Assignment 04- HIWA Custom Labtainer

Horribly Insecure Web Application Lab report

Department of computer science

Adelphi university

CSC – 380 -001 Computer and Network Security

Professor. Sung Kim

By – Dikshant Kakadiya

Date – April 3rd 2024

# Introduction

# We'll be examining an insecurely designed web application in this lab, and our tasks include finding the weaknesses and taking advantage of them. We will aim to complete 11 challenges in this lab, which will serve as prompts for us to investigate, identify, and exploit the vulnerabilities, accordingly. There are three sections to the challenges. The first one only requires us to examine the webpage and search for any potential security holes. In the following phase, we attempt to exploit those vulnerabilities by active assault, and in the last phase, we are expected to use advanced exploit techniques. Also, this lab is focused on simulating an attack on a web application.

# Objective

In this lab, you will analyze a web application, identify vulnerabilities, and exploit them.

# Preparation

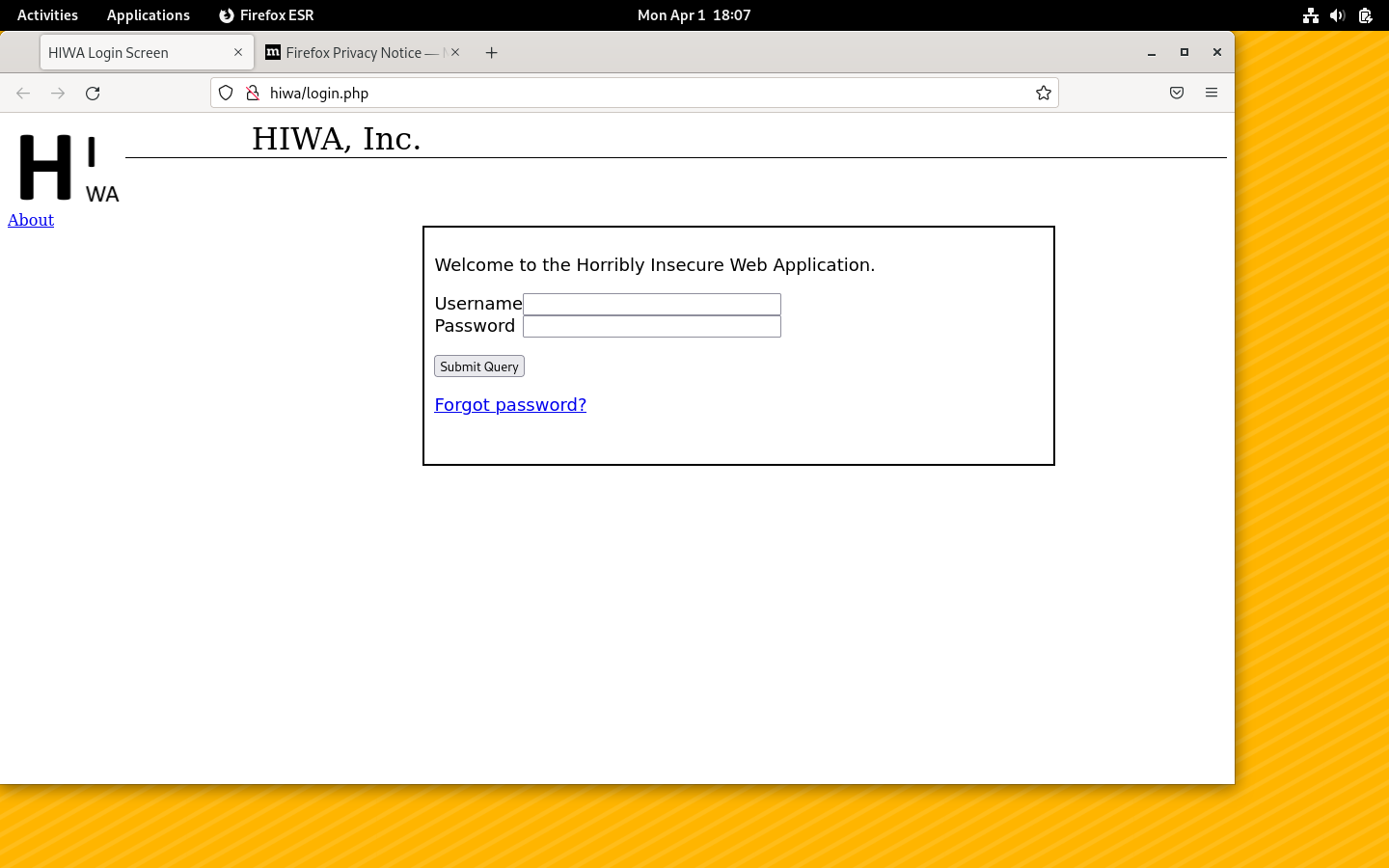
**Step 1**. Start your Labtainers virtual machine.

**Step 2**. Launch the hiwa labtainer by issuing the command

labtainer hiwa

**Step 3**. Open Firefox ESR by clicking on the Applications menu. Then open

<http://hiwa/login.php>



**The above image shows the set-up of HIWA lab**

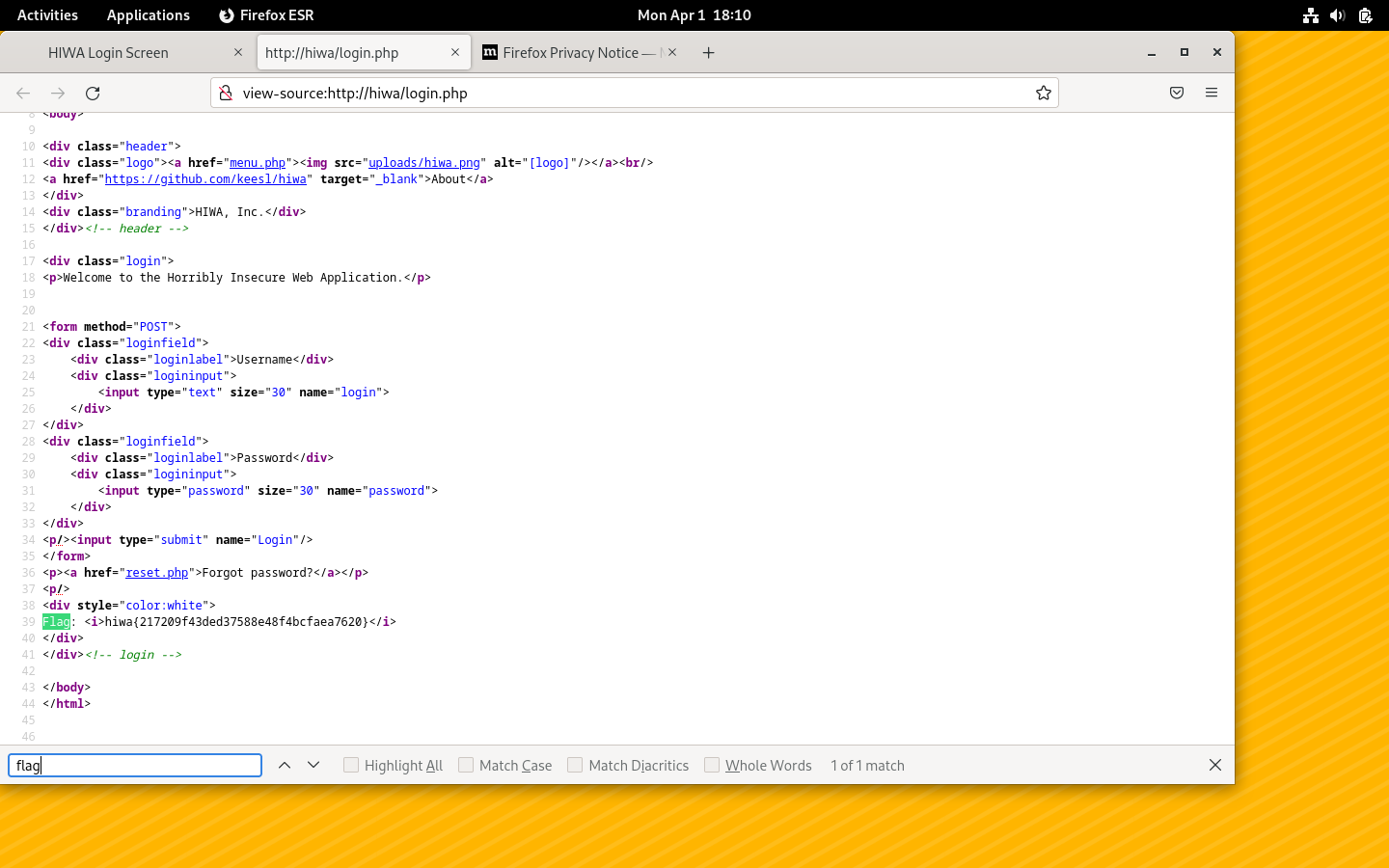
# Stage 1: Looking Around

In this lab packet, you will simulate participating in a Capture The Flag (CTF). A CTF is a gamified simulation of real-world security scenarios. You are presented with a number of challenges. The goal is to find “flags”, which (in this game), are simply text strings. The strings always have the format hiwa{…} (where the …) change for each challenge.

