Sistemas de Software Seguros
Segurança de Software
2024/2025
Class Project: Experiments with WebGoat (class 1)

1. Setup the System

Execute the following steps to initiate the class:

a) Start the image of the course in a virtual machine (VirtualBox player)

b) Initiate the WebGoat by opening a Terminal, changing the current directory to `~/apps/WebGoat`, and running: `java -jar webgoat*` . Then, you must run the web browser (Firefox) and open WebGoat by selecting the URL:

```
http://localhost:8080/WebGoat
```

c) To enter, either register a new user or utilize the account with the username "`guestss`" and the password "`guestss`". Select the "Sign in" button to initiate the lessons.

Usually, in each lesson, there are several *steps*, some of which provide context information about the vulnerability (appear in grey) while others allow you to try to exploit a vulnerability (appear in red before you solve the lesson). It would be best to start by reading the context information and then gradually move to the steps where you can try the vulnerabilities. In all lessons, attempt to solve them without help. If you reach a point where you can no longer make progress, then use the "Show Hints" to help you advance. You can also start a lesson from the beginning by pressing "Reset lesson".

NOTE: You can get more information about WebWolf from https://owasp.org/www-project-webgoat/ .

NOTE: To stop the WebGoat from executing, you can do `ctr^c` on the running Java program.

2. Broken Access Control

Try to solve the lesson Broken Access Control >> Insecure Direct Object References >> Steps 2 to 5

Direct object references allow access to information that would typically be inaccessible. In this lesson, we will attempt to read and update information about other website users. This lesson is carried out by executing the following steps:

Lesson 2: Start by authenticating as Tom, as explained in the lesson.

Lesson 3: Some profile attributes are shown, but others are not displayed in the browser interface. What are the **attributes that are not listed**? Maybe you can redo the request to show the profile and then find the missing attributes (e.g., with the **ZAP tool** --- see the end of the document --- you can intercept and view the contents of the response or with the **Browser Developer Tools**).

Lesson 4: Maybe we can see our profile using an **alternative request format**. This method is probably generic because it would also allow us to know the information of other users. Still, to do that, we need to provide some information that univocally identifies that particular user. How can we modify the request to ask for your profile in such a way? Minor changes to the request can go a long way.

Lesson 5: Now, let's try to **see another user's profile**. How can we find the identifier of another user? Maybe we can do trial and error … (HINT: Maybe you can use ZAP to experiment with different IDs. Try increasing the ID value of Tom. It is not as simple as adding 1, but it is also not too far off).

3. Path Traversal

A path(directory) traversal is a vulnerability where an attacker can access or store files and directories outside the location where the application is running. This may lead to reading files from other directories and overwriting critical system files in case of a file upload.

Try to solve the lesson Injection >> Path Traversal >> Step 2 to 4

Lesson 2: Try to upload the file from **/home/ss/ptcat.jpg** to a location outside the guestss webgoat user. Maybe you can try to upload the file to a level up of guestss. First, you must download the file **ptcat.jpg** from the course Moodle website to the /home/ss to perform this lesson. (NOTE: You can use ZAP or the Browser Developer Tools to see the content of the exchanged messages).

Lesson 3: Now, the application has some protection against path traversal attacks. Execute the same exercise as lesson 2 but try to circumvent the protection implemented. Maybe you can experiment with different names to understand what the application does to inputs when you try to perform the attack.

Lesson 4: Once again, the developer enhanced the protection against path traversal attacks. Try to execute the same exercise of lesson 2, circumventing this new protection. (HINT: Maybe you will need to change the request transmitted by the browser to the web application, e.g., with the ZAP).

4. Cross-Site Scripting (XSS)

Try to solve the lesson Injection >> Cross-Site Scripting >> Step 7

1) The objective is to use the shopping form to reflect input from the user. How could one do it?

2) The form is quite similar to other sites. It has a set of products, prices, and quantity. It is possible to change the quantity and see the updated price. Finally, the user can provide the credit card number and access code and pay for the selected goods.

3) By trial and error, experiment with different things (combinations of letters and digits) on the various input fields to see what kind of validations are performed on the server. If you find some field that has potential for a reflected XSS, you can try the following script: `<script> alert("FCUL!!!")</script>`

   NOTE: be careful that you use the right `"` (and not some other character set by your editor/OS like “ ”)

Delivery of the Report

The output of the class project is a report answering all the questions and including the justifications for the responses. Each group should deliver the report either by submitting it in the course Moodle page or, if there is some difficulty with this method, by emailing it to the professor of the TP class. The file type should be a pdf.

Deadline: 11 November 2024 (there will be no extensions)

=====================================
The following exercises are optional!
=====================================

Try also to do the following exercises.

