Notebook 1
CS495 Web and Cloud Security
PDX | Winter 2022

Evan La Fleur

1.1 Setup

[Completed on January 5th, 2022]

The Setup section involved getting all accounts created and applications set up. This section was broken down into multiple sections which consisted of the following:

- Creating Slack Account and joining Slack channel
- Setting up Linux VM for Pentesting
- Gitlab Account creation
- Creation of Repository on Gitlab
    - Adding Professor and TA as collaborators
- Setting up Git with your VM
- Creating a PortSwigger Account to be able to complete CTFS
- Creation of an account on Google Cloud Platform
- Creation of two sinks in the Google Cloud Platform
    - Linking them to gitlab repo
- Creation of two virtual machines in Cloud Platform for the pentesterlab.com labs
    - Made sure vms were not running when not in use

Link to Gitlab: https://gitlab.com/elafleur1/w22websec-lafleur-evan

1.2 Web Programming

This assignment provided instruction on how to set up the python environment as well as walk through a basic program layout.

For my python setup I am going to start with using VSCode as my weapon of choice.

For these screenshots my Mcecs username is the name showing up in the terminal. I modified my terminal to just display my username in brackets([elafleur]).

1.2.3 Sequential Program

getUrls.py U

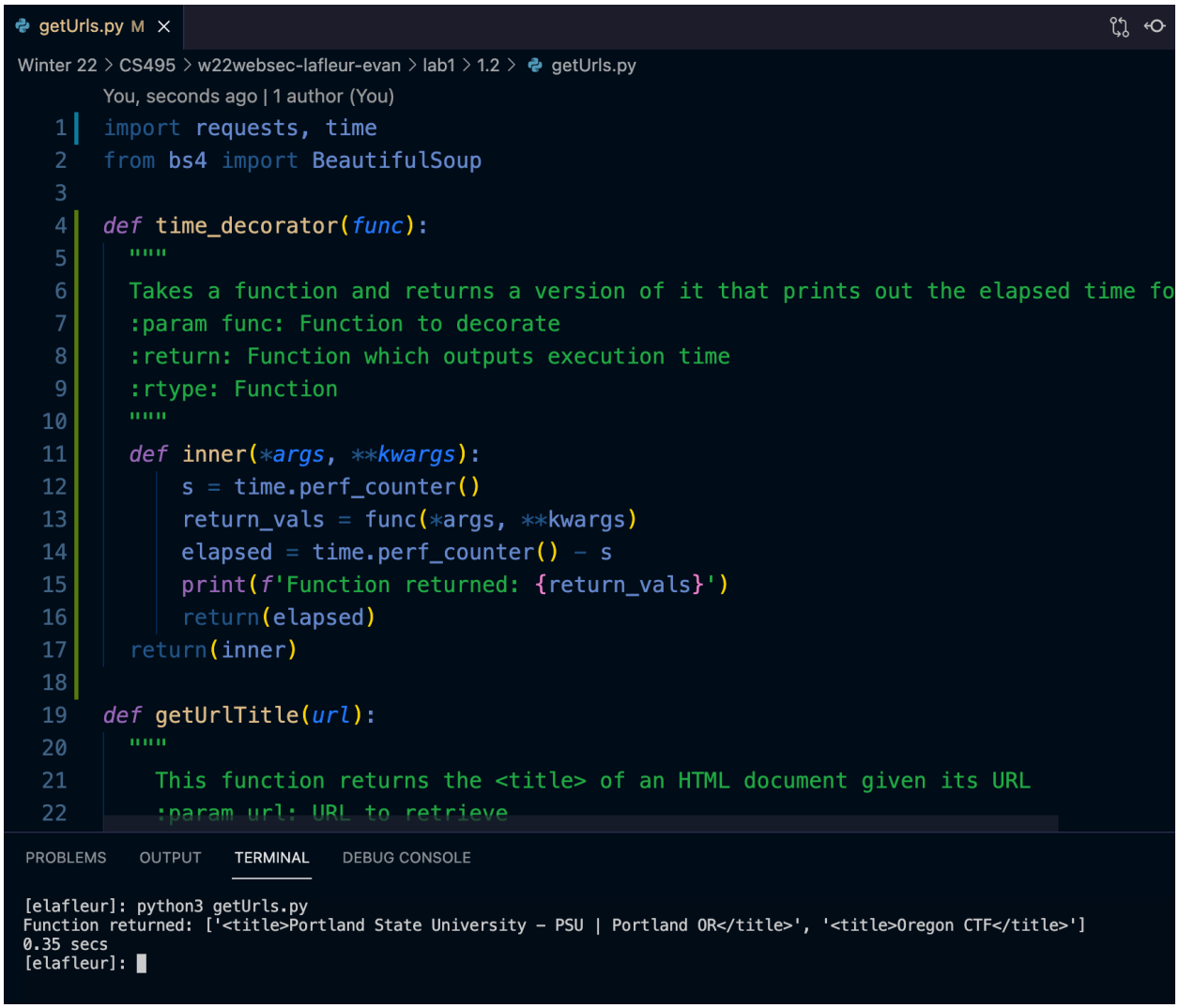
Winter 22 > CS495 > w22websec-lafleur-evan > lab1 > 1.2 > getUrls.py

```python
import requests
from bs4 import BeautifulSoup

def getUrlTitle(url):
    """
    This function returns the <title> of an HTML document given its URL
    :param url: URL to retrieve
    :type url: str
    :return: Title of URL
    :rtype: str
    """
    resp = requests.get(url)
    soup = BeautifulSoup(resp.text,'html.parser')
    title = str(soup.find('title'))
    return(title)

def getSequential(urls):
    """
    Given a list of URLs, retrieve the title for each one using a single synchronous process
    :param urls: List of URLs to retrieve
    :type urls: list of str
    :return: list of titles for each URL
    :rtype: list of str
```

PROBLEMS OUTPUT TERMINAL DEBUG CONSOLE

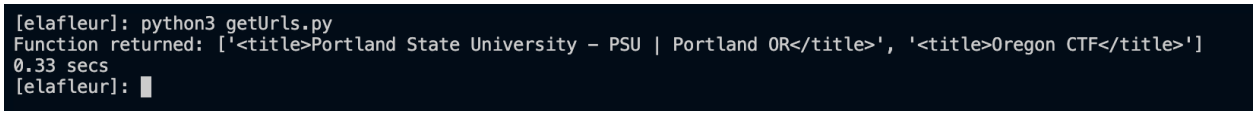
```
[elafleur]: python3 getUrls.py
['<title>Portland State University - PSU | Portland OR</title>', '<title>Oregon CTF</title>']
[elafleur]:
```

1.2.4 Timing Decorator



1.2.5 Run Requential Program

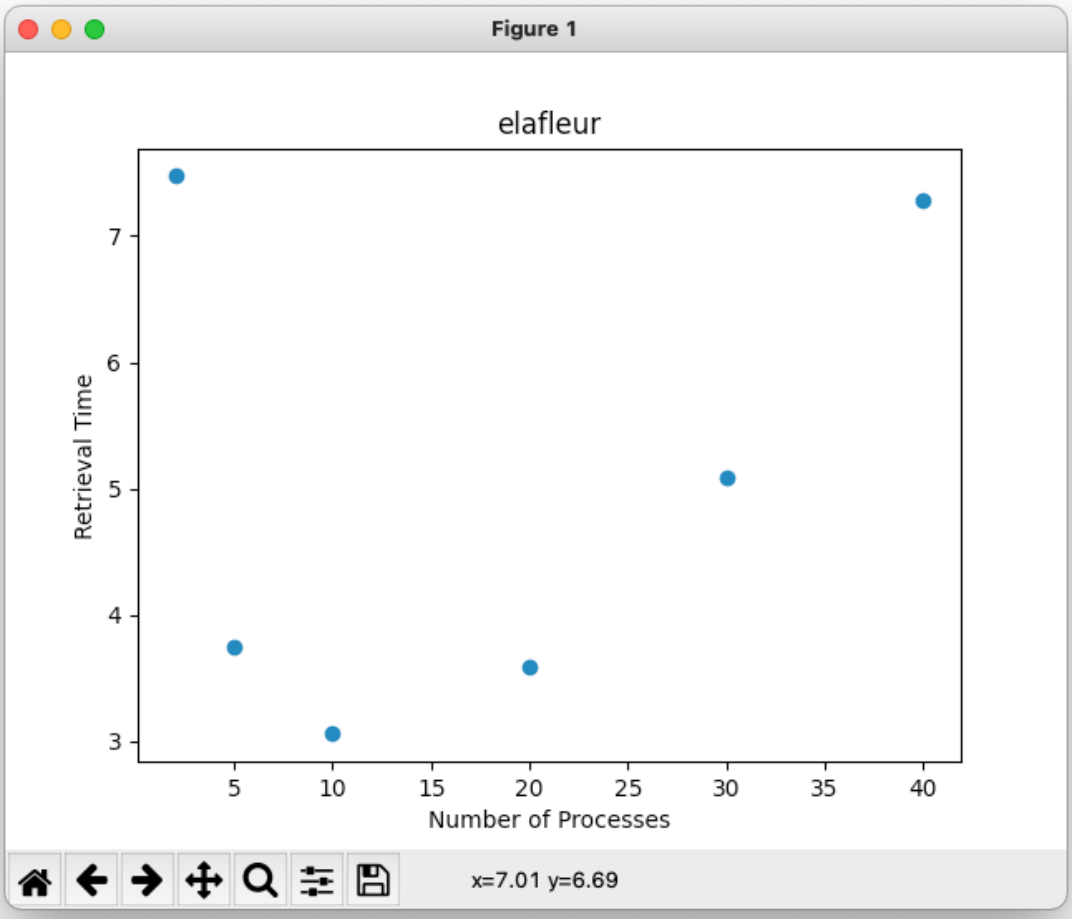
Attached is the final output for the ‘getUrls.py’ program.



1.2.6 Multiprocessing Program

```
PROBLEMS  OUTPUT  TERMINAL  DEBUG CONSOLE

[elafleur]: python3 getUrls_multiprocessing.py
Function returned: ['<title>Portland State University - PSU | Portland OR</title>', '<title>Oregon CTF</title>', '<title>YouTube</title>', '<title>GitHub: Where the world builds so
ftware · GitHub</title>', '<title data-react-helmet="true">Target : Expect More. Pay Less.</title>', '<title>LinkedIn: Log In or Sign Up</title>']
2 0.92
Function returned: ['<title>Portland State University - PSU | Portland OR</title>', '<title>Oregon CTF</title>', '<title>YouTube</title>', '<title>GitHub: Where the world builds so
ftware · GitHub</title>', '<title data-react-helmet="true">Target : Expect More. Pay Less.</title>', '<title>LinkedIn: Log In or Sign Up</title>']
5 0.88
Function returned: ['<title>Portland State University - PSU | Portland OR</title>', '<title>Oregon CTF</title>', '<title>YouTube</title>', '<title>GitHub: Where the world builds so
ftware · GitHub</title>', '<title data-react-helmet="true">Target : Expect More. Pay Less.</title>', '<title>LinkedIn: Log In or Sign Up</title>']
10 0.80
[elafleur]:
```