For each challenge, record a screenshot with the flag in the lab report. When asked, explain how you found the flag.

**Challenge 1. Hidden in plain sight.**

Most good clues are hidden in plain sight! Take a closer look at the web page and its HTML source.



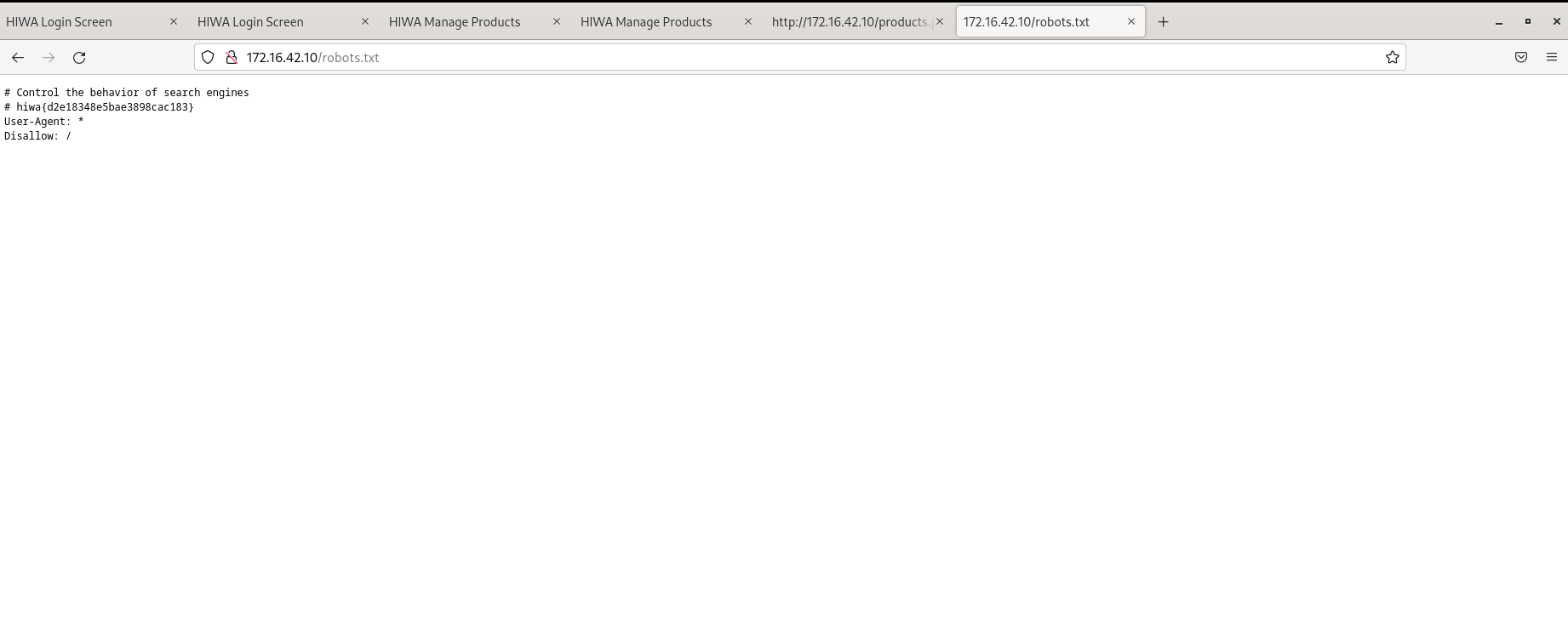
**After setting up the lab we pressed Control +U it open-source window as showed in above image. After looking at the code we found a flag. we can fund the flag by clicking control +f to open up find box.**

**Challenge 2**. Mr. Robot

Search engines crawl whatever web sites they can find. If you don't want to have your websites indexed, you can use a special configuration file posted to your web server to do that.

However, where you don't want me to look, I'll gladly poke around!

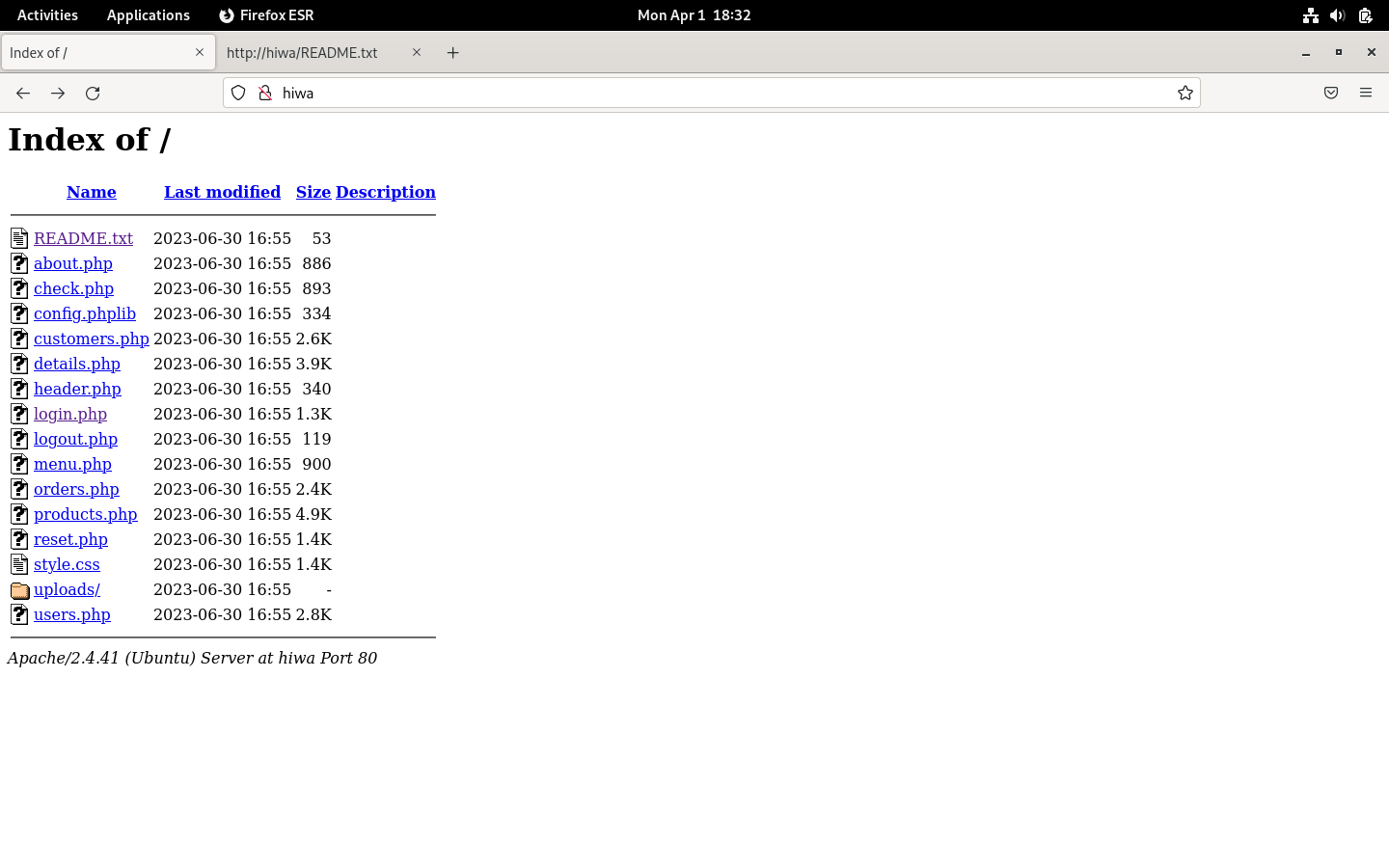
Hint: the file is called robots.txt and, if it exists, will always be located in the root directory of the web site.