Try to solve the lesson Broken Access Control >> Insecure Direct Object References >> step 5 (last part)

NOTE: *this part of the lesson is slightly more complex than the others !!*

Now, let's try to **modify the information associated with the profile of user Buffalo Bill**, namely, change the *role* to a lower value and the *color* to red. How can we achieve this?

(HINTS: To make the change, you have to modify the request for the profile in four different ways: (1) the original "profile" request uses the method GET. Maybe this is not the most appropriate method to update the information about a user. Which other method is better? (2) We must indicate to the user we want to update the data. You already know how to do it from the previous objective; (3) you need to provide novel information about the user. You can probably get the original information from the response to the request of the previous question. Then, you only need to provide the updated information; (4) lastly, the "Content-Type" field in the header might need to be updated to reflect the type of information you are passing to the server;)

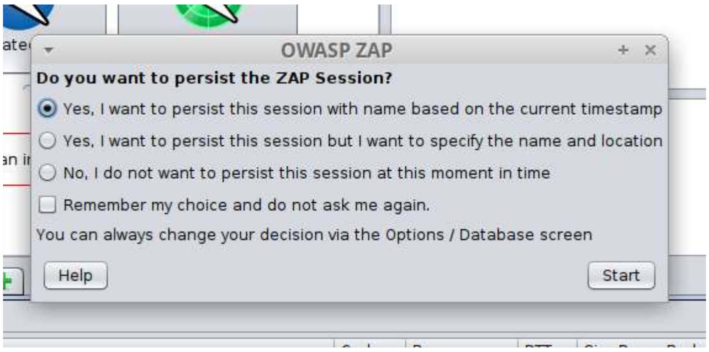
Annex:

ZAP Web Proxy

To understand better what is happening and also to help us carry out some of the attacks, we will use a web proxy called ZAP[1]. To launch the ZAP, you need to create a new Terminal and run the command:

```
~/apps/ZAP/ZAP/zap.sh
```

After doing this, you should have a new window with the ZAP. In ZAP, select the following in the pop-up window:



OWASP ZAP

Do you want to persist the ZAP Session?

● Yes, I want to persist this session with name based on the current timestamp

○ Yes, I want to persist this session but I want to specify the name and location

○ No, I do not want to persist this session at this moment in time

□ Remember my choice and do not ask me again.

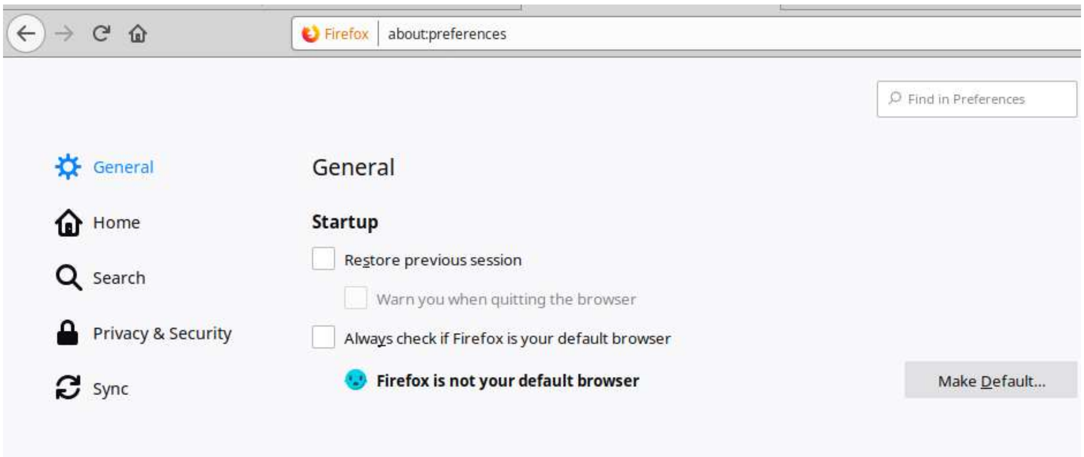
You can always change your decision via the Options / Database screen

In the next pop-up window, they might ask you if you want to update the add-ons. Just select close for now.

Then, make sure ZAP is well configured to receive connections from the browser. Select tab `Tools -> Options…` to get a new window. Next, in this window, select `Network -> Local Servers/Proxies`. Finally, make sure that the Address is `localhost` and the Port is set to "`8088`".