1.2.8 Matplotlib

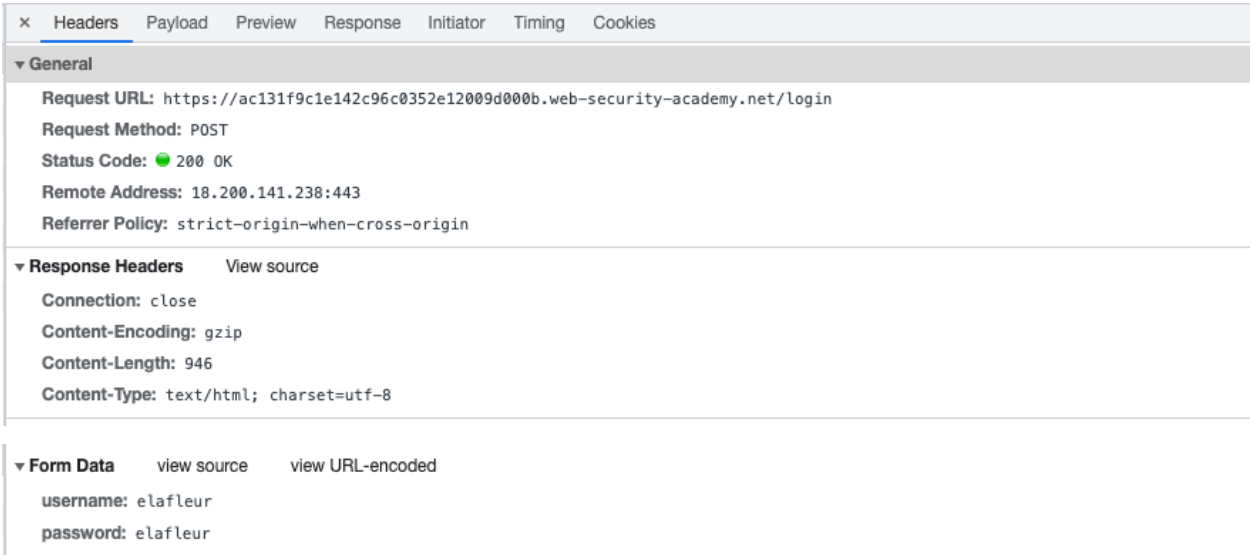


1.12 Asynchronous Program

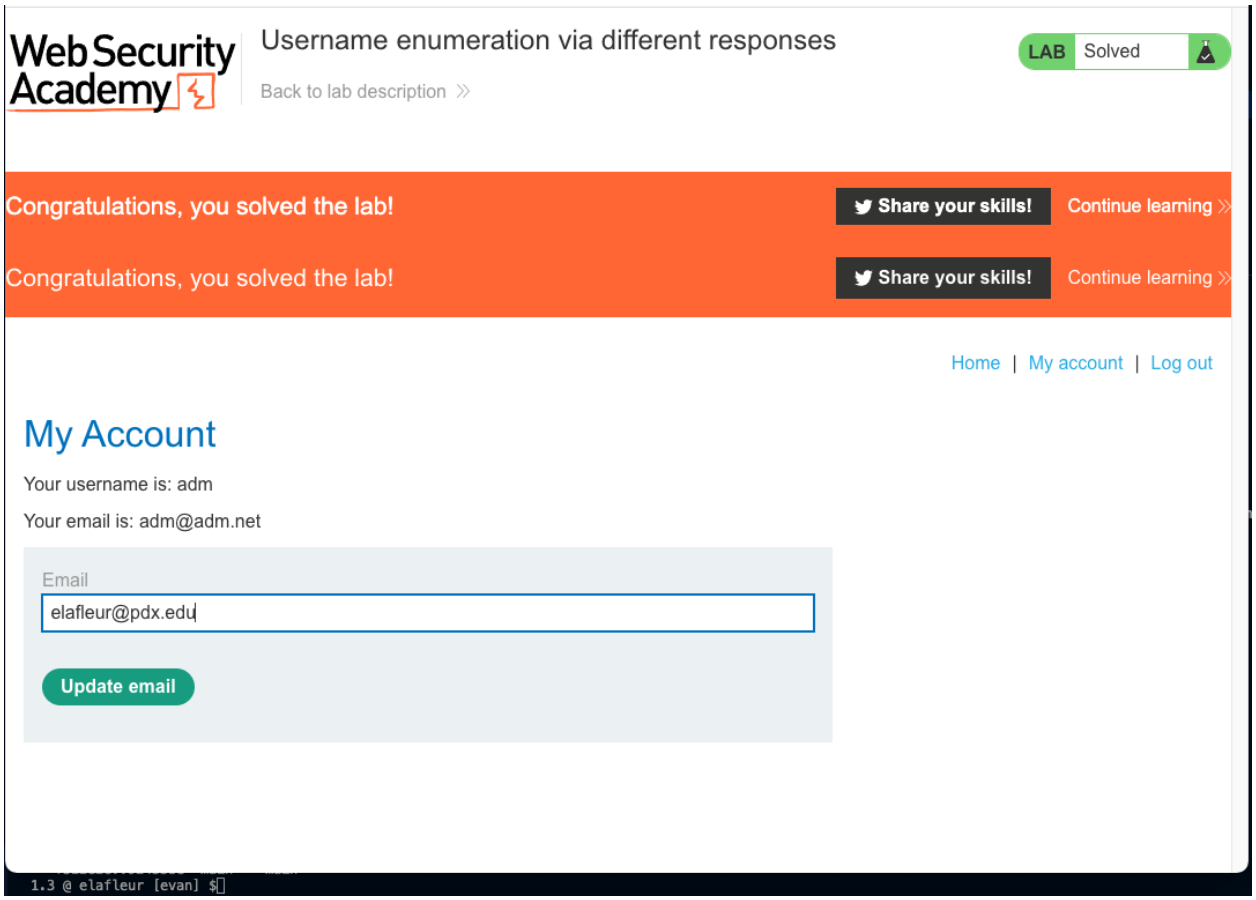
```
Function returned: ['<title>YouTube</title>', '<title>YouTube</title>', '<title>Google Scholar</title>',
'<title id="main-title">Google Play</title>', '<title>  Google Maps  </title>', '<title>Sign in - Google
e for all your files</title>', '<title>Sign in - Google Accounts</title>', '<title>Google Sites</title>',
>', '<title>Gmail</title>', '<title>Google</title>', '<title>Google</title>', '<title>Google</title>', '<
ied Advertising and Analytics</title>', '<title>Google</title>', '<title>Google</title>', '<title>Google
, '<title>Google</title>', '<title>\n      Google Account\n    </title>', '<title>Google Chrome - Downloa
e>Google</title>', '<title>Google Photos</title>', '<title>Google</title>', '<title>Google</title>', '<ti
Async version: 2.12
1.2 @ elafleur [evan] $
```

1.3 Broken Authentication

It appears that my OS and version of chrome stores the payload under the "payload" tab instead of the "headers" tab. I attached both screenshots below for reference.

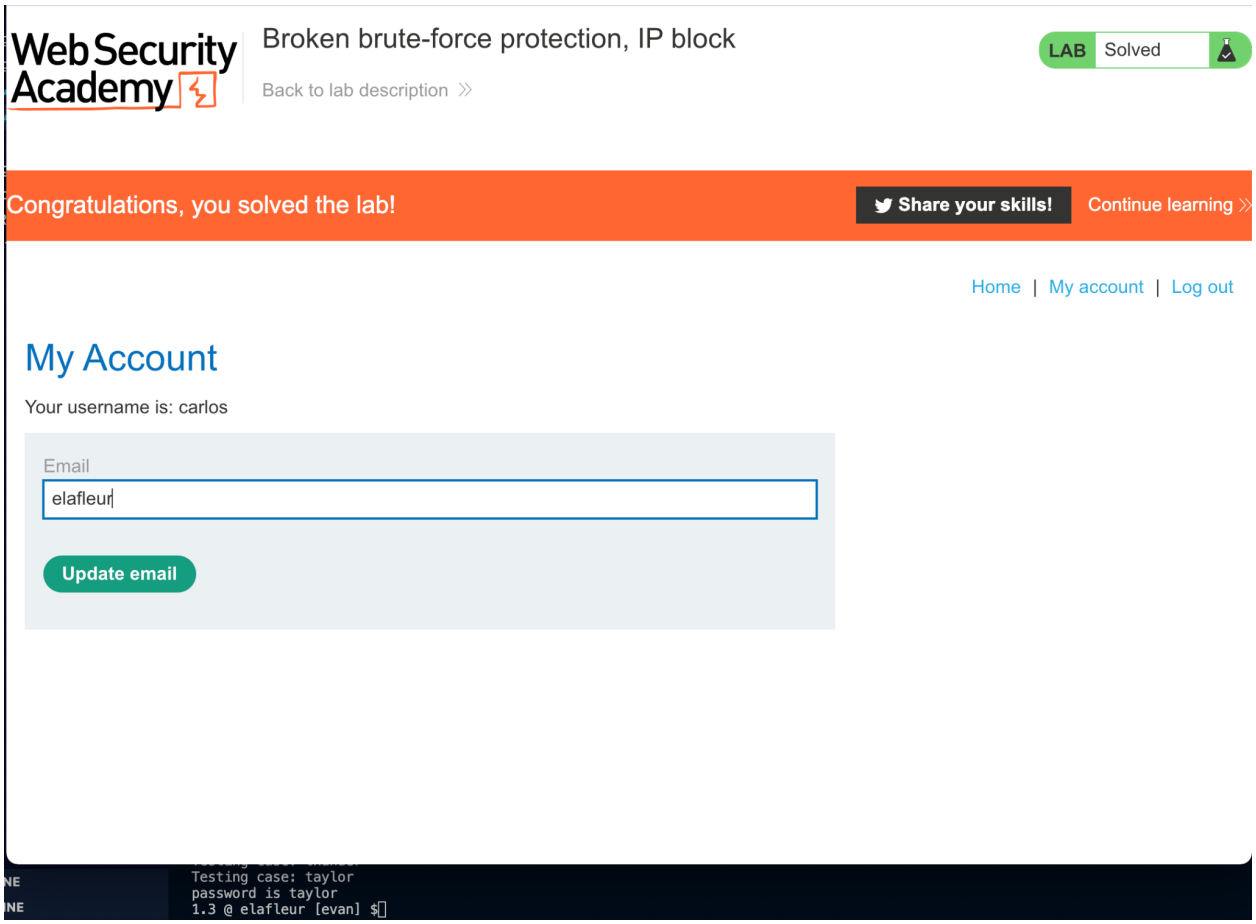


1.3.3 username-enumeration-via-different-responses



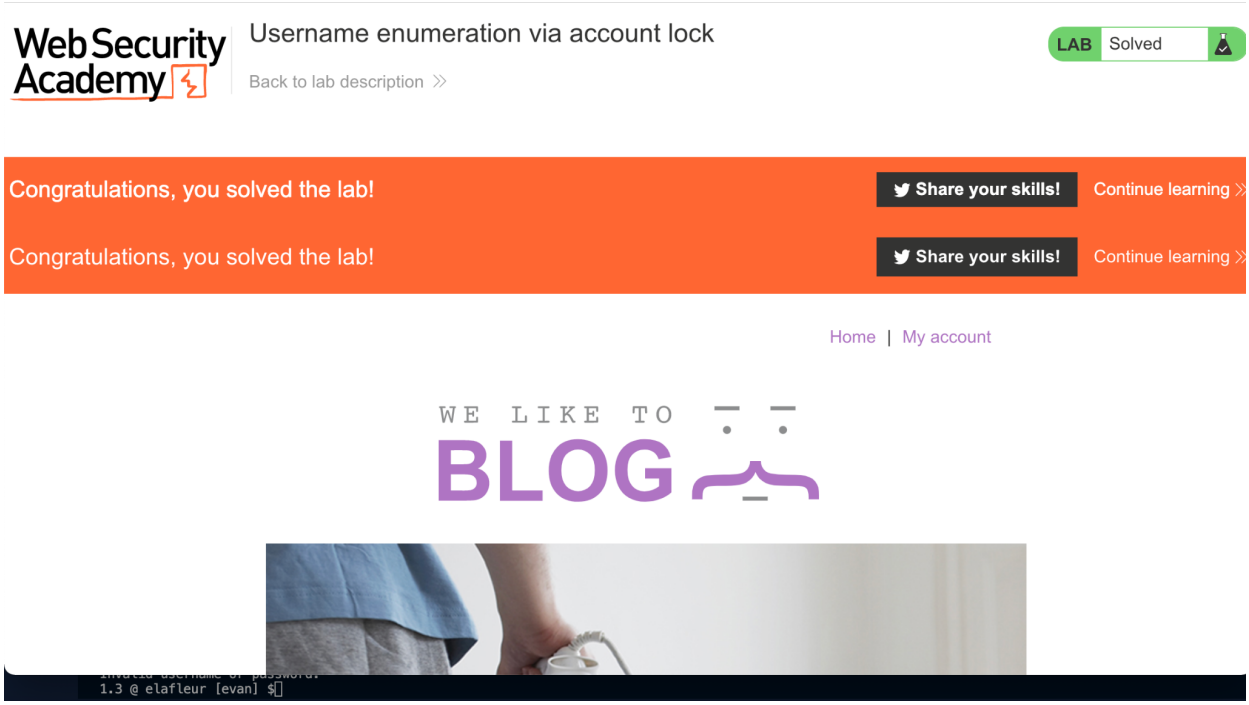
Solved the portswigger challenge:
[username-enumeration-via-different-responses]

1.3.4 broken-bruteforce-protection-ip-block



1.3.4 username-enumeration-via-account-lock

For this lab, I had some trouble brute forcing the password, as I had made it through the password list multiple times and it was unsuccessful in finding the password.
After letting the website reset itself and regenerate, the program finally cracked the password and username.



1.3.5 oauth-authentication-bypass-via-oauth-implicit-flow

- What is the DNS name of the identity provider?
  - Oauth?
- What is the client_id that is sent to the identity provider as a URL parameter?
  - xos5jx4nouix8mw326emr
- What is the value of the redirect_uri (e.g. the client application's callback URL) that the identity provider will send the user back to after authentication and consent is performed?
  - https://ac561fe01ef34688c0ef455e003200e7.web-security-academy.net/oauth-callback&response_type=token&nonce=62046532&scope=openid%20profile%20email
- What scopes are being requested by the client application for the user to authorize?
  - openid profile email
- What kind of response_type is being requested from the identity provider?
  - token
- What is the Location the user is sent to that implements the authentication login form on the identity provider's site?
  - https://ac561fe01ef34688c0ef455e003200e7.web-security-academy.net/oauth-callback

- What is URL is the form data sent to when the user logs in as specified in the action attribute of the form?
    -
- What is the URL the form is sent to?
    - https://oauth-ac551f841ecb46b1c0a445f702c60052.web-security-academy.net/interaction/_Hq1NL2CPSL5btF2_zgq8
- What is the access token the user will relay to the client application via its OAuth callback URL?

- What is the e-mail address associated with the wiener account?
    - wiener@hotdog.com
- What is the function of the first two const lines?
    - The URLSearchParams API being called is typically used to gain access o the pieces of a URL. It will allow better manipulation of th equery string.
- What content is being retrieved from the identity provider in the first fetch?

```
headers: {
        'Authorization': 'Bearer ' + token,
        'Content-Type': 'application/json'
      }
```

- What 3 values are being sent to the client application in the second fetch?

```
body: JSON.stringify({
            email: j.email,
            username: j.sub,
            token: token
        })
```

- What location is the user redirected to at the end of the implicit flow?

https://ac1d1f581f5d755cc07b68e3003d003b.web-security-academy.net/oauth-callback#access_token=3-wSMCDnQwkOxXXOaybunhe0xhpG2UKJlHey8CDG607&expires_in=3600&token_type=Bearer&scope=openid%20profile%20email

1.4 HW 1 (2fa-bypass-using-a-brute-force-attack)

```
hw1 @ elafleur [evan] $python3 hw1.py
ac561fa41f594758c019a2e800ef0062.web-security-academy.net
Logging in as carlos:montoya
Login response:
<!DOCTYPE html>
...
   <form class=login-form method=POST>
      <input required type="hidden" name="csrf" value="7baAzhSvQtYgInqu0HYNGpPBHkc2ZaLx">
      <label>Please enter your 4-digit security code</label>
      <input required type=text name=mfa-code>
      <button class=button type=submit> Login </button>
    </form>
...
</html>
```

Objective:
The object for this assignment was to develop a brute force program that will be able to determine the value of a 4 digit authentication code. This program is going to be written in python and will interact with the portswigger lab listed below.

