The Challenge

In this activity, you will assume the role of a pen tester hired by a bank to test the security of the bank‚Äôs authentication scheme, sensitive financial data, and website interface.

#### Lab Environment

We'll use the \*\*Web Vulns\*\* lab environment. To access it:

- Log in to the Azure Classroom Labs dashboard.

- Find the card with the title \*\*Web Vulns\*\* or \*\*Web Vulnerability and Hardening\*\*.

- Click the monitor icon in the bottom-right.

- Select \*\*Connect with RDP\*\*.

- Use Credentials (azadmin:p4ssw0rd\*)

- The lab should already be started, so you should be able to connect immediately.

- Refer to the [lab setup instructions](https://cyberxsecurity.gitlab.io/documentation/using-classroom-labs/post/2019-01-09-first-access/) for details on setting up the RDP connection.

Once the lab environment is running, open the HyperV manager and make sure that the OWASPBWA and Kali box is running.

- Then, login to the Kali VM and navigate to the IP address of the OWASPBWA machine.

- Click the option for 'WebGoat' and start the WebGoat app.

- Use the credentials: `guest`:`guest`

On the bottom of the left side of the screen, click on `Challenge` and then choose `The Challenge`.

\*\*Note:\*\* A common issue with this lab is the Challange activity failing to start successfully. Hit the `Restart the Lesson` button in the top right if you get an error starting the activity.

### The Challenge Instructions

#### Challenge #1

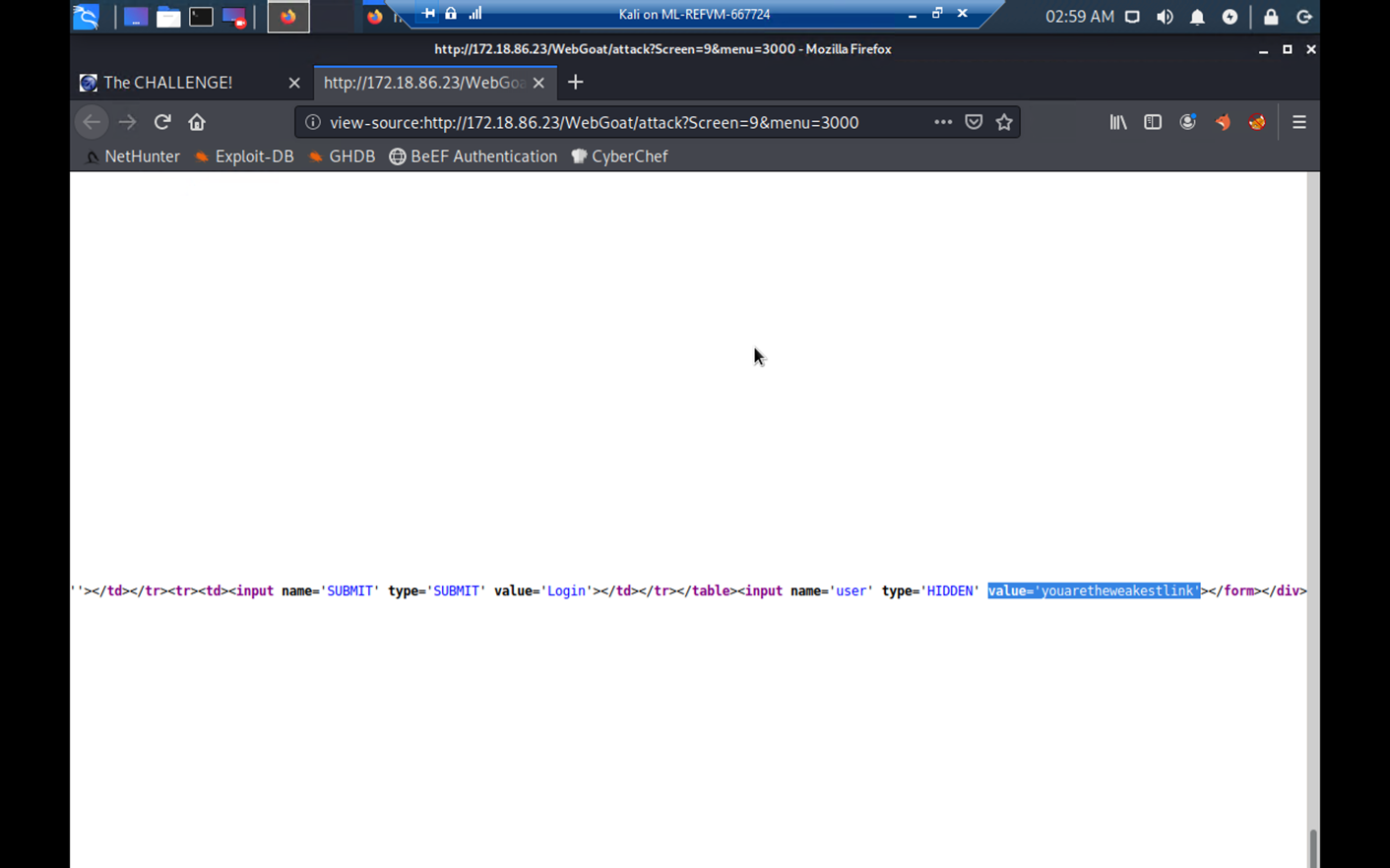
Your first mission is to break the authentication scheme. There are a number of ways to accomplish this task.