[1] There is some information on the tool on the ZAP project page where you will find links to the manual (just search for "ZAP OWASP" in your favorite search engine)

Next, you need next to configure the browser to start using ZAP as a proxy. On Firefox, indicate "`about:preferences`" in the URL field:



General

Startup

□ Restore previous session

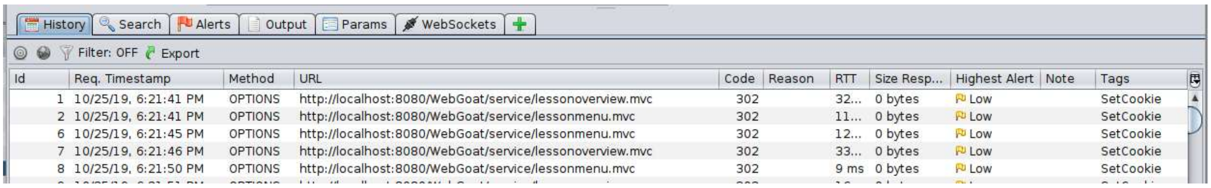
□ Warn you when quitting the browser

□ Always check if Firefox is your default browser

Firefox is not your default browser

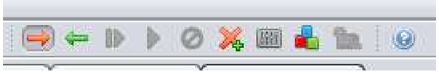
At the end of the "`General`" page, select in the region of Network Settings the "`Settings`". On the new window, choose "`Manual proxy configuration:`" and indicate as HTTP Proxy "`localhost`" and as Port "`8088`". Additionally, select "`Also use this proxy for HTTPS`" and remove from the "`No Proxy for:`" field all information.

Now, if you go to the History tab of ZAP, you will start seeing the messages that are exchanged automatically between WebGoat and the browser (e.g., keepalive messages).

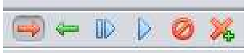


Id	Req. Timestamp	Method	URL	Code	Reason	RTT	Size Resp...	Highest Alert	Note	Tags
1	10/25/19, 6:21:41 PM	OPTIONS	http://localhost:8080/WebGoat/service/lessonoverview.mvc	302		32...	0 bytes	Low		SetCookie
2	10/25/19, 6:21:41 PM	OPTIONS	http://localhost:8080/WebGoat/service/lessonmenu.mvc	302		11...	0 bytes	Low		SetCookie
6	10/25/19, 6:21:45 PM	OPTIONS	http://localhost:8080/WebGoat/service/lessonmenu.mvc	302		12...	0 bytes	Low		SetCookie
7	10/25/19, 6:21:46 PM	OPTIONS	http://localhost:8080/WebGoat/service/lessonoverview.mvc	302		33...	0 bytes	Low		SetCookie
8	10/25/19, 6:21:50 PM	OPTIONS	http://localhost:8080/WebGoat/service/lessonmenu.mvc	302		9 ms	0 bytes	Low		SetCookie

Now to intercept the requests performed from the browser with ZAP, set the button "`Set break on all requests`" (the green right arrow). If the right arrow is orange, the intercept request operation is already set.



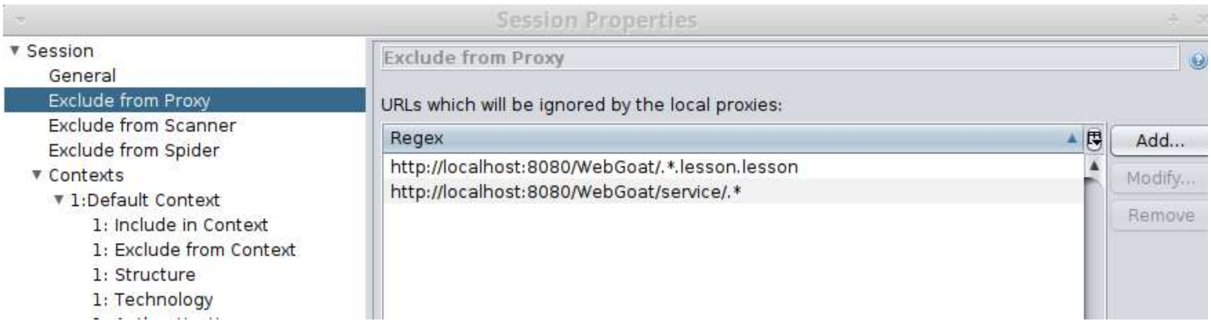
After the break on a request, you can modify the contents of the message, and then you can send it to the web application by using one of the two buttons to submit the request (the two blue arrows next to the orange/green right arrows):



Similarly, you can intercept the response from the Web Server with ZAP, by pressing the "`Set break on all responses`" (the green/orange left arrow).

Sometimes, you want to experiment multiple times to change a request until you find the right combination of parameters. You can go to the History pane and select the message that you want to experiment with. Then, right-click on the `History pane -> Open/Resend with request editor …` and then a new window appears. Here, you can see the request and response, and modify the request, and resend it (and observe the corresponding response).

Some of the messages may be unnecessary for our tasks, and therefore, we can try to hide them from our analysis. To do so, right-click on the `History pane -> Exclude from -> Proxy`, and then a new window appears. In this window, you may add the following, and then you can say OK:



Exclude from Proxy

URLs which will be ignored by the local proxies:

Regex
http://localhost:8080/WebGoat/.*.lesson.lesson
http://localhost:8080/WebGoat/service/.*

When you finish using ZAP, you need to change again the Network Settings configuration of Firefox to "`Use system proxy settings`".