[2fa-bypass-using-a-brute-force-attack]

Process:

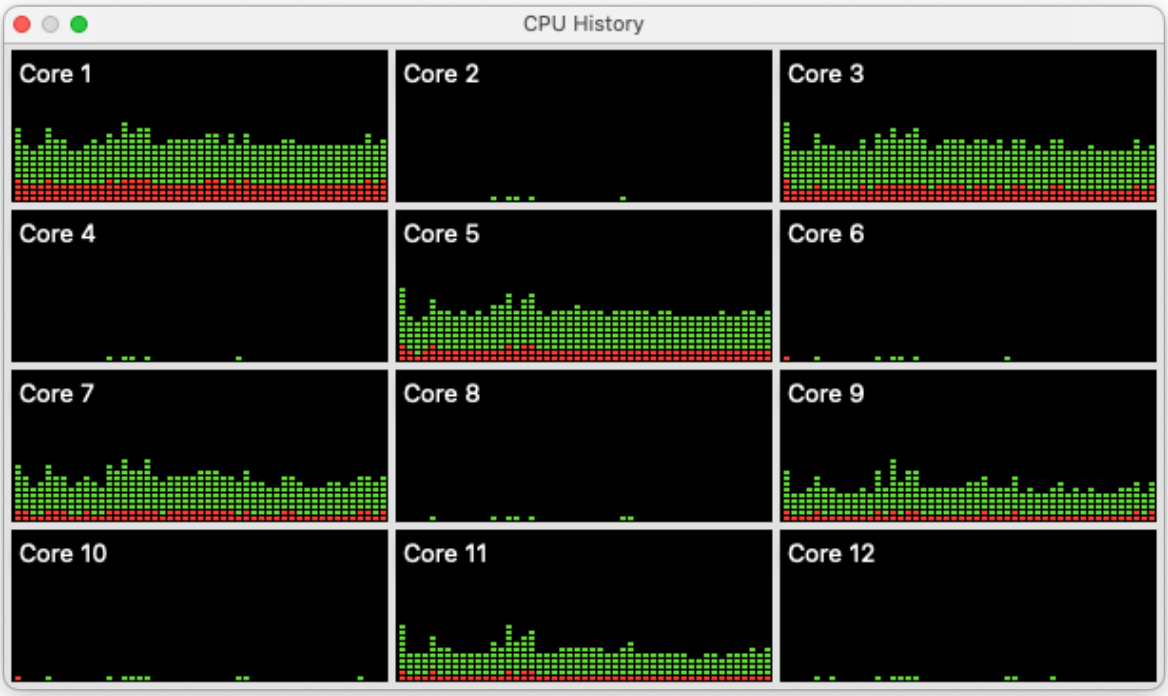
When this assignment was started, I figured it would be a simple thing to complete. I was able to successfully develop a program that would make its way through the list of digits (0000 - 9999). However this was done by just checking them one at a time. Since each query took approximately 1-2 seconds to complete, doing it this way was going to take some time. To speed things up, I first attempted to apply the multiprocessor and the threading library but had struggled for about 8 hours just attempting to get the function to properly work with multiple threads. I ended up rewriting this code multiple times over the course of three days just trying to get something to work.

Finally I was able to develop a program that would successfully determine the value for the authcode but the program did not end, it continued

attempting new codes which made the correct code get lost in the mix. To overcome this I found a webpage discussing Event Handling, and with this i was able to create an event that the program would only continue if it did not detect this event stopped.

After completing this program, i attempted to make it run more efficiently by increasing cores but decided it was efficient enough. The run time ranges anywhere from 58 seconds all the way up to 290 seconds.

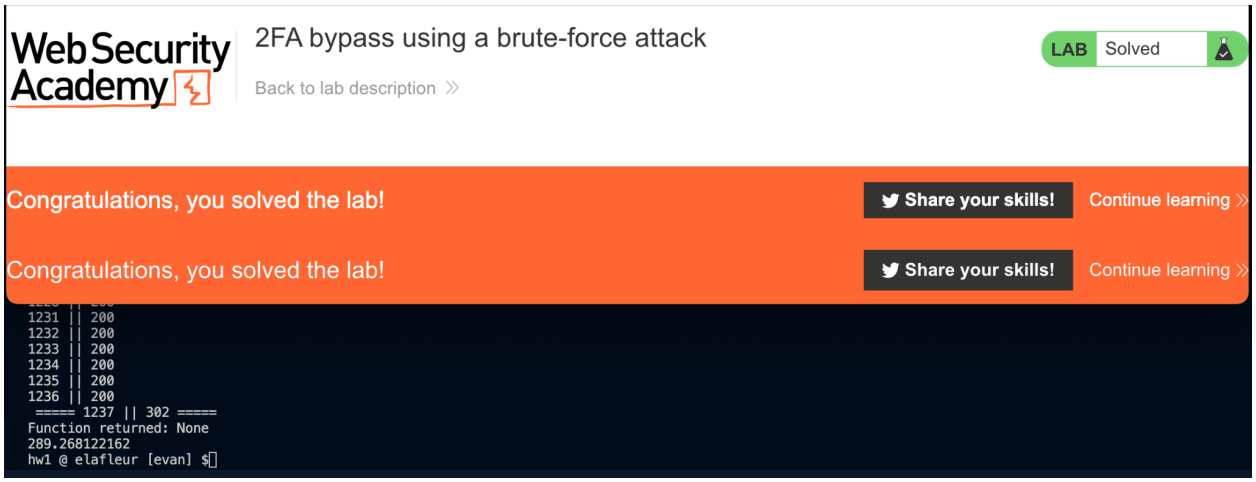
Improvement:
One thing that can be done to potentially improve efficiency is to implement a random number generator to generate the numbers instead of going from 0 to 9999.
Another thing I had attempted to do was implement a divide and conquer when some of the pool would start at 0 and the other part of the pool would start at 000. I think this would be the best way to determine this efficiently but it was taking a lot more time and knowledge than i had.



In this image above you can see that even when running the program at current spec, at least on my unit I was not maximizing the total power I could with this program. At the time of capture, the only thing running was the python program.

For the code of my assignment, I chose to have it display a simple layout where it showed the number being tested, and the response code given from the webpage. That way I was able to reference and make sure I had the correct code once it completed.

```
1228 || 200
1231 || 200
1232 || 200
1233 || 200
1234 || 200
1235 || 200
1236 || 200
 ===== 1237 || 302 =====
Function returned: None
289.268122162
hw1 @ elafleur [evan] $
```



1.5 Broken Access Control

1.5.1-1.5.2 file-path-traversal

simple

Completed the simple lab, but simply traversing to the root directory and requested the “/etc/passwd” file.

absolute-path-bypass

Completed this lab by taking the absolute path and injecting the "/etc/passwd" directly without traversing.

sequences-stripped-non-recursively

In this lab, since the web application automatically removes traversals, by simply adding a second traversal around the "../" it will still keep the additional traversal.

Web Security Academy

File path traversal, traversal sequences stripped non-recursively

LAB Solved

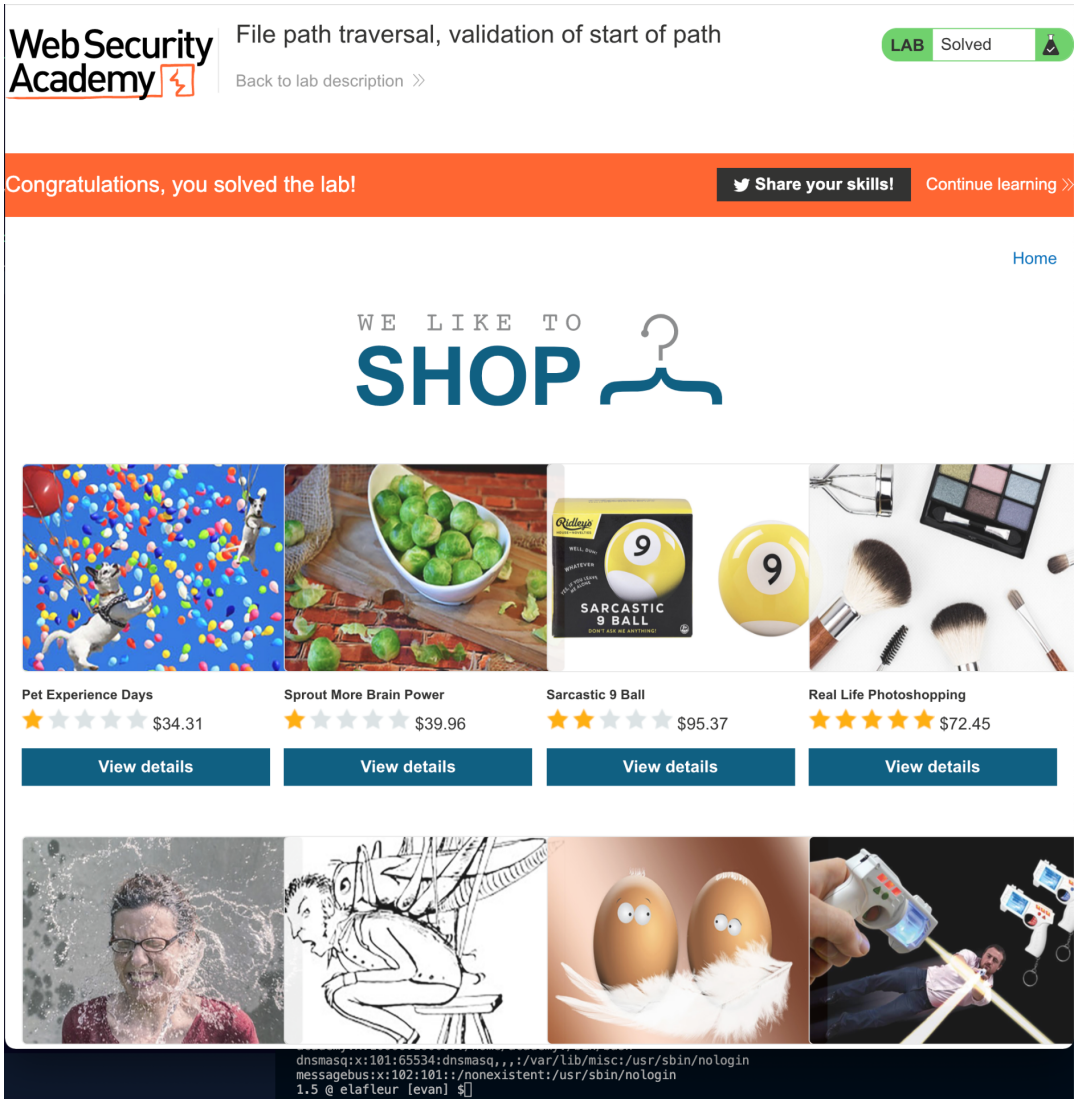
Back to lab description »

Congratulations, you solved the lab! Share your skills! Continue learning »