**To complete the task we typed “robots.txt” in URL. Let to window as shown in above window and we found a flag.**

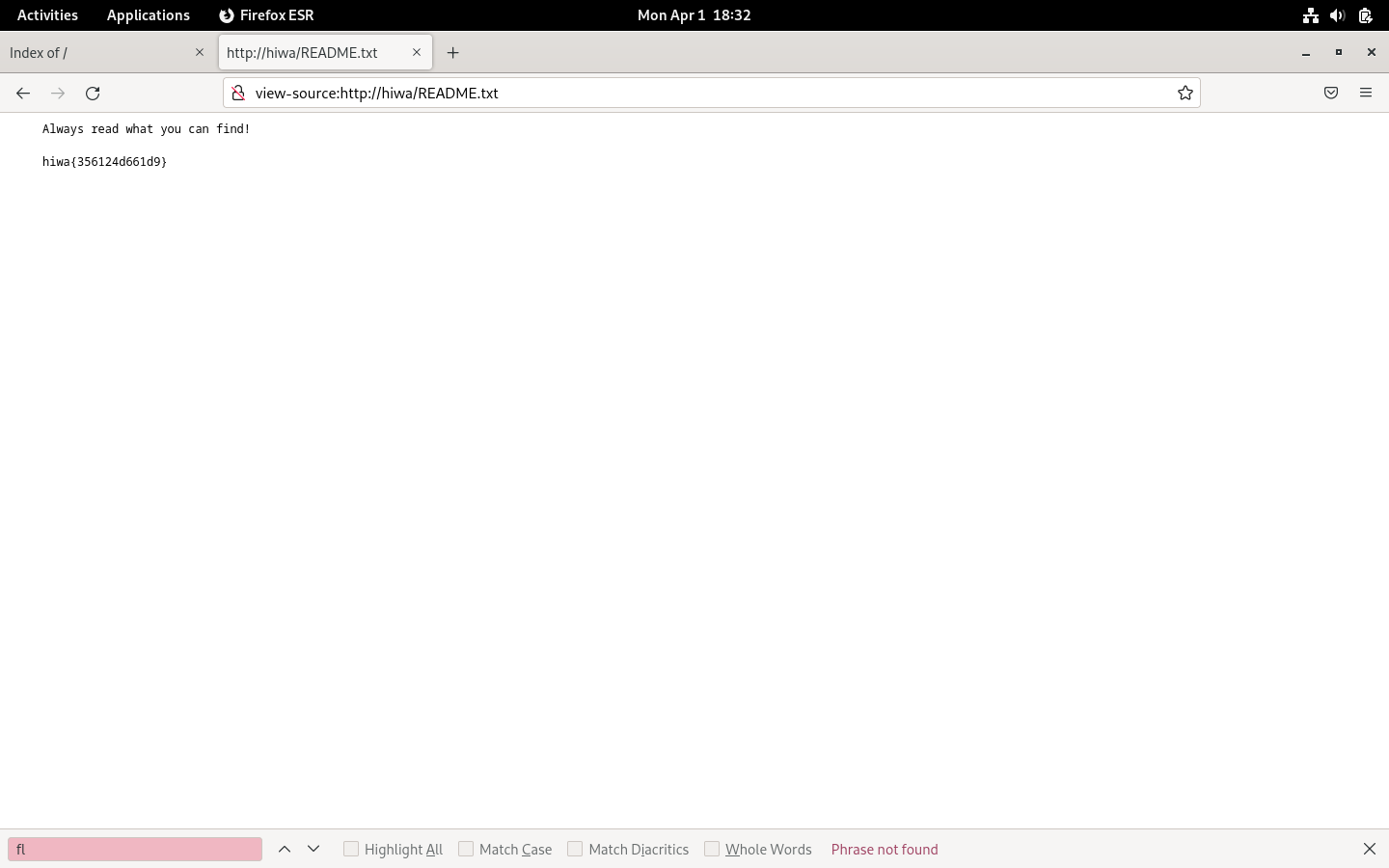
**Challenge 3**. Read That Fine Manual (RTFM)!

The URL box of the browser is fully under your control! By playing around with the filename (the part after the first /), you can sometimes find interesting information. Remove everything after the / in the URL, and then hit ENTER to load the page.

What is the flag that you find in the first file?

**For this challenges we needed to do 2 steps.**

**Step 1 was removing the “/login.php” for the URL. And it will lead to a window that is shown above.**



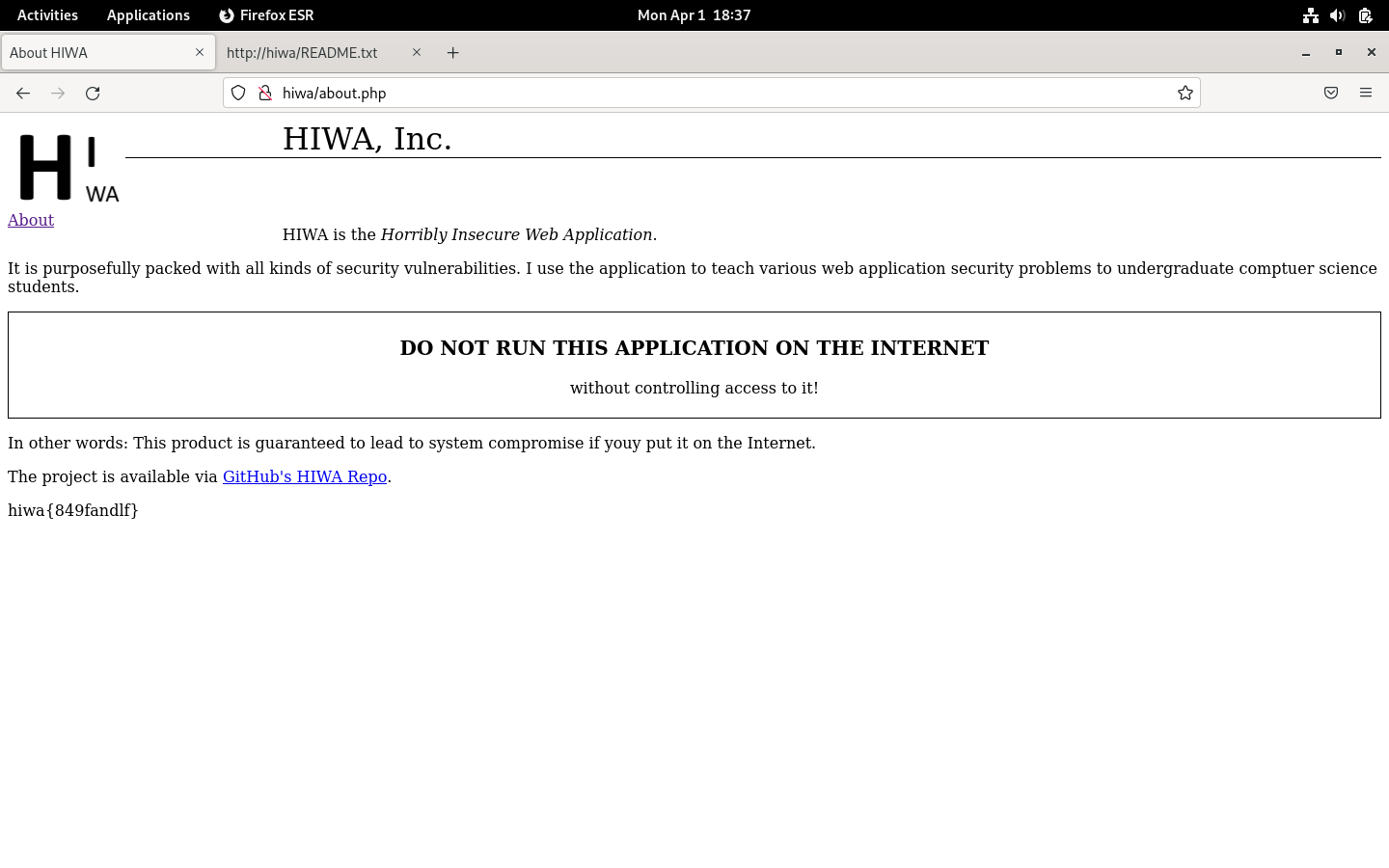
**Now we will look in to step2. Is opening the file called redme.txt by clicking on file from the menu for step 1. Also the content of readme.txt file is shown above. Also we found a flag here too.**

**Challenge 4**. Turn over every stone

Good coding practices can help, but they can also hurt. Look closely, to see what access you might have that wasn't necessarily meant for you.

