

SYSTEM SECURITY

41182

Workshop 4: Vulnerable components

School of Computer Science

UTS Internal

Objectives:

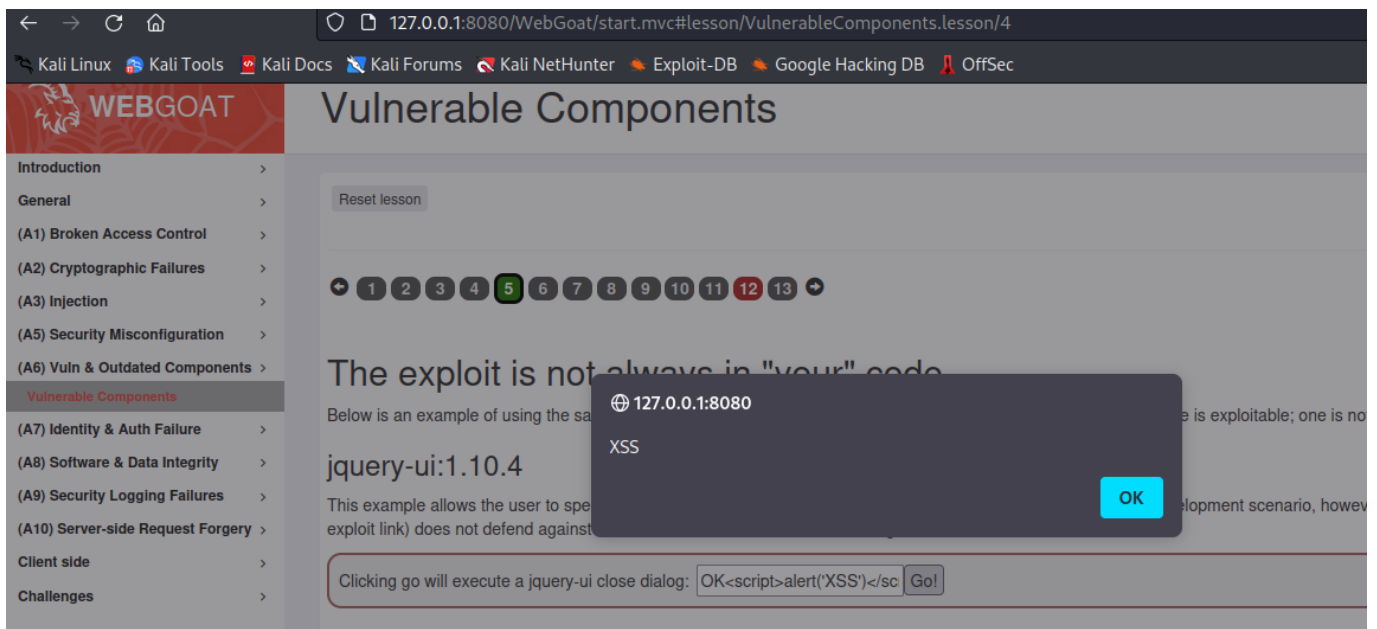
- Gain awareness that the open source consumed is as important as your own custom code.
- Gain awareness of the management, or lack of management, in our open source component consumption.
- Understand the importance of a Bill of Materials in determining open source component risk

1. Vulnerable components – WebGoat 2023.3 Lessons (A6)

This WebGoat Lesson contains a lot of readings. Please read them carefully.

- 5) Hint: Just click the Go! Buttons.

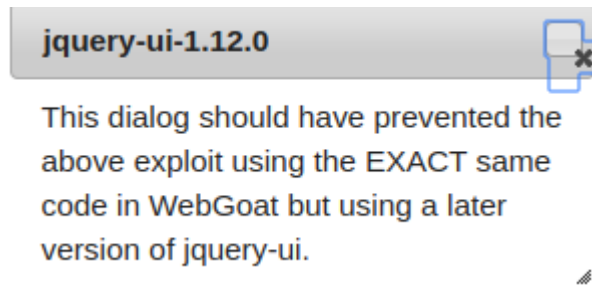
For the first one, you will see a pop-up window.



jquery-ui-1.10.4

This dialog should have exploited a known flaw in jquery-ui:1.10.4 and allowed a XSS attack to occur

For the second one, you do not see the XSS alert anymore.



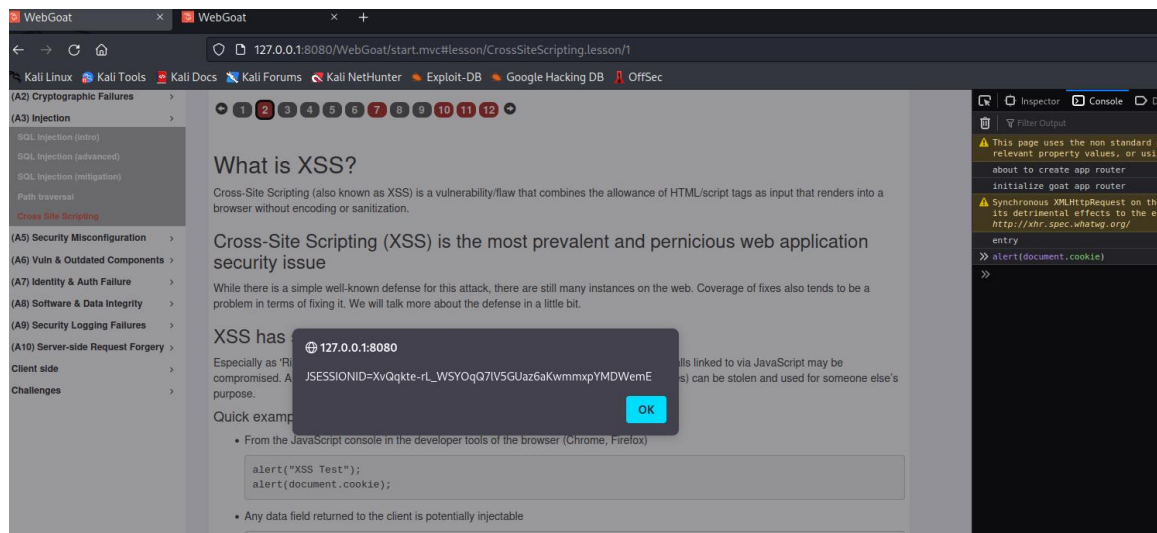
This example is used to show you the risks of using open-source libraries. You do not need to know what is exactly happening with the jquery module. But if you would like to know more about this jquery-ui vulnerability, please refer to [1] and [2] for more information.