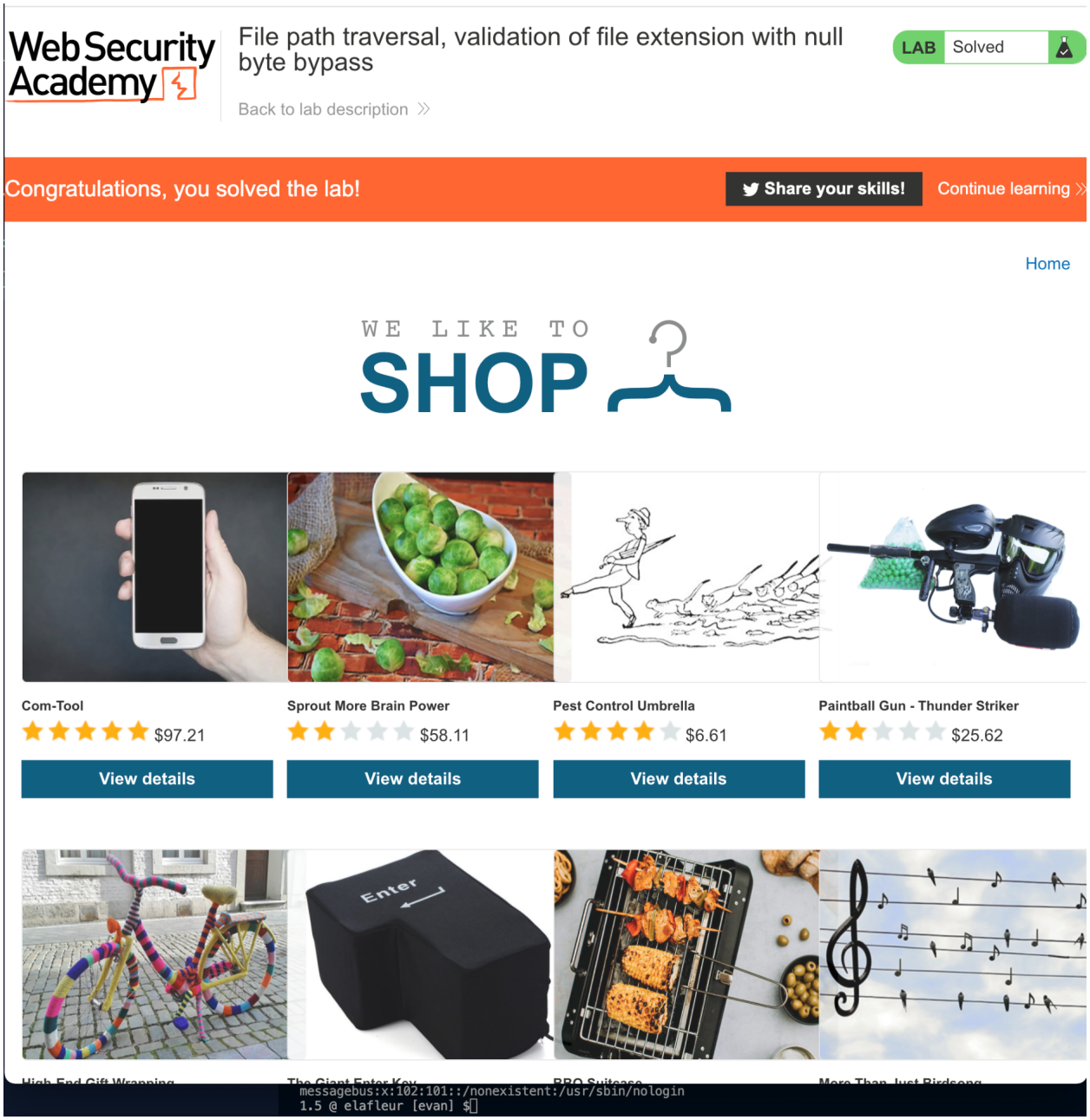
Home

Superfluous-url-decode

Validate-start-of-path

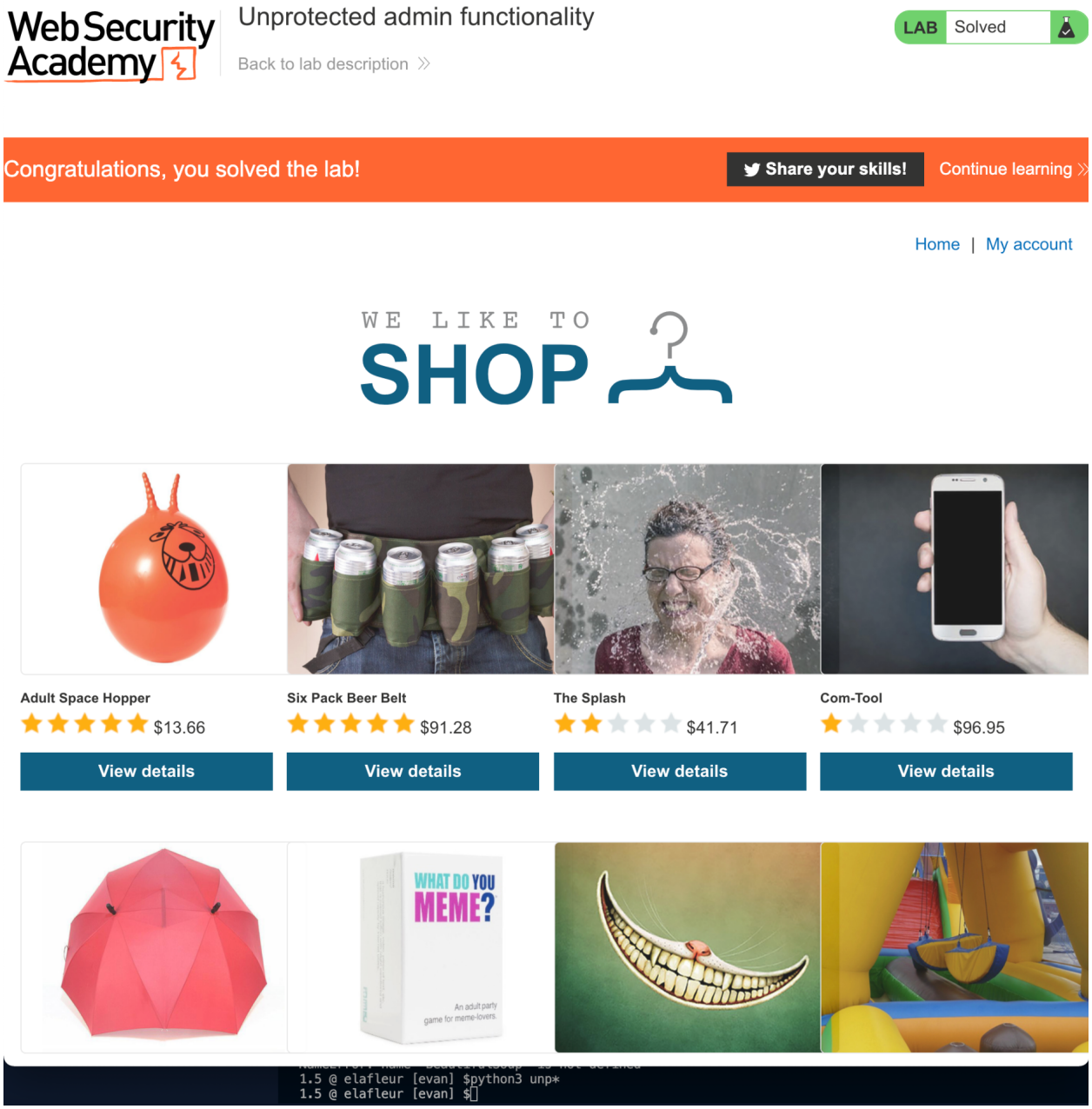


Validate-file-extension-null-byte-bypass

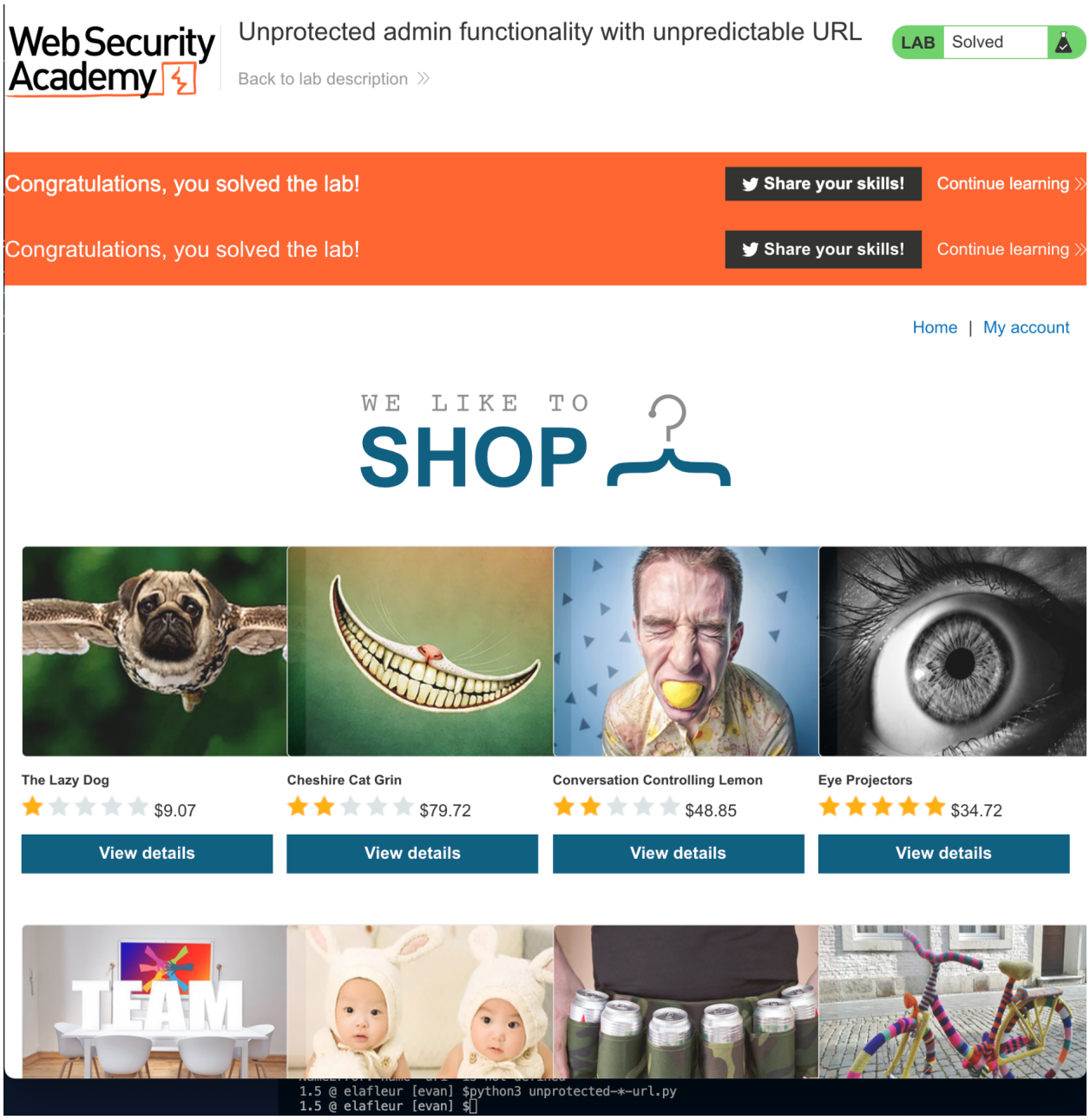


1.5.3-1.5.9 access-control

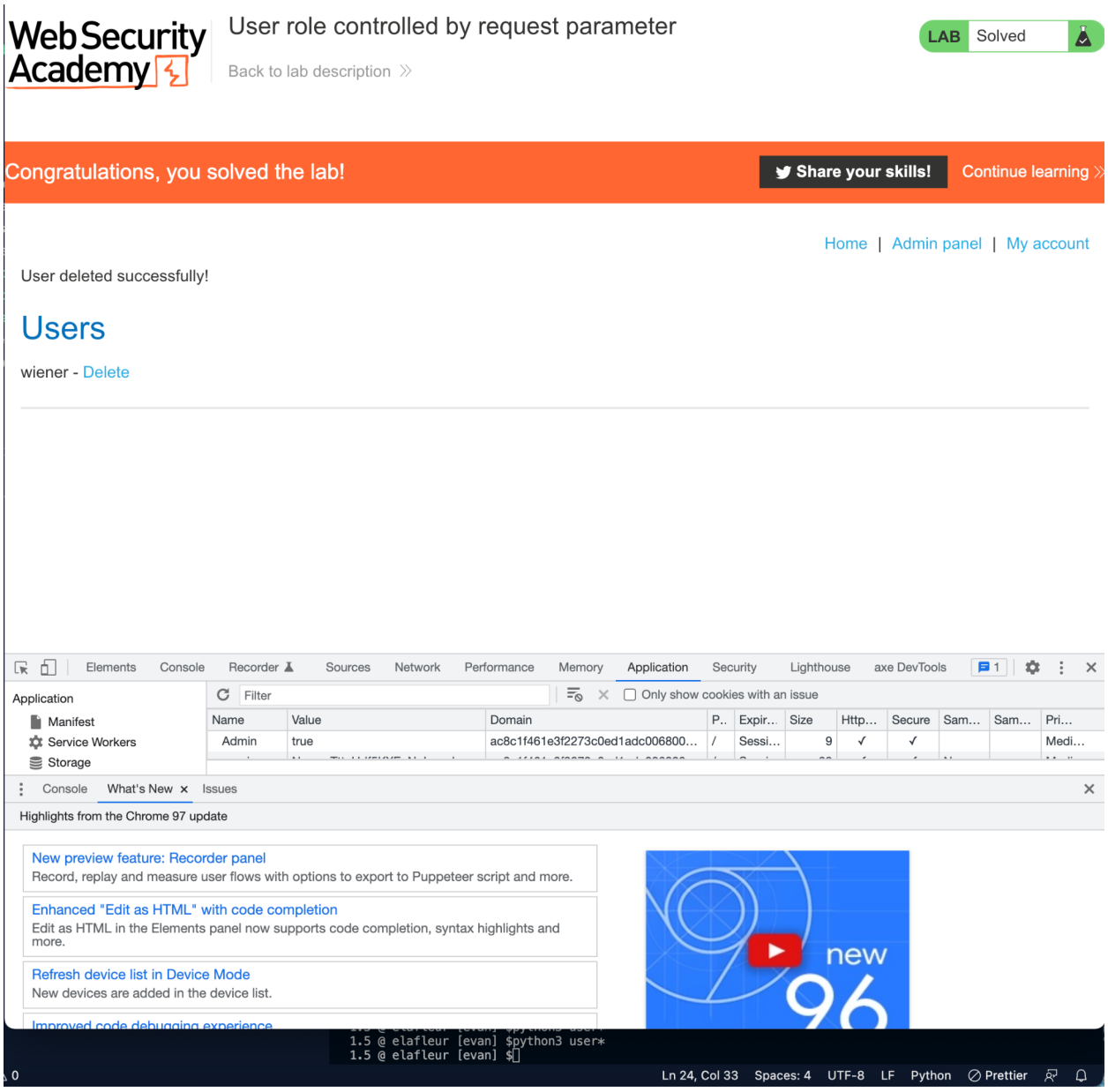
Unprotected-admin-functionality



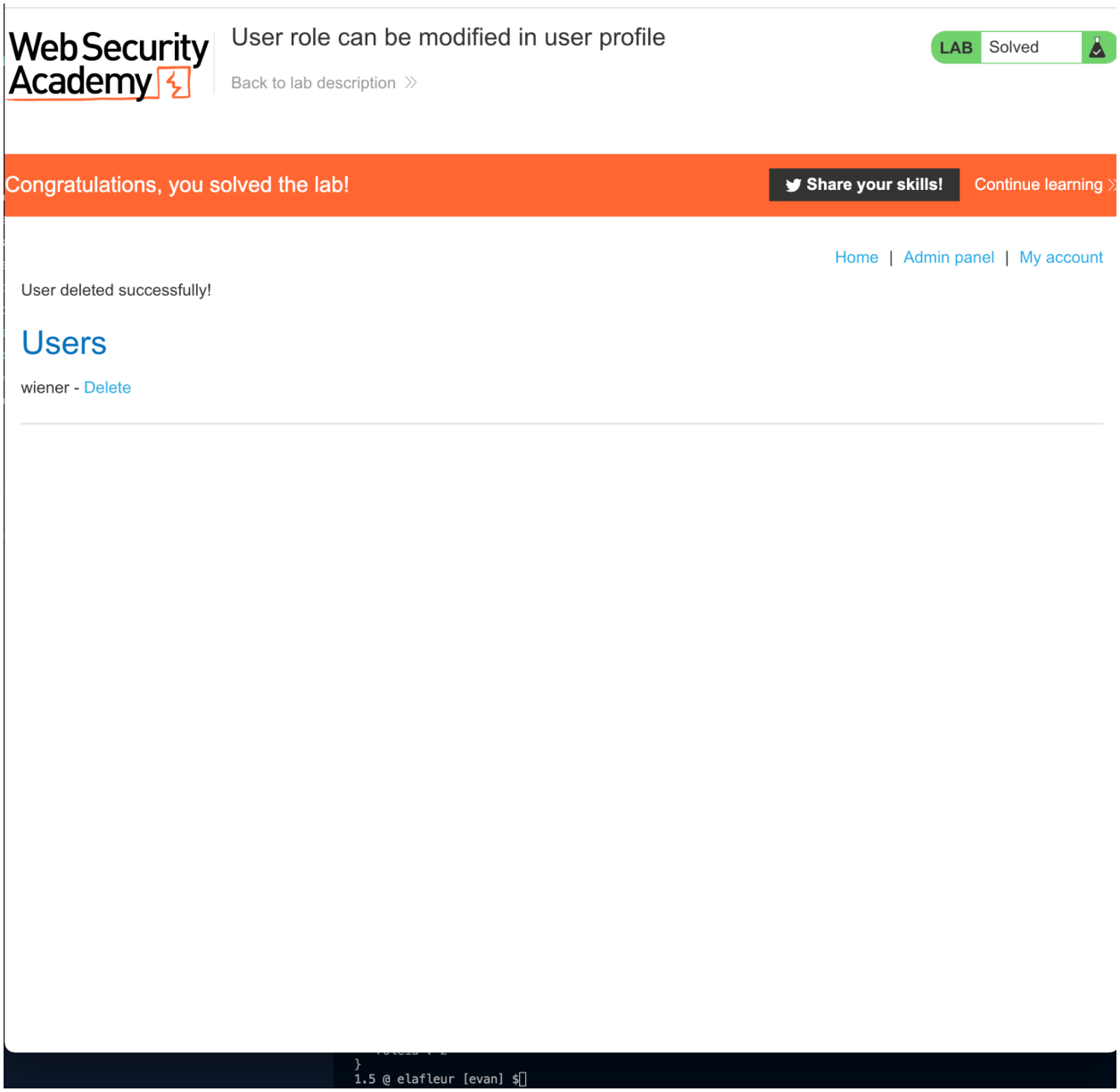
Unprotected-admin-functionality-with-unpredictable-url