Hint: Look in a place where programmers often leave hints for themselves. You may have to use “View Source” again!

What is the flag stored in the programming library’s comments?

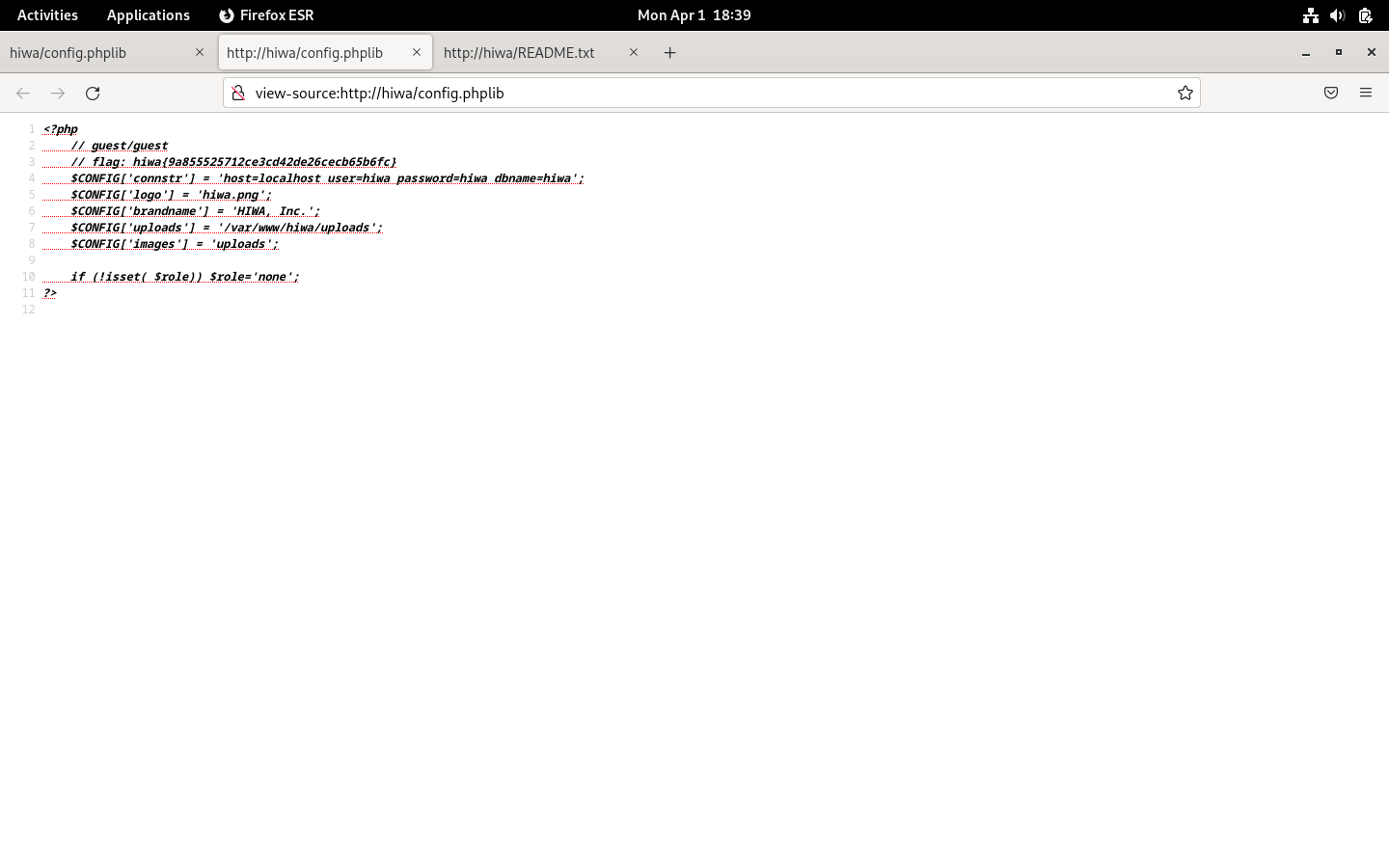


**Thus, continuing in the programmer's directory, we looked at the following file, “about.php”, and saw a hint of some kind that the programmer had left for themself. Thus, we discovered the flag at the bottom of the page when we opened that file. As shown in above image.**

**Challenge 5**. One More.

Just in case it isn’t abundantly clear yet. You should look at each and every page you can access. If it is readily available, inspect the rendered page, as well as its HTML source.

Find one more readily available flag.



**We examined the contents of each file in the root directory one by one for this task. However, after looking through the “config.phplib” file's components, we discovered a flag here too, as shown in above image.**

# Stage 2: Active attacks

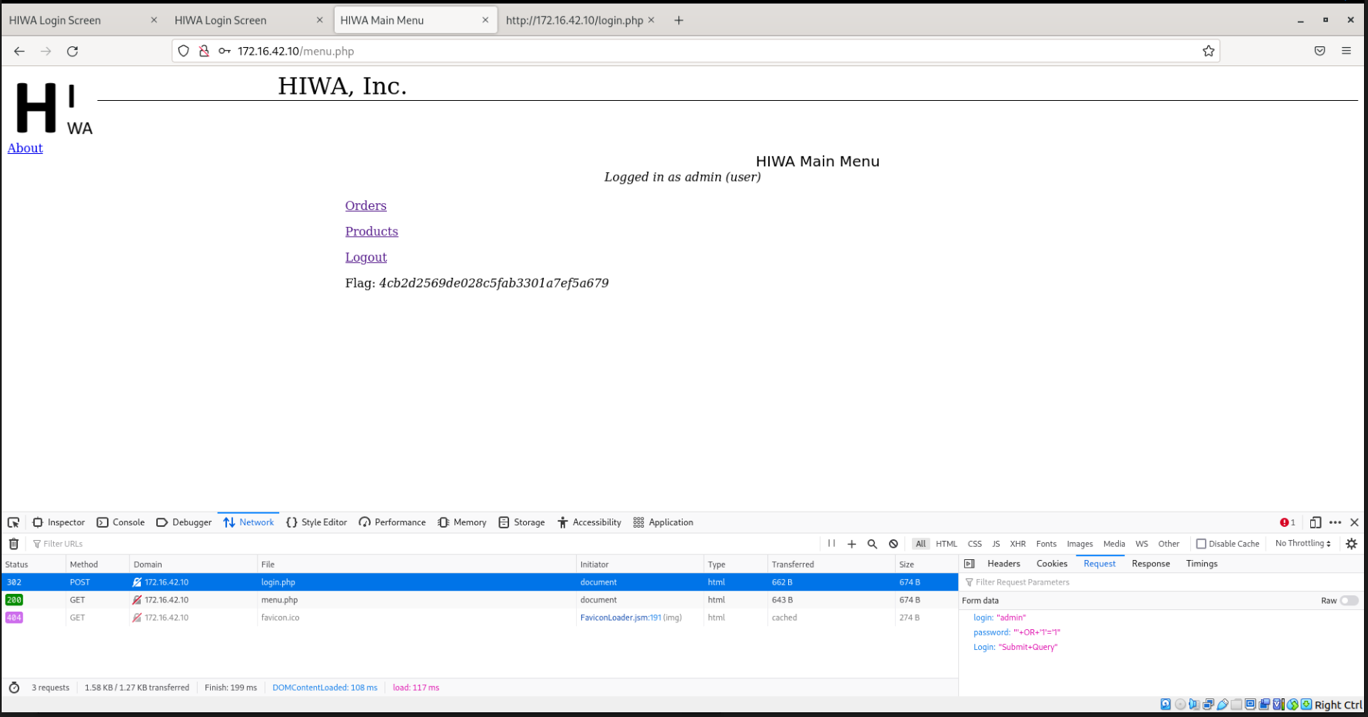
In the next stage, you’re going to gain access to the application. Return to <http://172.16.42.10/login.php>