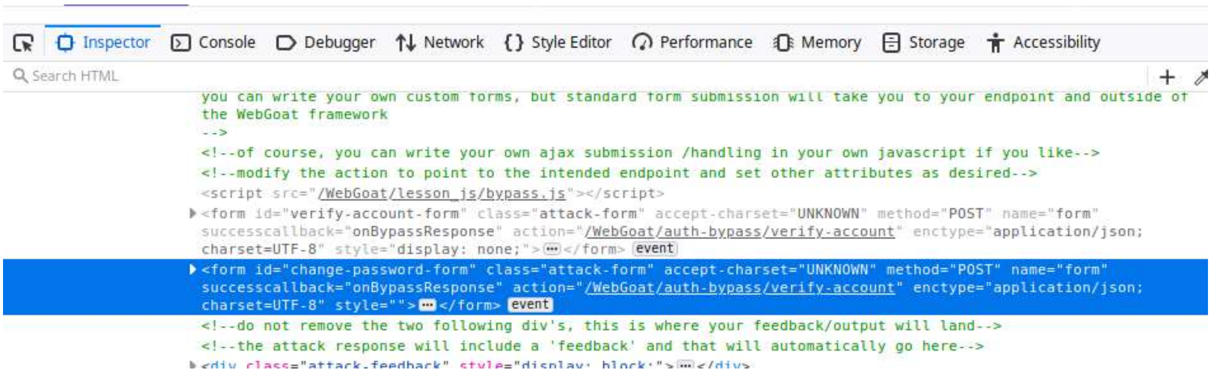
OTHER ISSUES

There is a message in Firefox that appears on every web page: *"you must log in to this network before you can access the internet*". Create a new tab and `about:config` into the address bar, press Enter and accept any warning, then copy and paste the following into the search box `network.captive-portal-service.enabled` Now on the search result, click the Toggle button at the right-hand end to change from True to False. You may need to restart Firefox to implement the change.

Browser Developer Tools

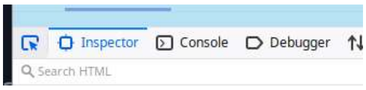
You can use the browser developer tools to perform various operations on the web page. In order to start the developer tools support, you just need to right-click on the part of the web page that you are interested in and then select `Inspect`.

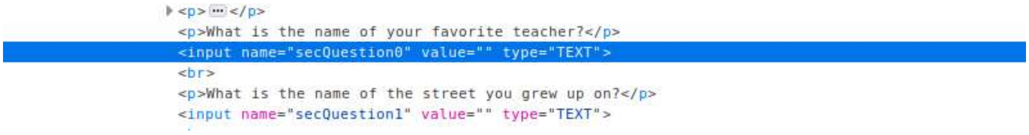
For example, you want to change the name of a form field. First, go to the Inspector tab:



```
you can write your own custom forms, but standard form submission will take you to your endpoint and outside of
the WebGoat framework
-->
<!--of course, you can write your own ajax submission /handling in your own javascript if you like-->
<!--modify the action to point to the intended endpoint and set other attributes as desired-->
<script src="/WebGoat/lesson_js/bypass.js"></script>
▸<form id="verify-account-form" class="attack-form" accept-charset="UNKNOWN" method="POST" name="form"
successcallback="onBypassResponse" action="/WebGoat/auth-bypass/verify-account" enctype="application/json;
charset=UTF-8" style="display: none;">⋯</form> event
▸<form id="change-password-form" class="attack-form" accept-charset="UNKNOWN" method="POST" name="form"
successcallback="onBypassResponse" action="/WebGoat/auth-bypass/verify-account" enctype="application/json;
charset=UTF-8" style="">⋯</form> event
<!--do not remove the two following div's, this is where your feedback/output will land-->
<!--the attack response will include a 'feedback' and that will automatically go here-->
▸<div class="attack-feedback" style="display: block;">⋯</div>
```

Then, you can select and see the HTML code corresponding to a field in the form by first clicking on the button to the left of the Inspector tab and then pressing the field on the web page.





```
▸ <p>⋯</p>
  <p>What is the name of your favorite teacher?</p>
  <input name="secQuestion0" value="" type="TEXT">
  <br>
  <p>What is the name of the street you grew up on?</p>
  <input name="secQuestion1" value="" type="TEXT">
```

Next, you change the name of the field by right-clicking on the HTML corresponding to the field and selecting `Edit as HTML`. After you finish changing, just press on some other part of the developer tools for the change to take effect.

Another tab that is interesting to experiment with is the Network. Here, you can see the various messages that are being exchanged together with their content.