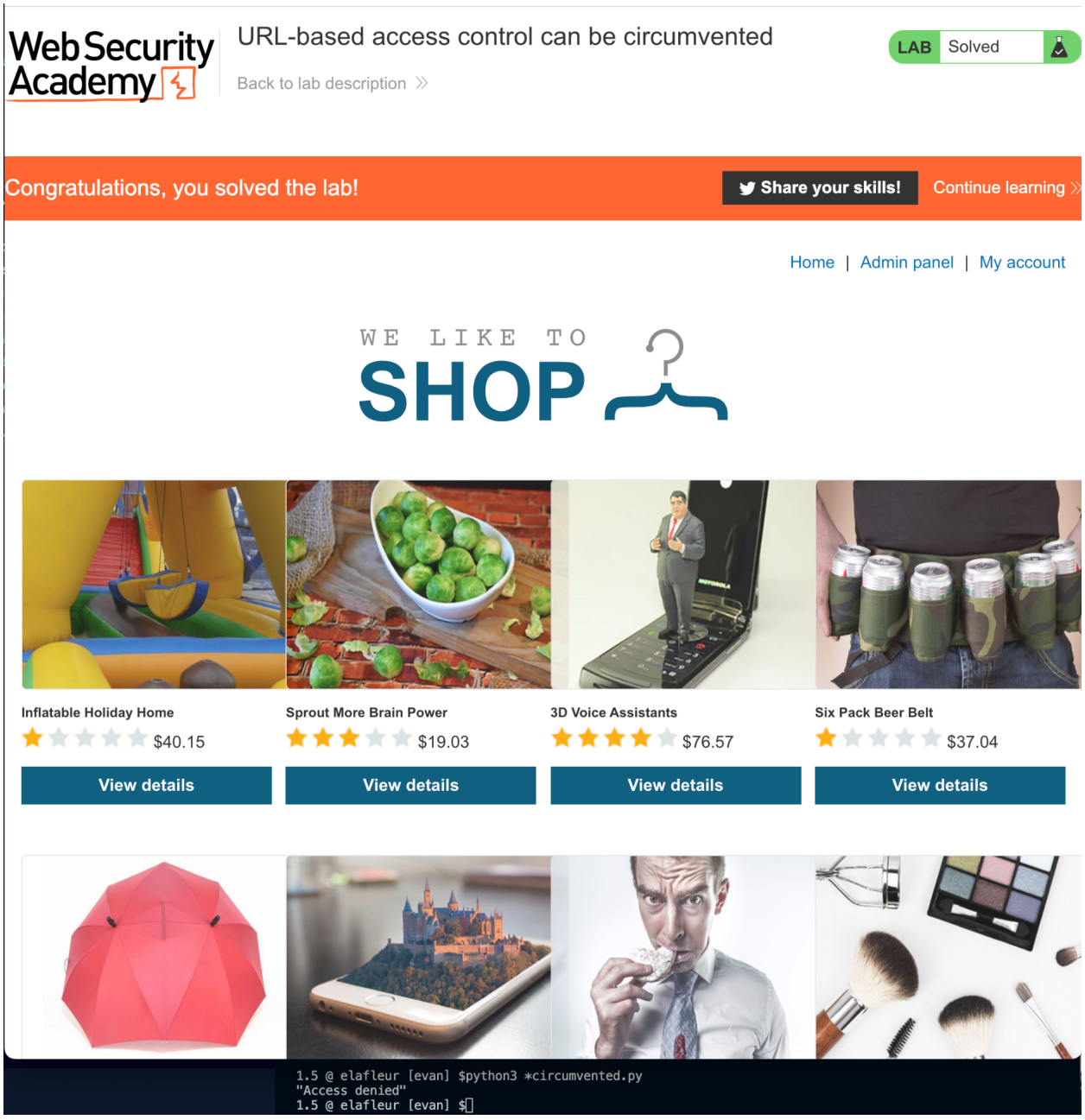
User-role-controlled-by-request-parameter



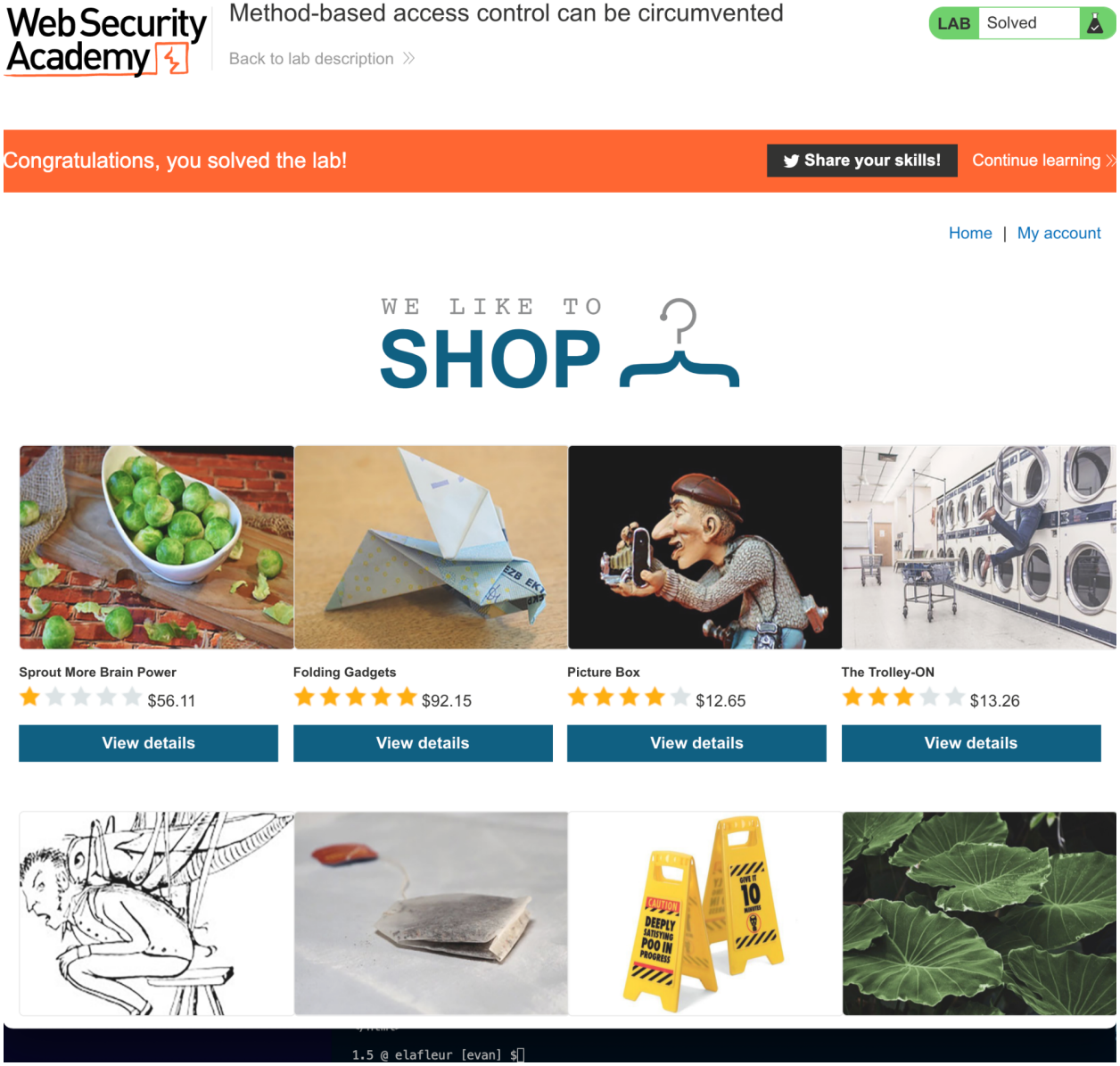
User-role-can-be-modified-in-user-profile


Web Security Academy
User role can be modified in user profile
Back to lab description
LAB Solved
Congratulations, you solved the lab!
Share your skills!
Continue learning
Home | Admin panel | My account
User deleted successfully!
Users
wiener - Delete
}
1.5 @ elafleur [evan] $

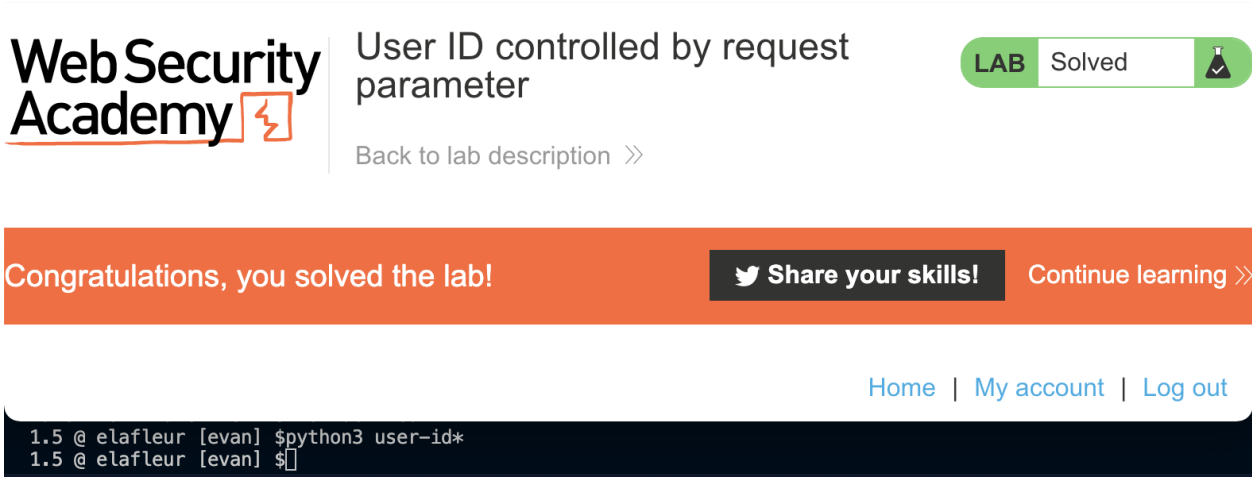
Url-based-access-control-can-be-circumvented



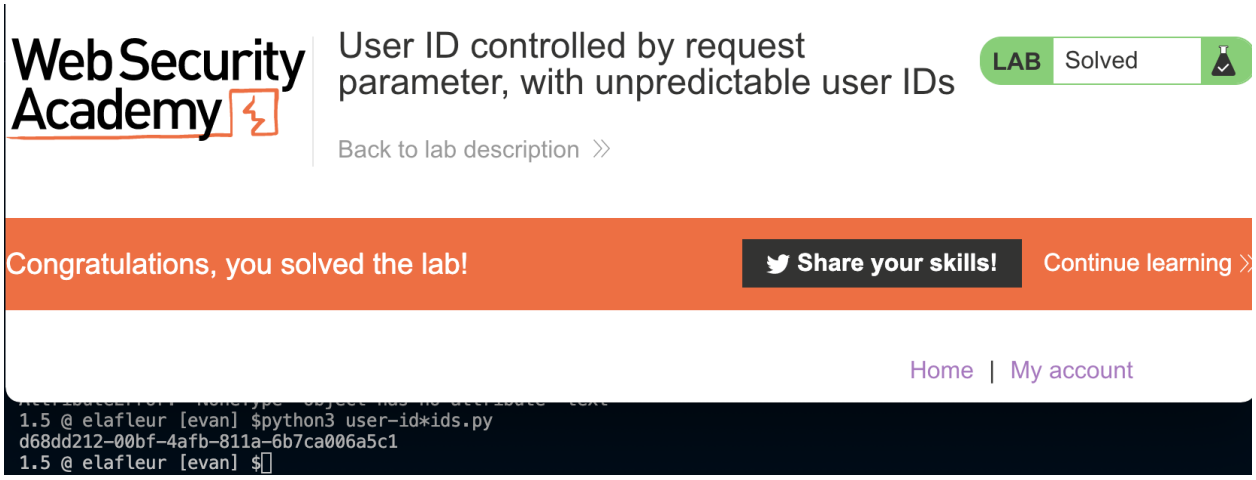
Method-based-access-control-can-be-circumvented



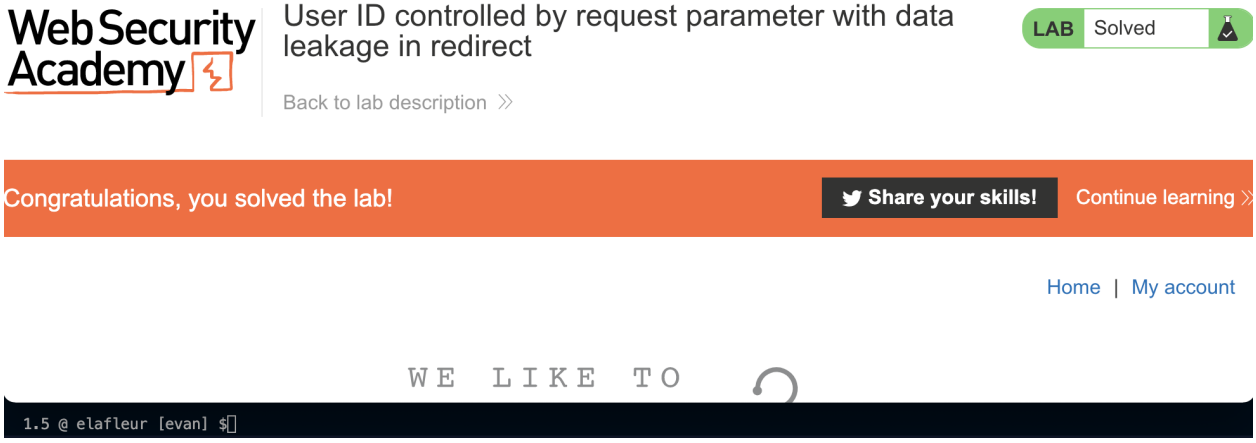
User-id-controlled-by-request-parameter


Web Security Academy
User ID controlled by request parameter
Back to lab description
LAB
Solved
Congratulations, you solved the lab!
Share your skills!
Continue learning
Home | My account | Log out
1.5 @ elafleur [evan] $python3 user-id*
1.5 @ elafleur [evan] $

