:

A screenshot of a computer

Description automatically generated with medium confidence

In this stage, we are given a hint of using the event handler attributes.

We are going to use the onmouseover attribute to complete this task.

As we know in this stage the input is completely escaped like the < and > are completely escaped and cannot be used directly, so we use a command like:

123" onmouseover="alert(document.domain);



When this code is used, the value attribute will take the value of 123 and it will also have an additional function of onmouseover, whenever the mouse is hovered on the “alert(document.domain)” we will get the alert box with the domain name.

As long as the cursor is hovered on the form tag i.e.. the form element in the page, alert box will keep on generating .

A screen shot of a computer

Description automatically generated with medium confidence

Conclusion:

By completing 5 stages in this challenge, I came to know that having a knowledge of how a web application works is prominent rather than just having an idea of the web application in finding out a XSS vulnerability.

As XSS or Cross site scripting vulnerability is a OWASP top 10 vulnerability, it should be eliminated from each and every web application, so as to avoid the happening of remote code execution and exploitation of useful information.