**Challenge 6.** Jump the turnstile!

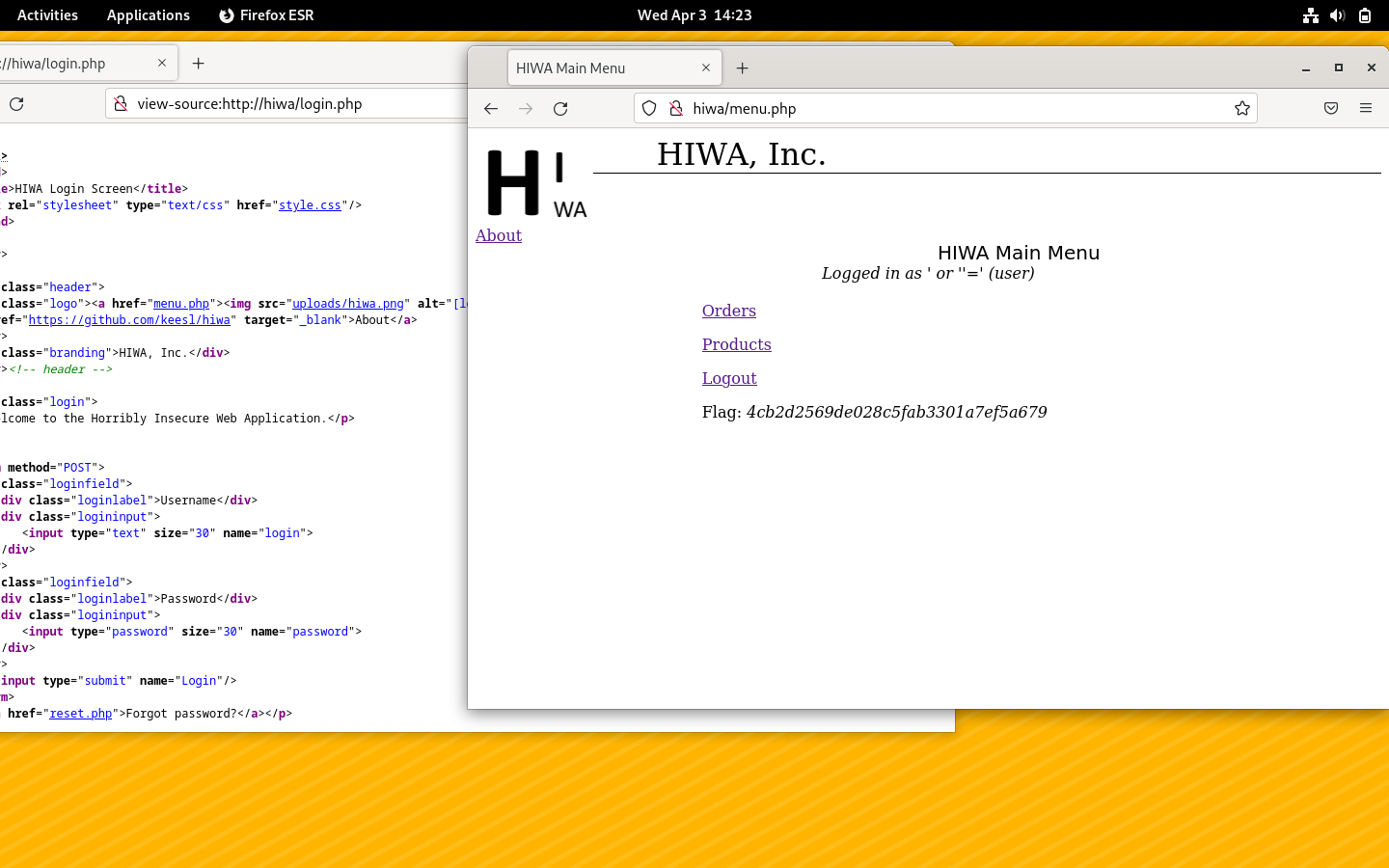
To go in through the front door is easy when you have a key. Or when the lock is broken. Can you gain access to the application, even if you don't have a password?

Hint 1: Input validation is often the root of all evil.

Hint 2: SQL injection is sometimes an effective technique to bypass authentication systems. Try a SQLi attack against the password field on the login page!



**In order to get past the authentication screen and into the HIWA system, we had to construct a SQL injection query specifically for this problem. We entered the SQL injection query “' OR '1'='1” in the user name fields. We were able to get access we seen I above picture.**

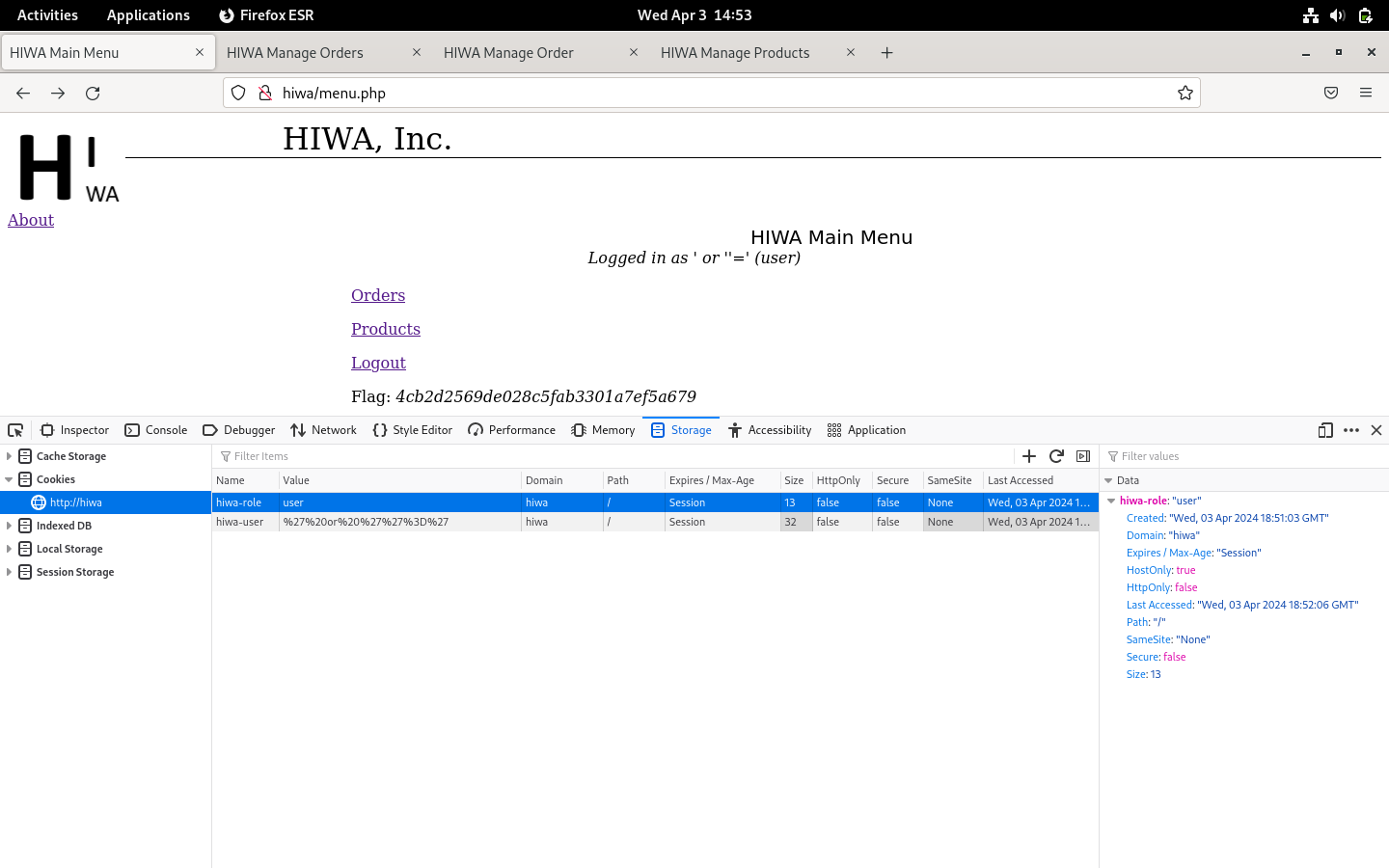
****

**In order to get past the authentication screen and into the HIWA system, we had to construct a SQL injection query specifically for this problem. We entered the SQL injection query “ ' or ''='” in the user name and password fields. We were able to get access we seen I above picture.**