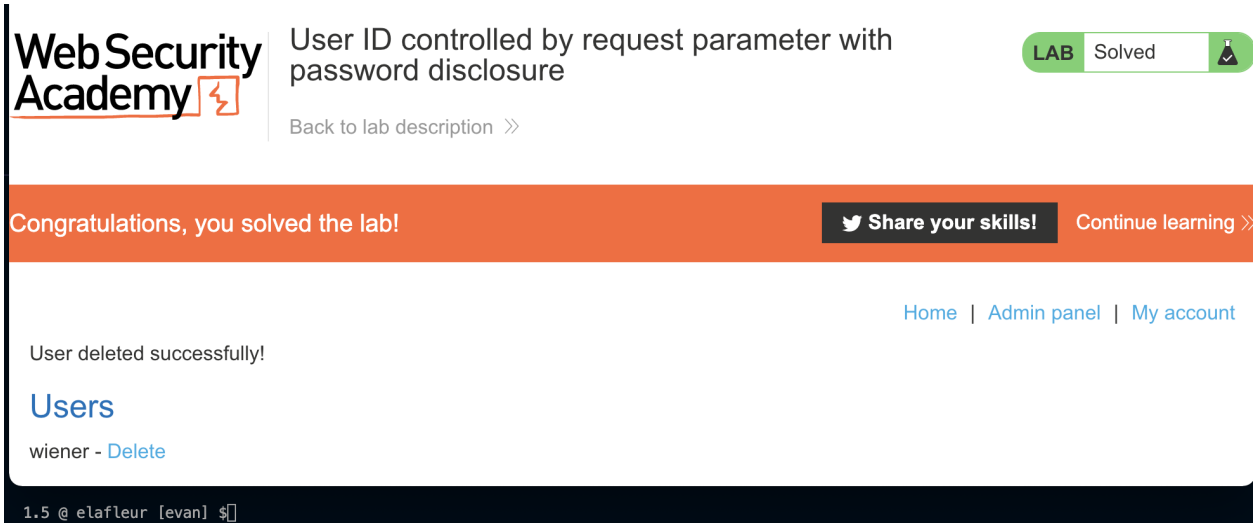
User-id-controlled-by-request-parameter-with-unpredictable-user-ids



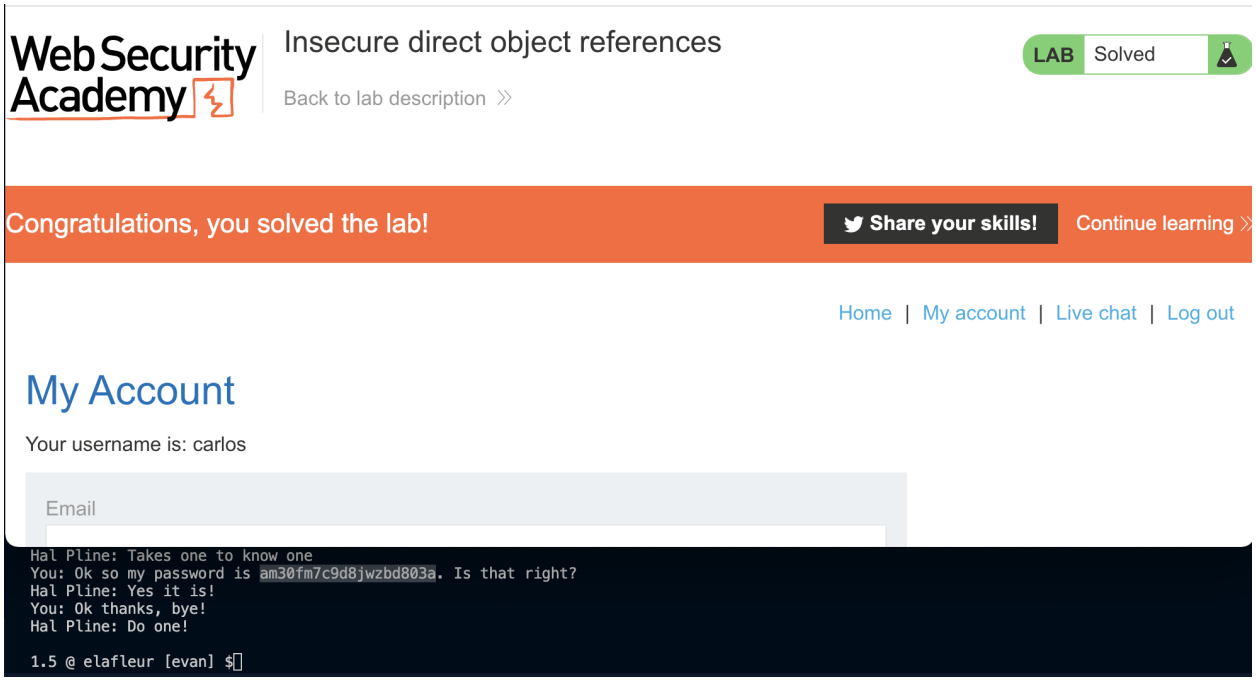
User-id-controlled-by-request-with-data-leakage-in-redirect



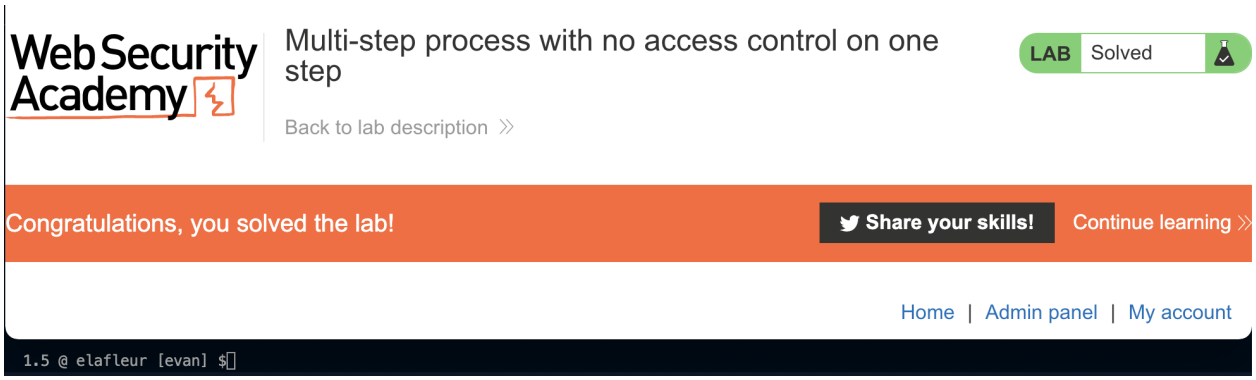
User-id-controlled-by-request-parameter-with-password-disclosure