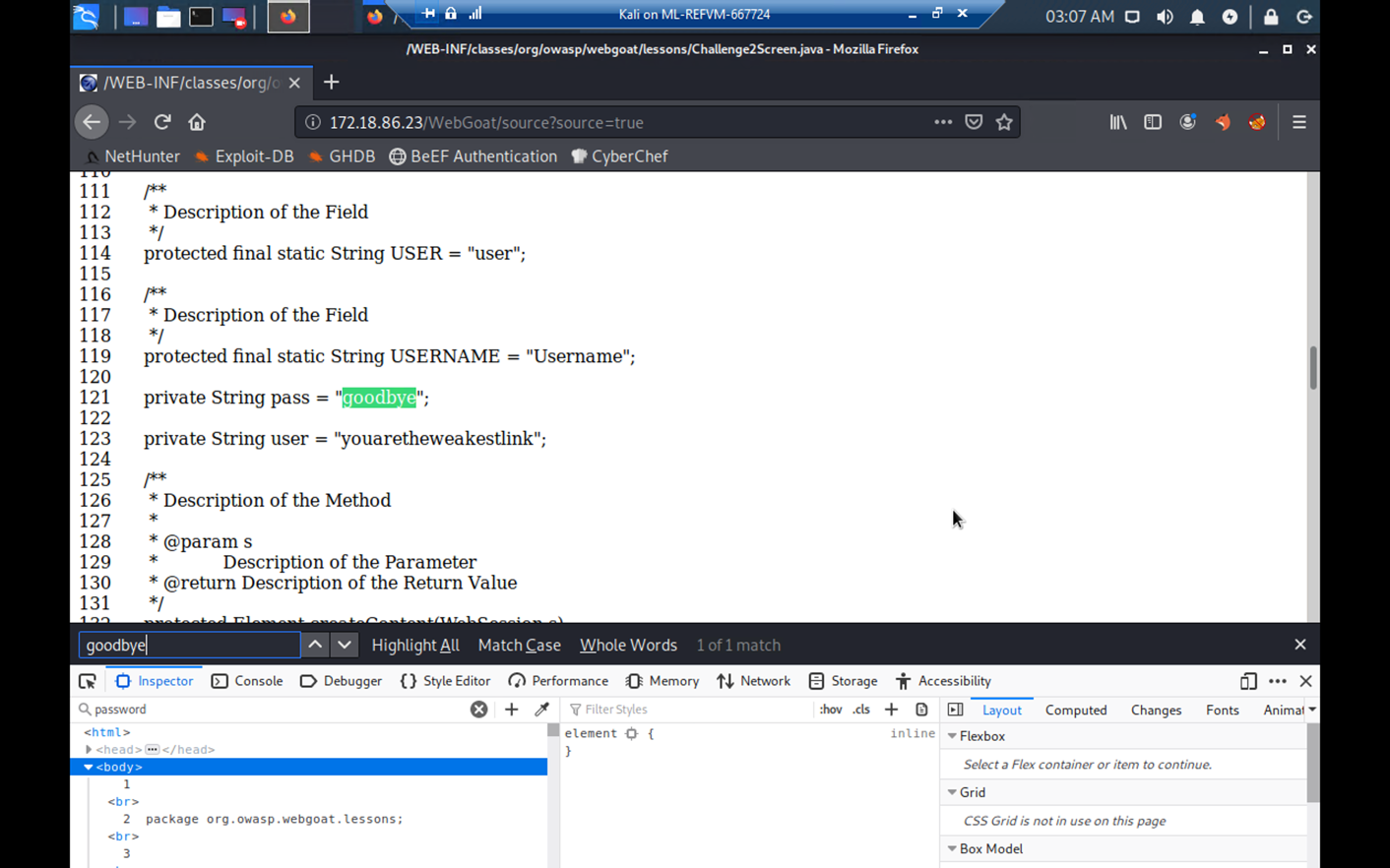
- \*\*Hint #1\*\*: Sometimes, form fields are shy!