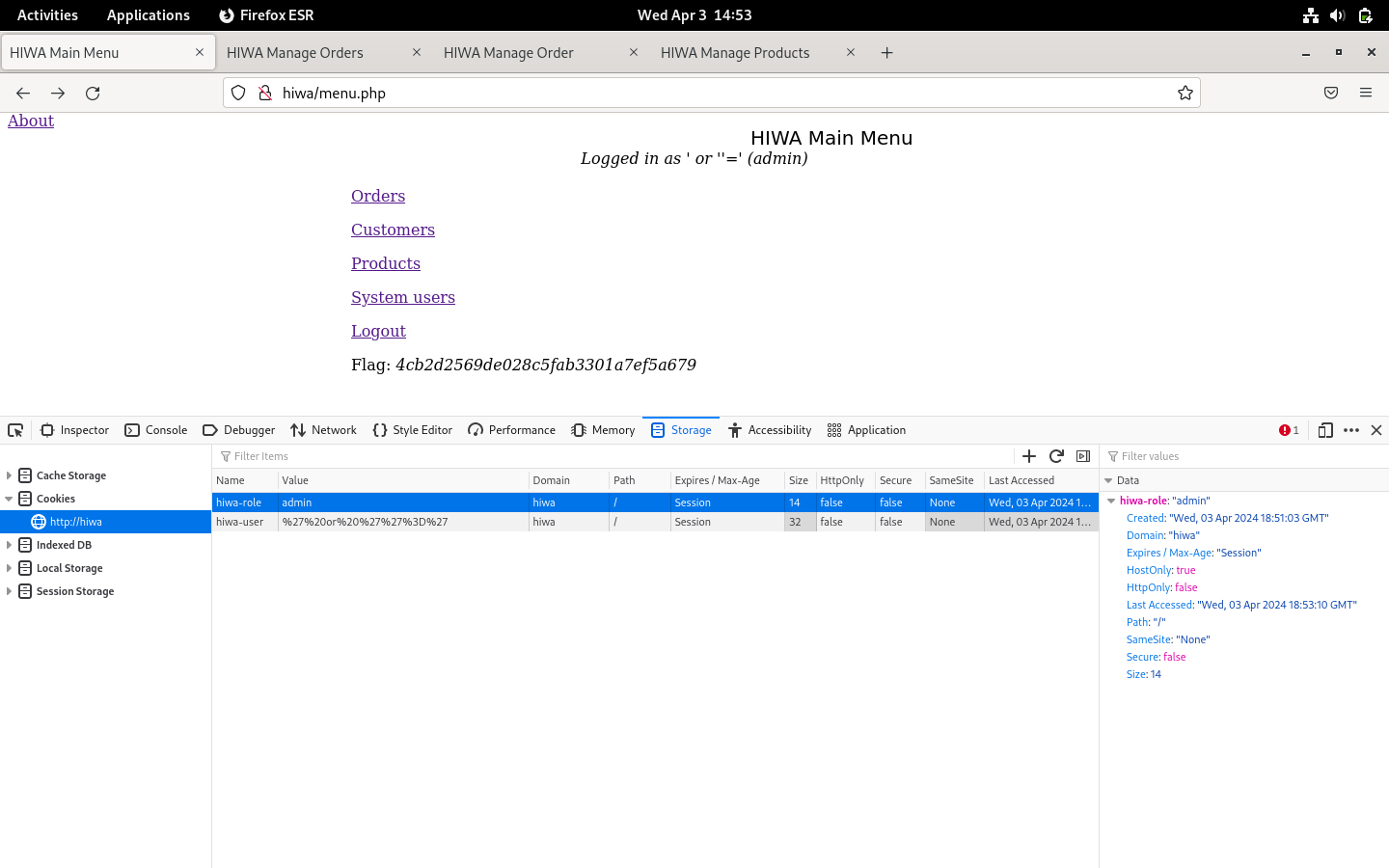
**Challenge 7**. Elevate Privileges

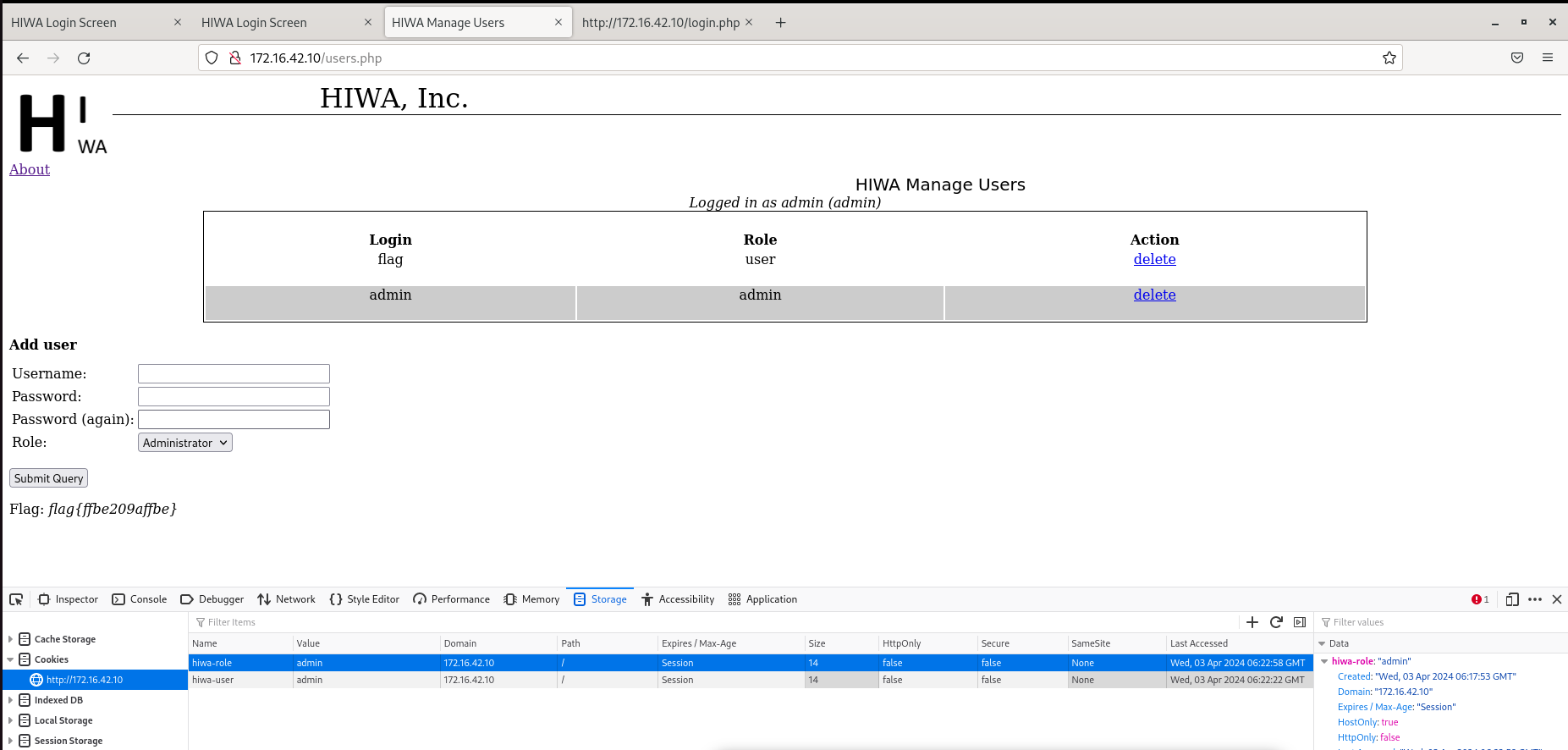
As you navigate through the HIWA system, you will notice that several scripts are not accessible to you. Find a way to access the restricted pages.

Hint: Web applications use cookies to maintain sessions, and sessions maintain the state of the applications. Cookies should never contain information that is potentially sensitive, since they can be manipulated by the user. If a web app is written insecurely, you may be able to influence its behavior by messing around with cookies!



**After we login to HIWA. We right clicked on the page and then hit the inspect (Q). then we click on storage and we found cookies tab. We see that in the above image. We can see that role HIWA-Role says User.**





**As we need to update the privileges form user to admin. We changed the user to admin in value table as seen in images above. We can conform it as HIWA-Role got updated to admin.**

**Challenge 8.** Logo

Change the logo of the HIWA application in such a way that all users of the application see the new version. Simply editing the HTML locally in your browser is NOT sufficient.

Note: there is no specific flag associated with this challenge. Just describe what you did! Hint: Look closely at the logo itself. Where is it stored? Can you control that location?