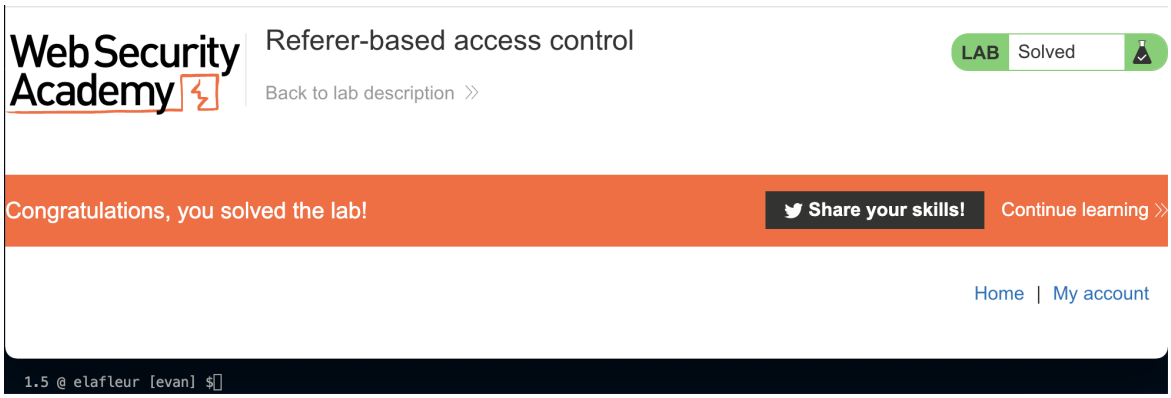
Insecure-direct-object-references



Multi-step-process-with-no-access-control-on-one-step

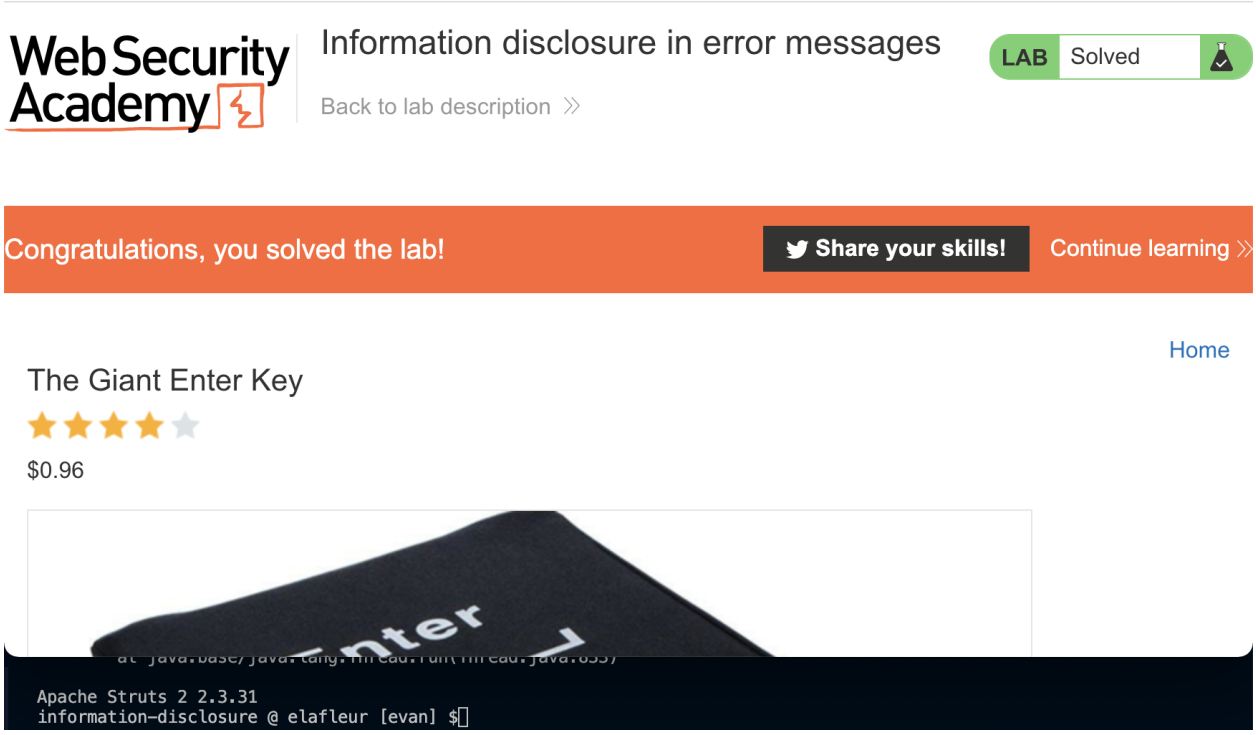


Referrer-based-access-control

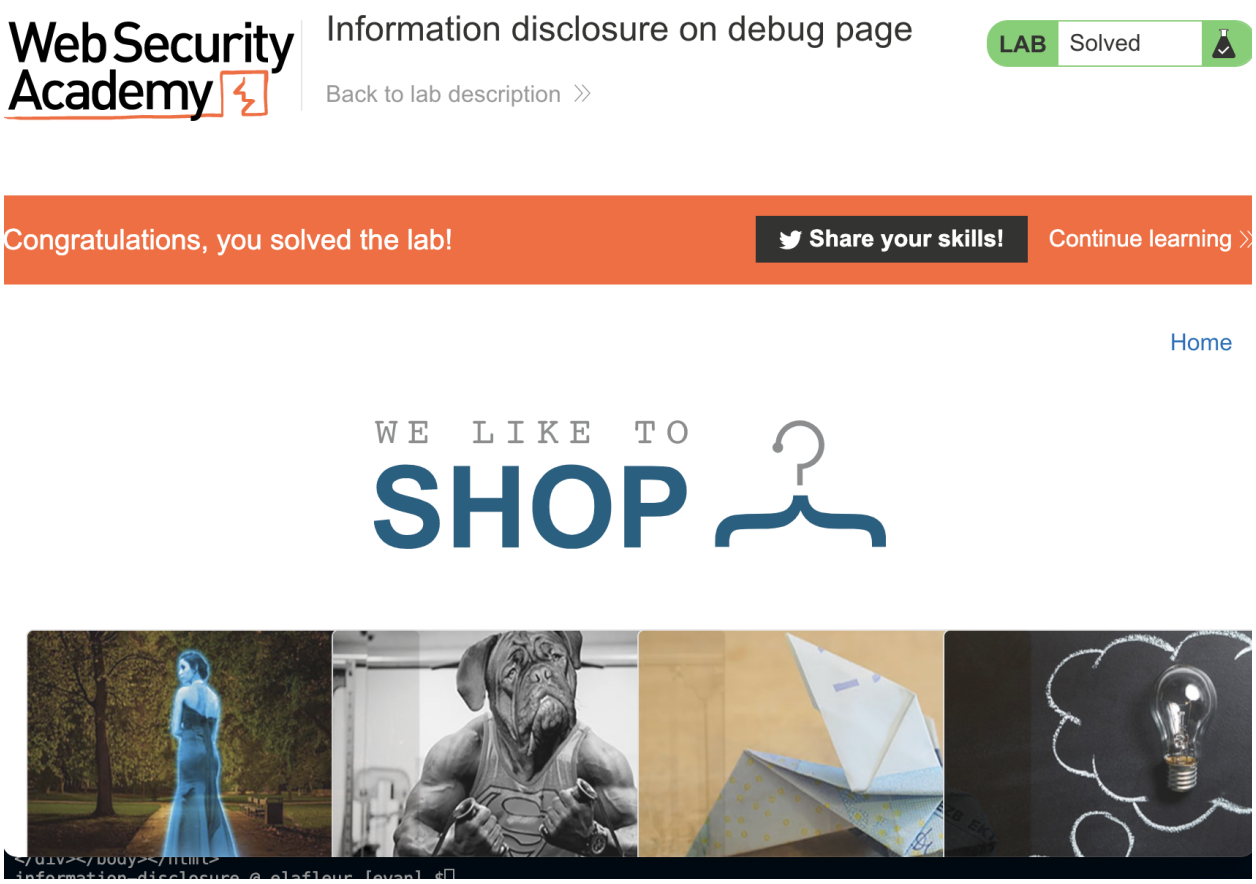


1.5.10 information-disclosure

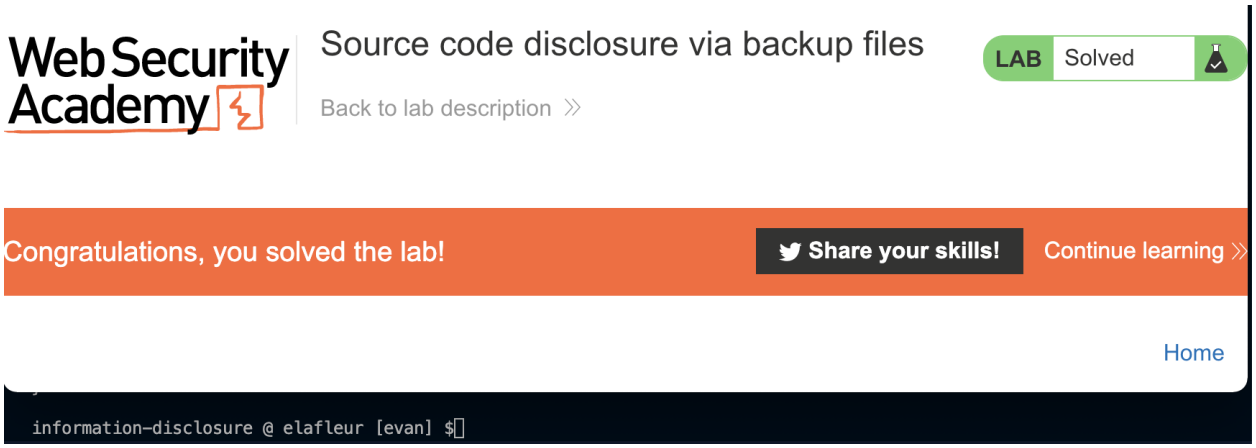
Lab-infoleak-in-error-messages



Lab-infoleak-on-debug-page

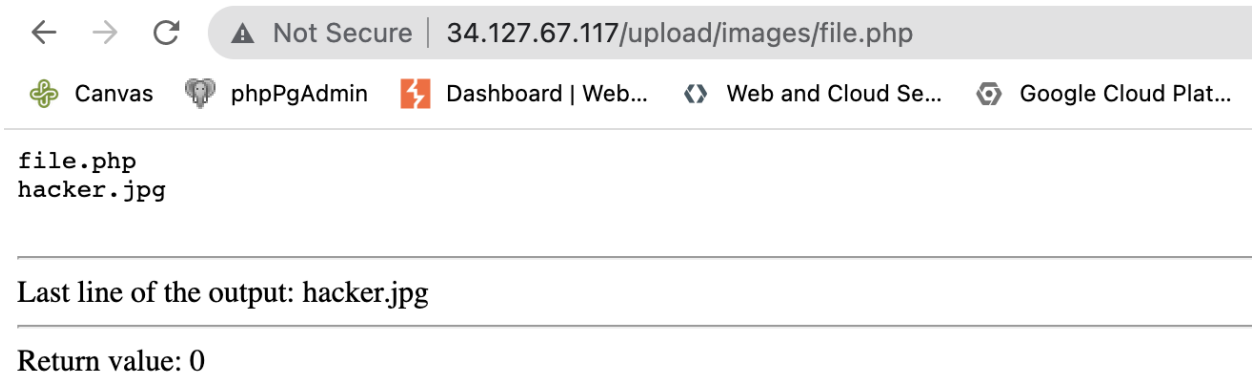


Lab-infoleak-via-backup-files



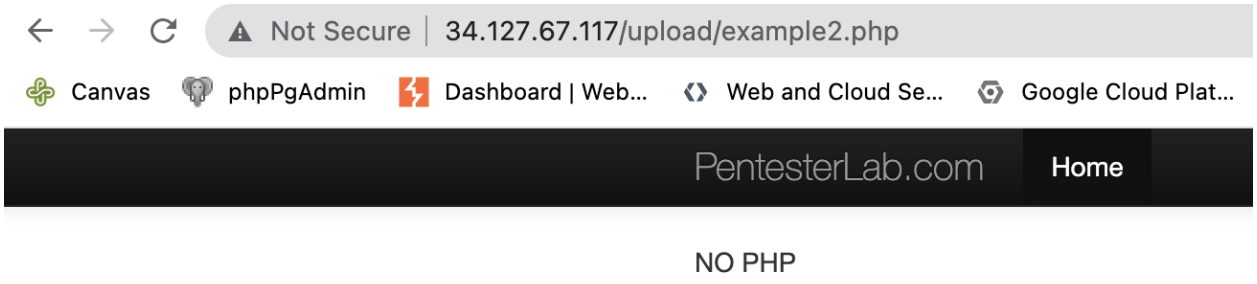
1.5.11 WFP1: File Upload

Example #1

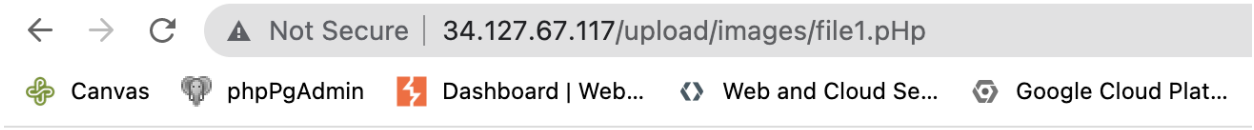


Example #2

Uploading the same file used in the previous exercise:



By changning the filename to ‘file.pHp’ it will bypass the programs check since the program check is case sensitive. This will then run the program.




```
file.htaccess
file.inc
file.php
file.shtml
file1.pHp
hacker.jpg
```

Last line of the output: hacker.jpg

Return value: 0

1.5.12-1.5.13 file-upload

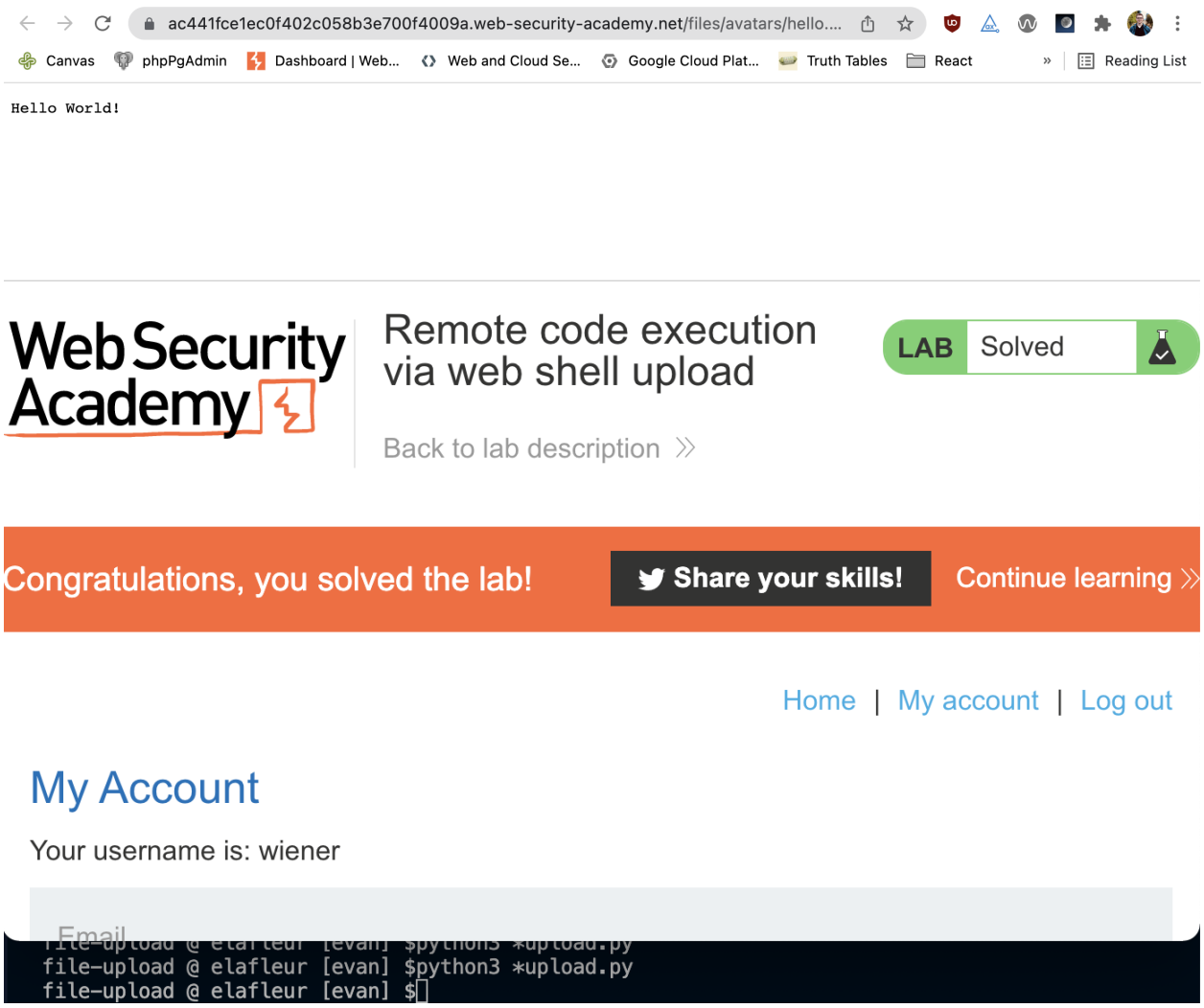
Lab-file-upload-remote-code-execution-via-web-shell-upload

- What are the names of the form fields that are of type hidden?

```
<input required type="hidden" name="csrf" value="Y6g0Pqet1J4LBQvDNrlP67m97Z2NK1Be">
```

```
<input type="hidden" name="user" value="wiener">
```

- Take a screenshot of the message and the URL the avatar is stored at on the server



lab-file-upload-web-shell-upload-via-content-type-restriction-bypass

Sorry, file type text/plain is not allowed Only image/jpeg and image/png are allowed Sorry, there was an error uploading your file.

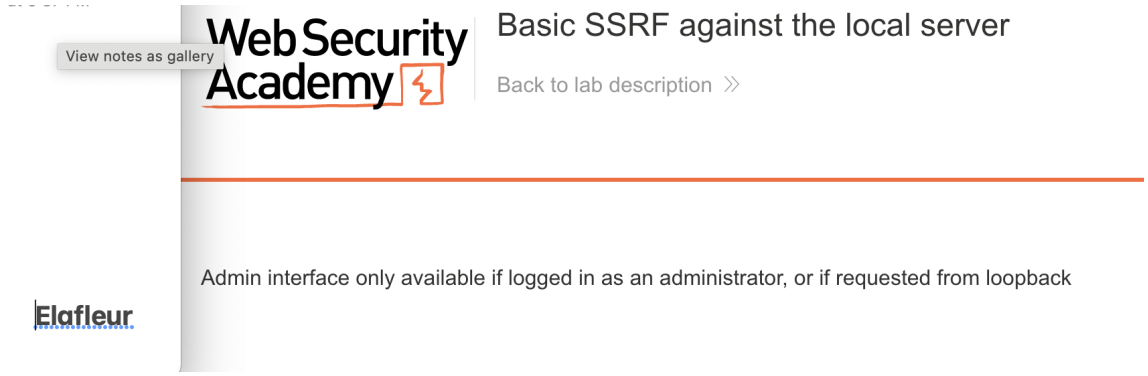
« Back to My Account

Return to previous page

1.6 SSRF

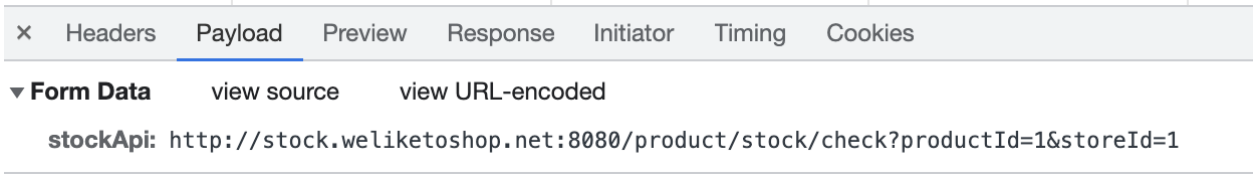
1.6.1-1.6.4 ssrf

basic-ssrf-against-localhost



▼ **Request Headers** View source

Accept: */*
Accept-Encoding: gzip, deflate, br
Accept-Language: en-US,en;q=0.9
Connection: keep-alive
Content-Length: 107
Content-Type: application/x-www-form-urlencoded
Cookie: session=IE5gpC5pAocVBFkY2NnVL8Jw3Xt4gfT8
Host: ac591fa61e4cfe48c0067c5e00a00075.web-security-academy.net
Origin: https://ac591fa61e4cfe48c0067c5e00a00075.web-security-academy.net
Referer: https://ac591fa61e4cfe48c0067c5e00a00075.web-security-academy.net/product?productId=1
sec-ch-ua: " Not;A Brand";v="99", "Google Chrome";v="97", "Chromium";v="97"
sec-ch-ua-mobile: ?0
sec-ch-ua-platform: "macOS"
Sec-Fetch-Dest: empty
Sec-Fetch-Mode: cors
Sec-Fetch-Site: same-origin
User-Agent: Mozilla/5.0 (Macintosh; Intel Mac OS X 10_15_7) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/97.0.4692.71 Safari/537.36



× Headers Payload Preview Response Initiator Timing Cookies

▼ **Form Data** view source view URL-encoded

stockApi: http://stock.weliketoshop.net:8080/product/stock/check?productId=1&storeId=1

Web Security Academy

Basic SSRF against the local server

Back to lab description »

LAB Solved

Congratulations, you solved the lab!