- \*\*Hint #2\*\*: Find the hidden JavaScript.

- \*\*Hint #3\*\*: You can appened `source?source=true` to the URL to read the source code.

After completing the first challenge, you will be provided with an option to continue to the next challenge.





#### Challenge #2

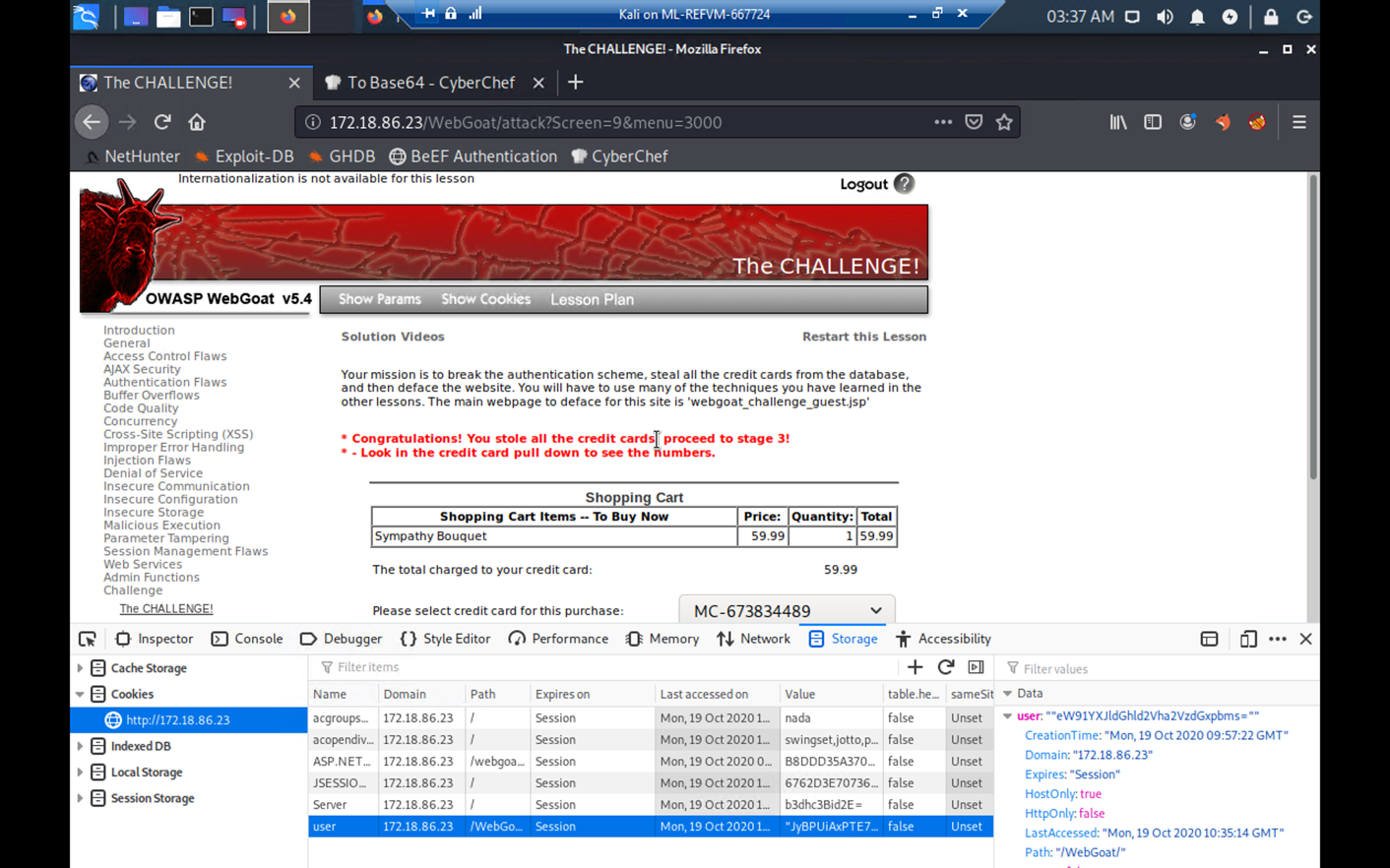
Next, steal all of the credit card numbers from the database.