We see in the headers.php file that it uses the config.phplib file. Where there are two variables ‘images’ and ‘logo’. The images variable tells the directory and the logo variable tells the filename of the logo to display. What we can do is using ssh/ftp replace this logo in the uploads folder with another file under the same name and it will start displaying that new image file in the logo placeholder.

We can also go to the products page and upload an image file with the name hiwa.png and that will replace the logo on the server. We can use the same technique like in challenge 11 and also write an injection script to delete the hiwa.png file and then upload a new one if needed.

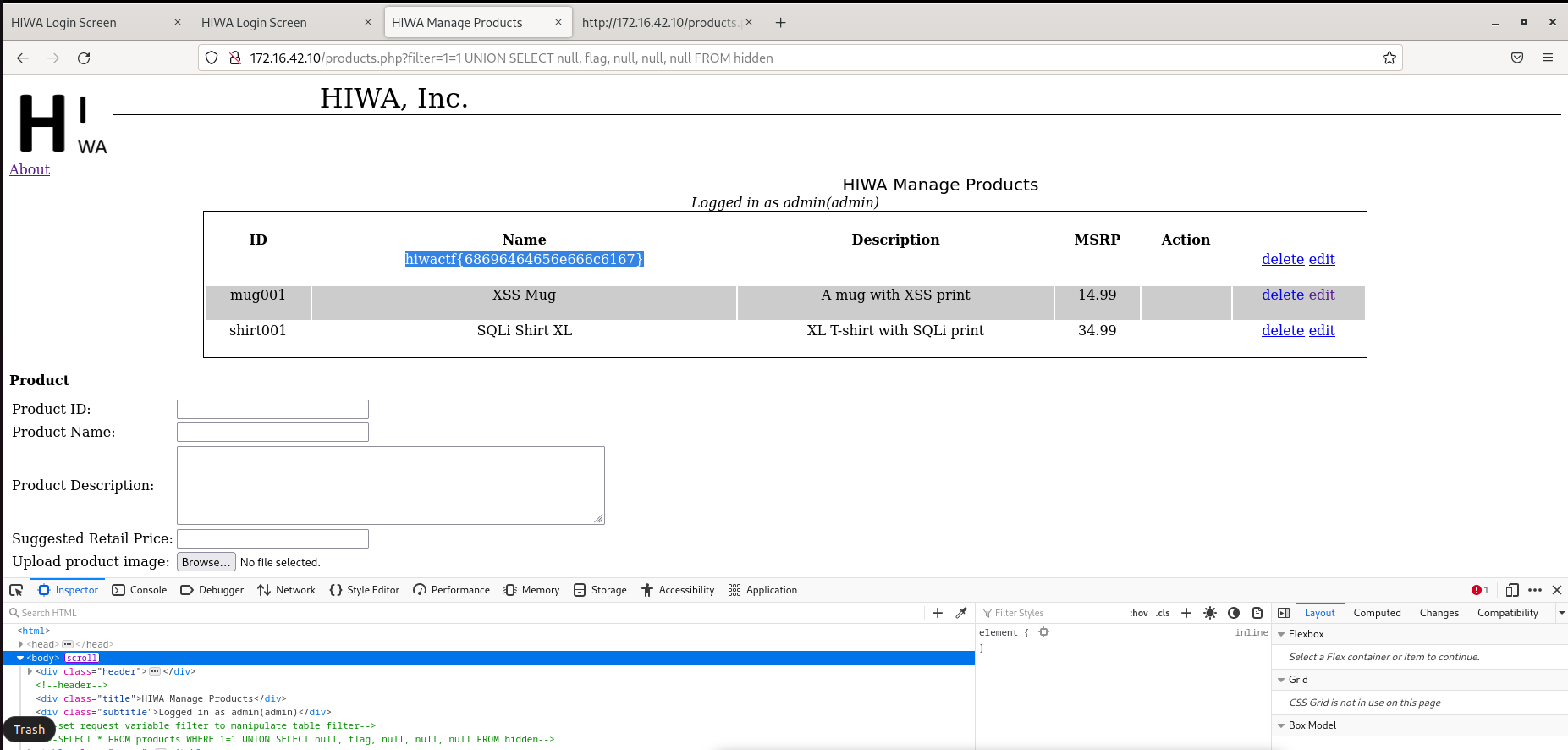
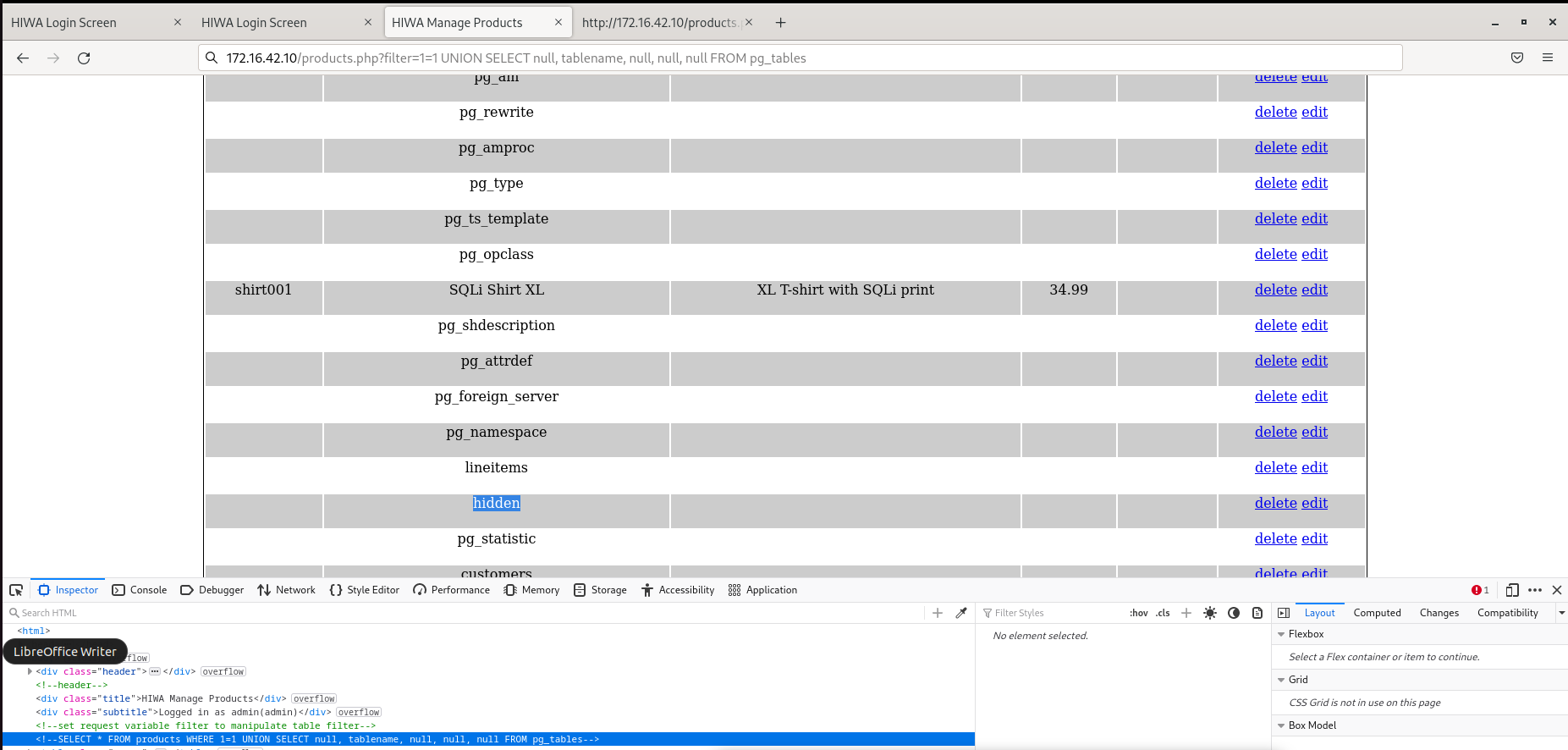
# Stage 3: Advanced Techniques

**Challenge 9**: Hidden

What is the flag in the hidden database table?