Note: there is a bug of the step 12) in this lesson, so we will skip this part.

2. Optional: Cross-Site Scripting (XSS) – WebGoat 2023.3 Lessons (A3)

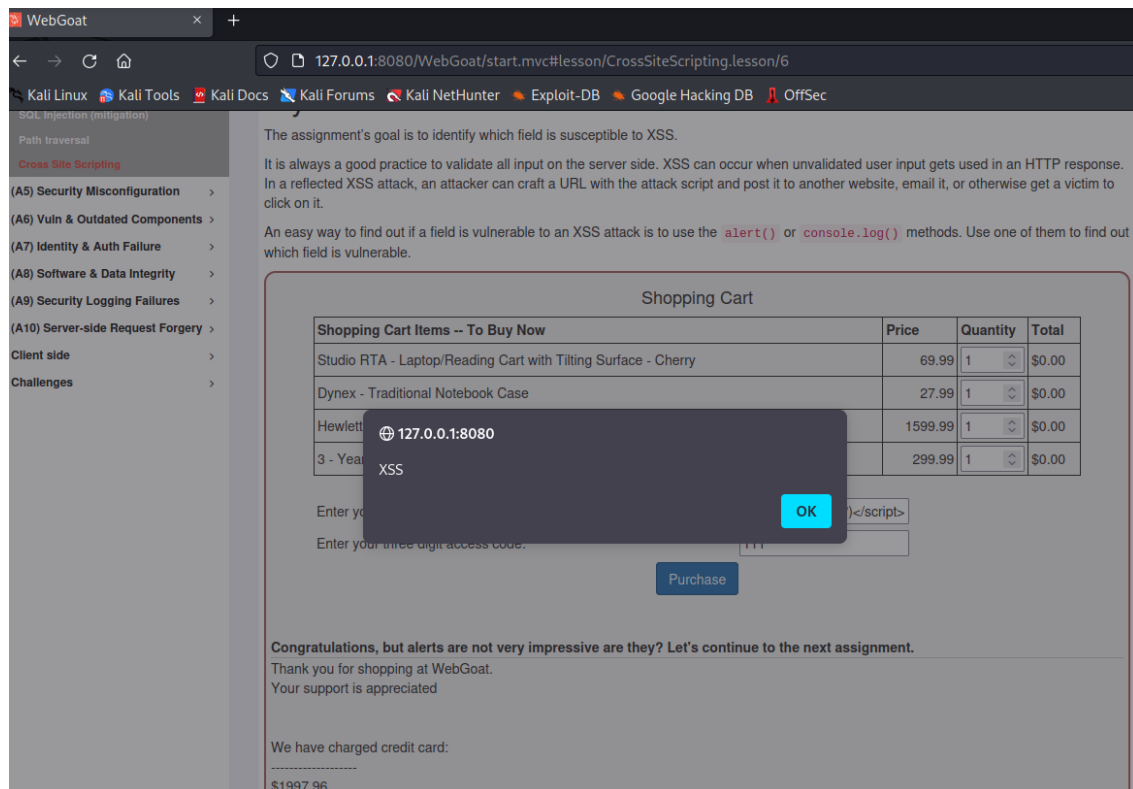
If you are interested in XSS, we have given an example of part of XSS lesson here. The details of XSS will be introduced in the “Web Security” lecture. Feel free to explore and you may use this Lesson A7 in your project.

2) Hint: Just follow the instructions. You will see the **same** cookies in the two different tabs of your browser (check more details in the video).



7) Hint: this is an example of the Reflected XSS attack. Reflected XSS attacks are the most basic form of a cross-site scripting attack. This attack happens when a user input is inserted into a webpage and a response to the user is immediately generated by the system and includes the user input in it without any content sanitisation (OWASP 2021). For example, when a website has a search function in the URL which echoes what the user has inputted, perpetrators can insert a malicious script into the search function, and this will result on the website executing and returning the script. The attacker can then conduct the attack on other users by sharing the infected URL link and any other users who load the link will execute the malicious script.

This step 7) gives an example of reflected XSS attack on a shopping cart checkout page. Users can input their desired quantity of products, credit card payment details, and 3-digit credit card access code. As the credit card number input box has a large character limit, meaning it could possibly fit a malicious script within it. Potential attacker would search for these security vulnerabilities. The vulnerability can be verified by insert an script like: `<script>alert("XSS")</script>`, and then press the “Purchase” button.



3. Discussions

Think about the following questions and discuss with your peer students:

- 1) Have you ever used any open-source libraries before? If yes, have you encountered any security issues?

References

- [1]. <https://snyk.io/test/npm/jquery-ui/1.10.4#:~:text=Overview-jquery%20Dui%20is%20a%20library%20for%20manipulating%20UI%20elements%20via.to%20execution%20of%20untrusted%20code.>
- [2]. <https://api.jqueryui.com/dialog/#option-closeText>