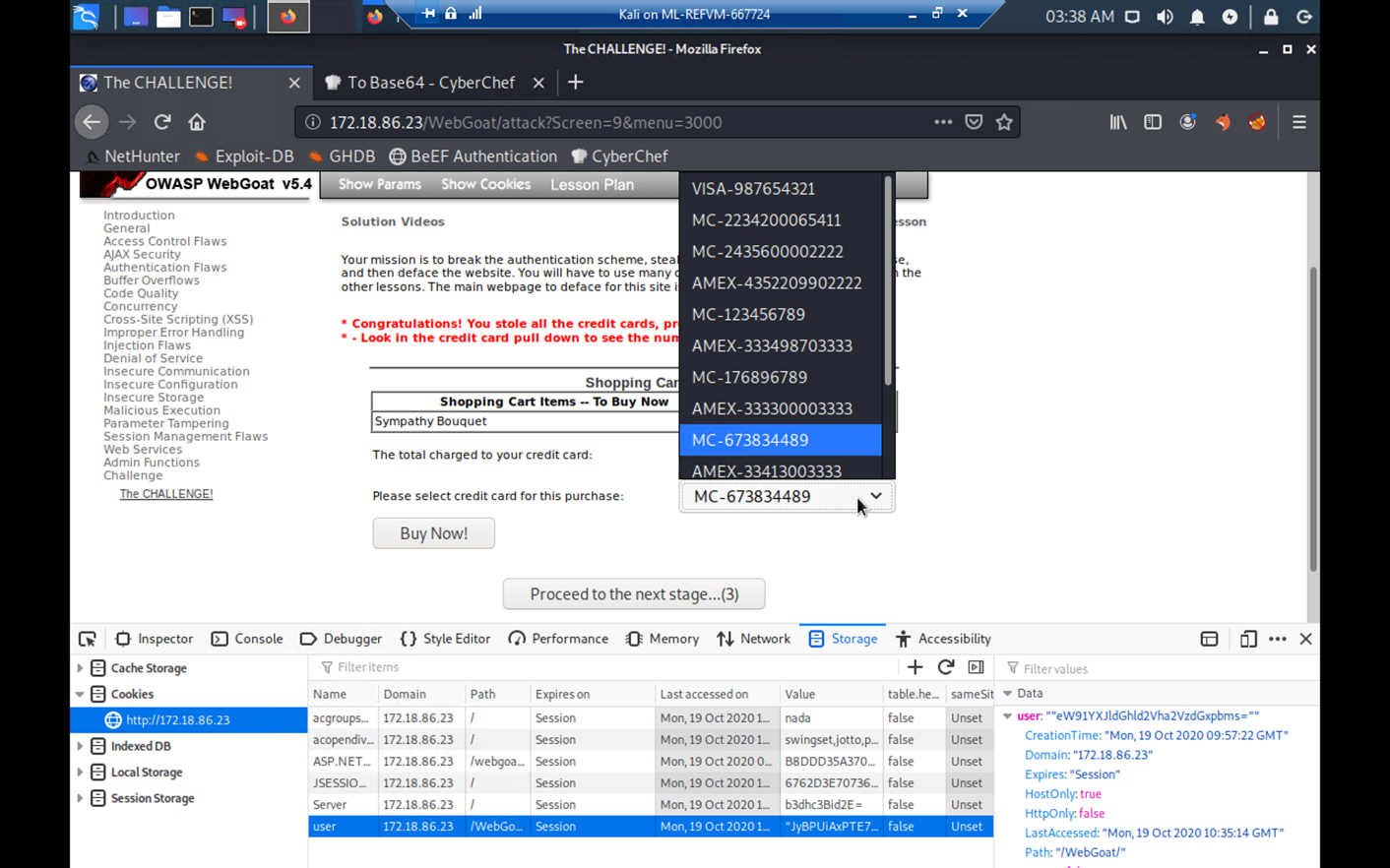
\*\*Hint #1\*\*: Sometimes cookies wear different clothes to change their appearances.