Hint 1: carefully view the HTML source of the products.php page

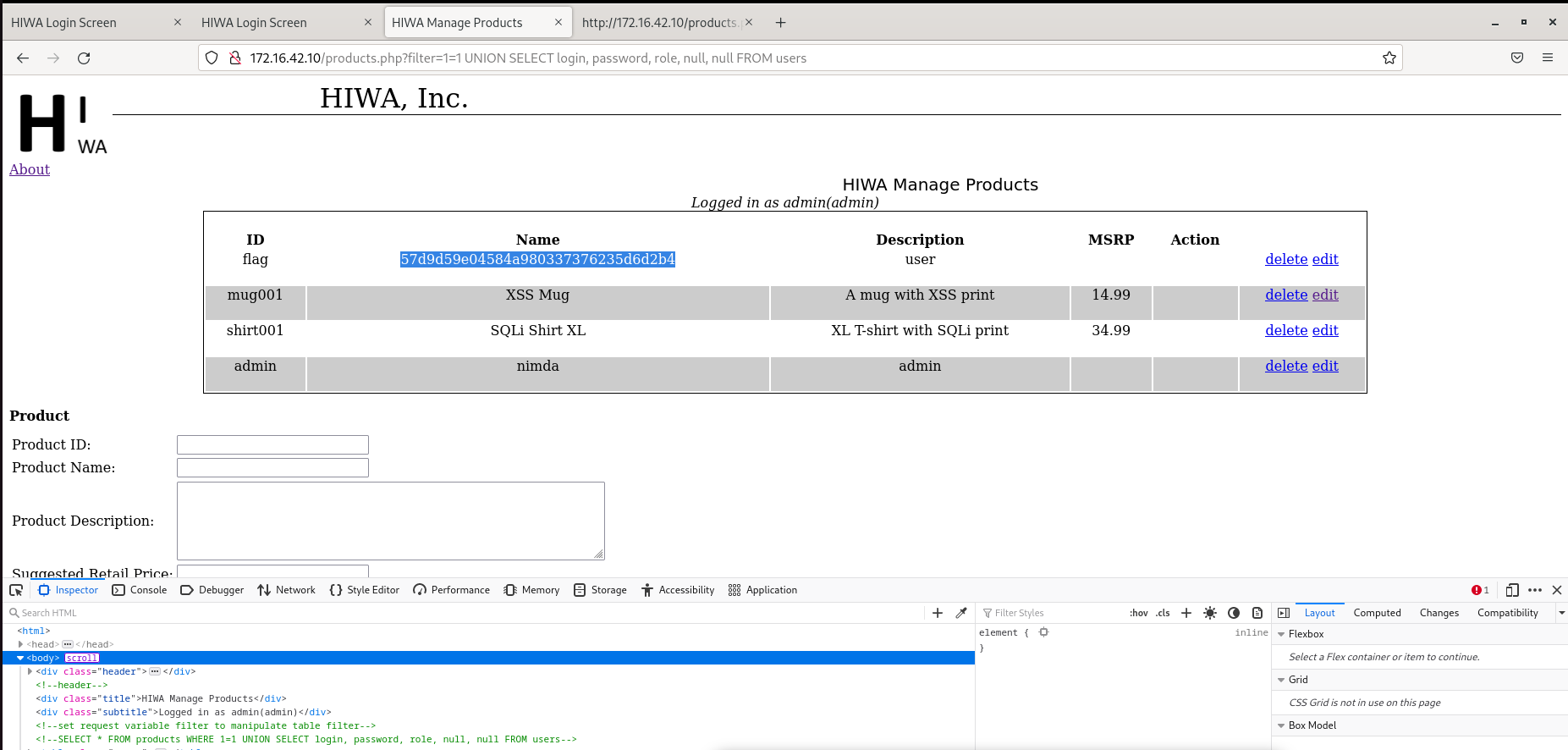
Hint 2: you’ll need to use a SQLi technique that lets you access tables you normally don’t have access to.



**Challenge 10**: Secrets...

What is the password of the 'flag' user in the HIWA application? Hint 1: carefully view the HTML source of the products.php page

Hint 2: you’ll need to use a SQLi technique that lets you access fields in tables you normally don’t have access to.

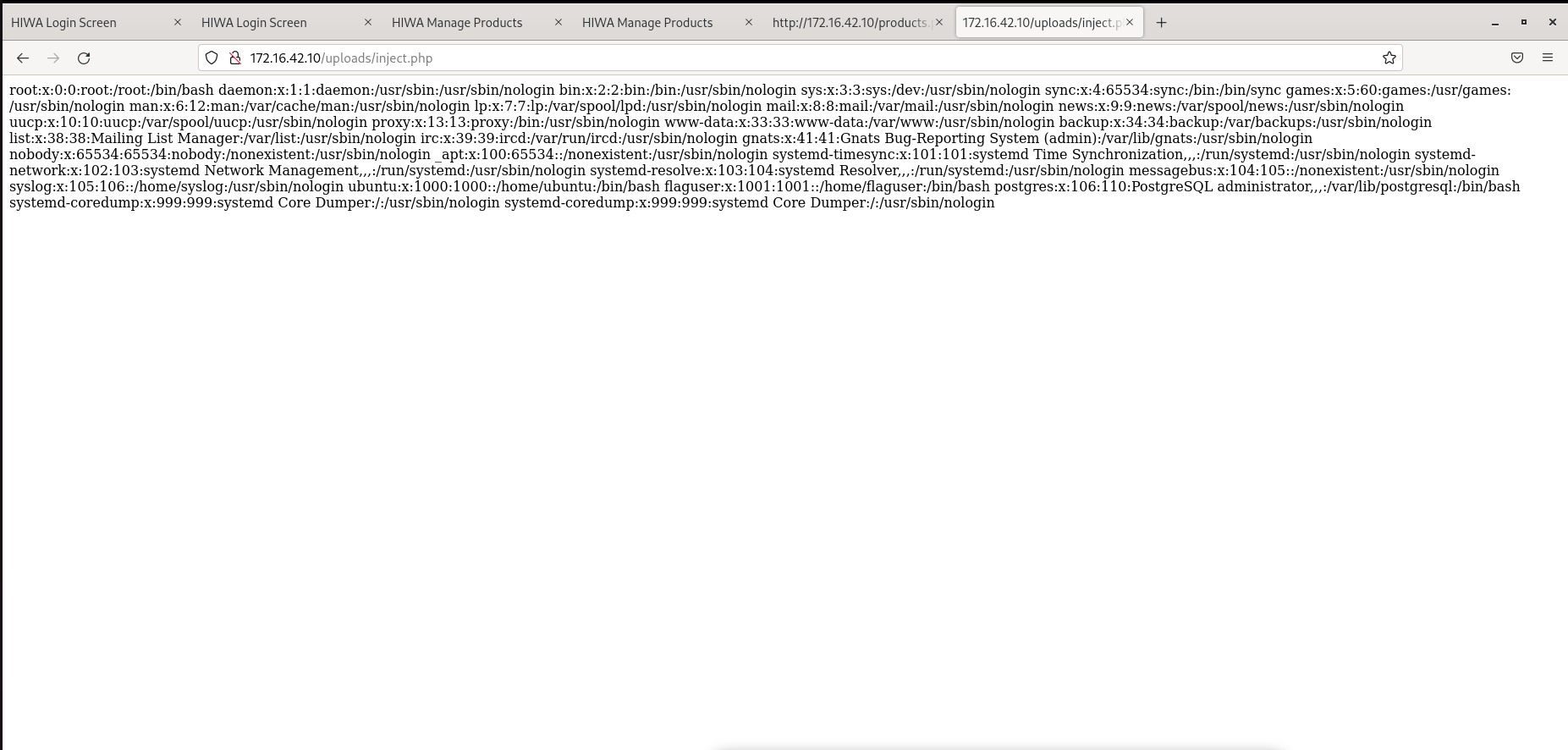


**Challenge 11**: Enumerate

Enumerate operating system users on the server running the application by exploiting a vulnerability in HIWA. The flag is in the home directory of the user who is labeled as having a flag.

Hint 1: We’re looking for operating system users. They are defined in /etc/passwd.

Hint 2: Can you run a shell command that shows you the contents of the password file?



**Background**

We followed the comprehensive and detailed HIWA lab instructions on the moodle for this lab, and we video that are listed on moodle page; it tells us exactly what to do. Other than that, we didn't make use of outside resources.

**Methodology/Results**

We looked at the instruction and flowed it step by step, we were able to attempt all 11 tasks. All of the results of this lab are documented with image of the work done ware added when needed.