- \*\*Hint #2\*\*: Break your way into the conversation and inject your own ideas.

After completing the second challenge, you will be provided with an option to continue to the next challenge.







#### Challenge #3

Your final act is to deface the website using command injection. Follow the walkthrough below to help you get started.

- After completing the second challenge, you will be provided with an option to continue to the next challenge.

![cracked credit cards](Images/credit\_cards-cracked.png)

- There should be two webpages at the bottom of the window. The one on top is the original, and the one on the bottom is the defaced webpage.

![original webpage](Images/original\_defaced.png)

- Start Foxy Proxy (WebScarab) to send all GET/POST requests from Firefox to the WebScarab proxy intercept.

![Foxy Proxy](Images/foxy\_proxy\_scab.png)

- Click \*\*TCP\*\* and then the \*\*View Network\*\* button and send the request to WebScarab.

![View Network](Images/view\_network\_tcp.png)

- The WebScarab window will open.

- In the \*\*URL Encoded\*\* tab, find the \*\*File\*\* and \*\*Value\*\* form fields.

- This is where you will perform your command injection.

![File Field](Images/webscarab\_file\_value\_field.png)

- Next, perform a test and see if this shell is vulnerable to command injection.

- Type the following command into the Value field: `tcp && whoami && pwd`.

- \*\*Note:\*\* Windows users can type `tcp && dir`. `dir` will return the directory as proof of vulnerability.

- Click \*\*Accept Changes\*\*.

![whoami](Images/whoami\_pwd\_image.png)

- On the next window, click \*\*Accept Changes\*\* twice.

![accept](Images/webscarab\_2nd\_window.png)

- Scroll to the bottom of the \*\*Current Network Status\*\* window and observe the results for both of the `whoami` and `pwd` commands.

![whoami & pwd](Images/whoami\_pwd.png)

- The results show that we are the root user and our current working directory is `/var/lib/tomcat6`.

- This verifies the vulnerability, so proceed to the next step.

- Next, we'll locate the `webgoat\_challenge\_guest.jsp` file.

- Type the following command: `tcp && cd / && find . -iname webgoat\_challenge\_guest.jsp`.

- \*\*Note\*\*: Windows users will need to type: `tcp && dir /s 'webgoat\_challenge\_guest.jsp'`

![find command](Images/webscarab\_find\_command.png)

- The absolute path is: /`.

![absolute path](Images/webscarab\_abolute\_path.png)

- Remember, our present working directory is `/var/lib/tomcat6`. Therefore, the relative path is `webapps/WebGoat/webgoat\_challenge\_guest.jsp`.

\*\*Now it's your turn\*\*

- Now that we know where the webpage is, your task will be to deface the website. Keep in mind the following:

\* Use \*\*WebScarab\*\* to perform command injection.

\* When performing command injection, you will need to select a field that WebScarab can return commands to. These fields are typically located in a drop down.

\* You will also need to locate and edit the the webpage's source code: `webgoat\_challenge\_guest.jsp`

\* Your final command will:

\* Change to the location of the `webgoat\_challenge\_guest.jsp` file.

\* \*\*and\*\* echo `You've been hacked.` to the `webgoat\_challenge\_guest.jsp` file.