Share your skills! Continue learning »

Home | My account

Gym Suit

$27.15

1.6 @ elafleur [evan] $

basic-ssrf-against-backend-system

Web and Cloud Security
https://codelabs.cs.pdx.edu/cs495/

Web Security Academy

Basic SSRF against another back-end system

Back to lab description »

LAB Solved

Congratulations, you solved the lab! Share your skills! Continue learning »

Home | My account

```
1.6 @ elafleur [evan] $python3 basic-ssrf-system.py
100
101
102
103
104
105
106
107
108
Admin interface at 192.168.0.108
```

Ssrf-with-blacklist-filter

Web Security Academy

SSRF with blacklist-based input filter

Back to lab description »

LAB Solved

Congratulations, you solved the lab! Share your skills! Continue learning »

Home | My account

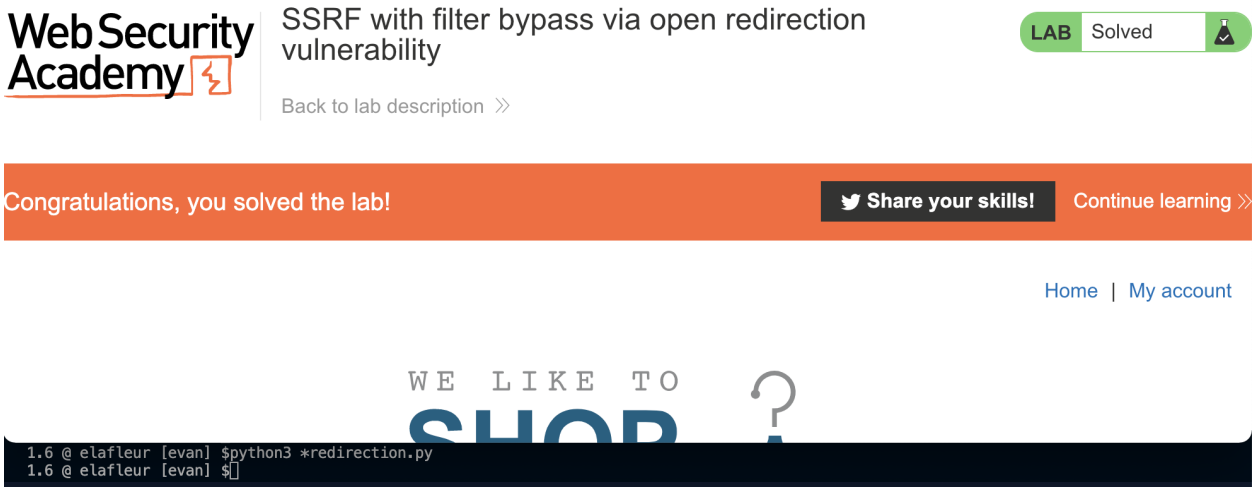
Weird Crushes Game

★★★★★

$70.23

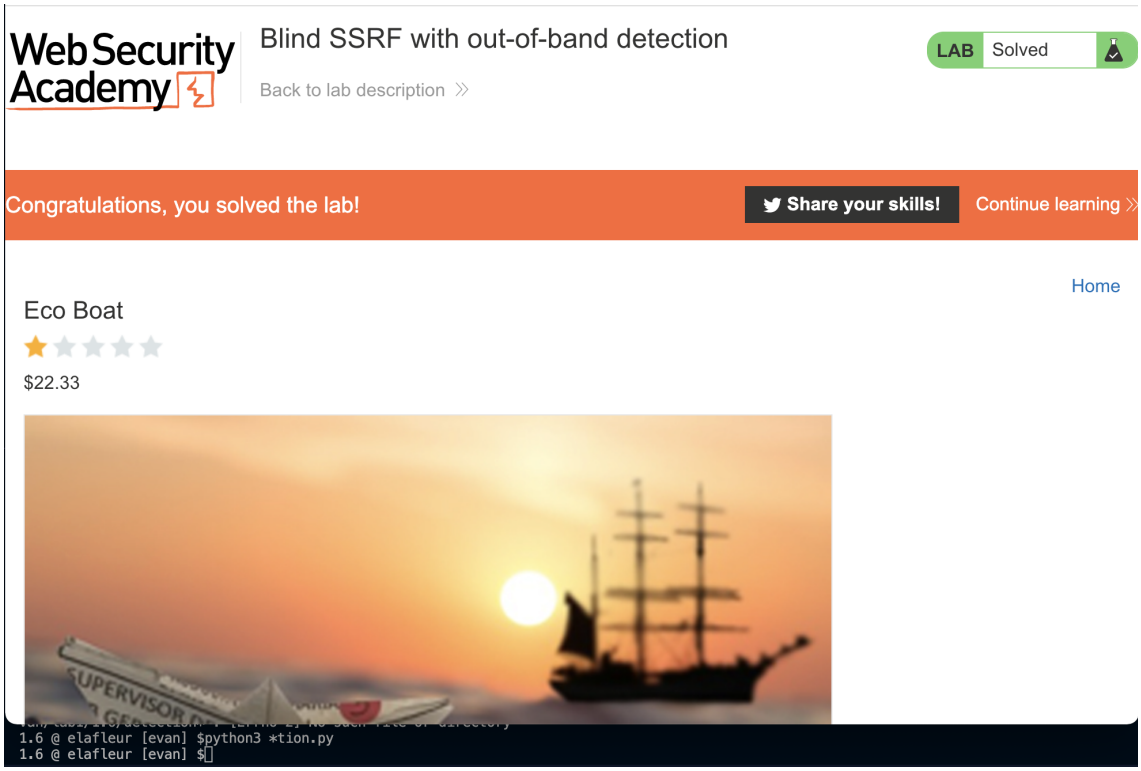
```
1.6 @ elafleur [evan] $
```

Ssrf-filter-bypass-via-open-redirection



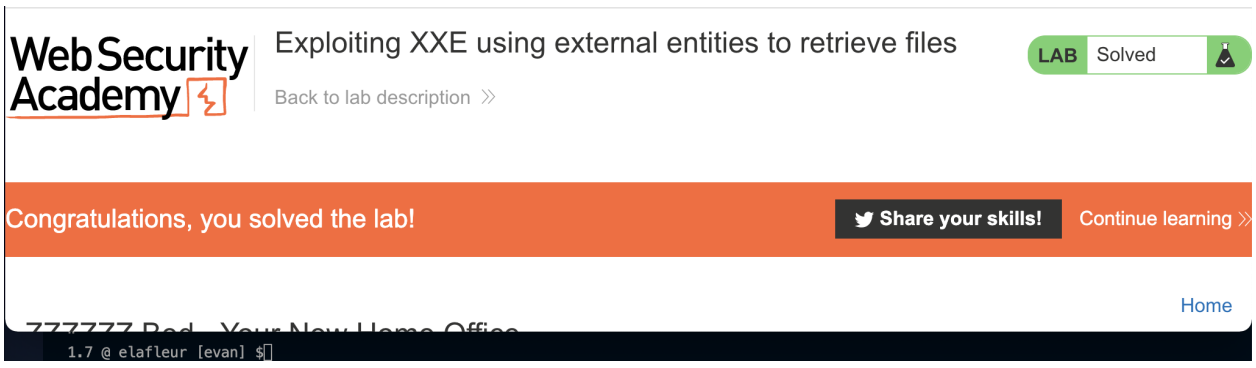
1.6.5 ssrf/blind

out-of-band-detection

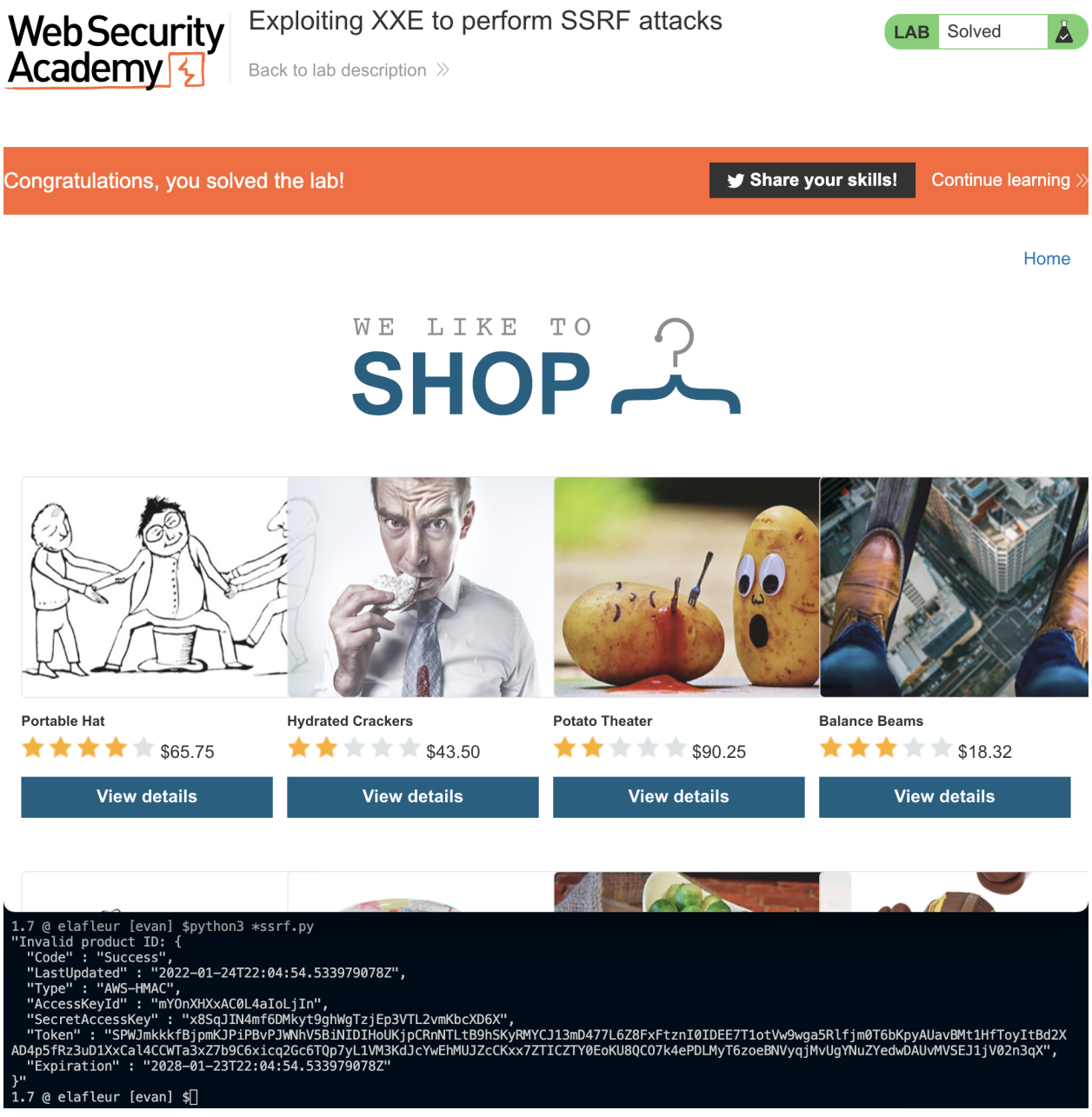


1.7 XXE

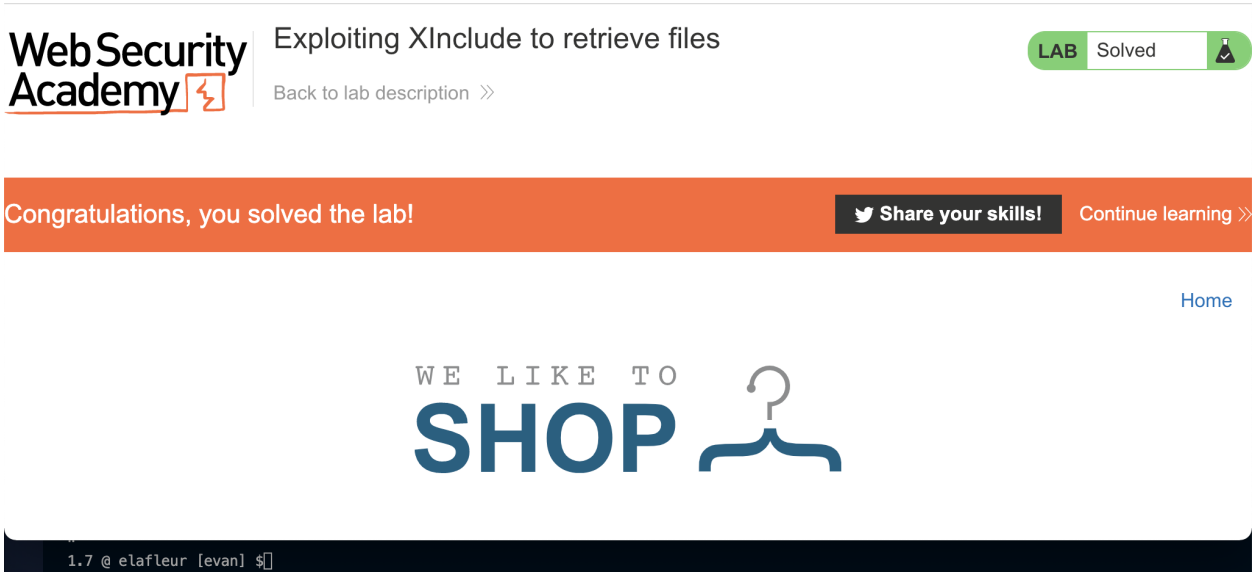
Exploiting-xxe-to-retrieve-files



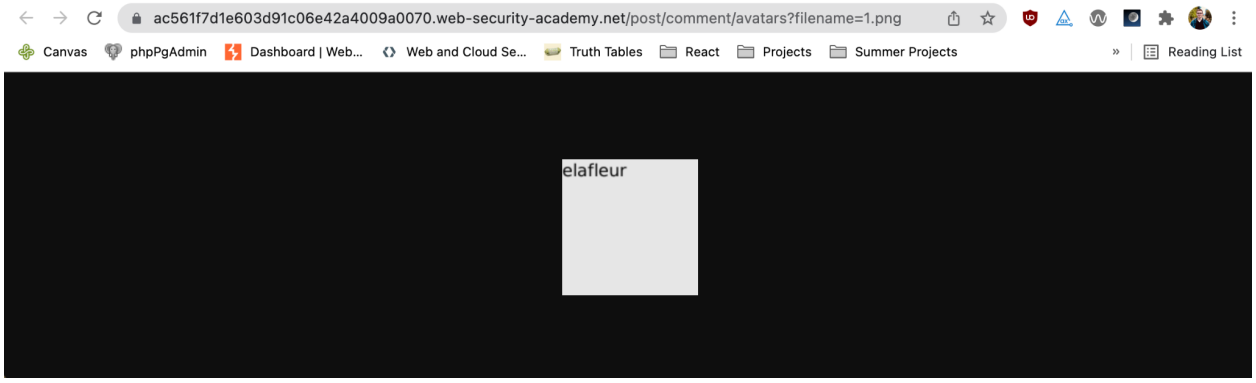
Exploiting-xxe-to-perform-ssrf

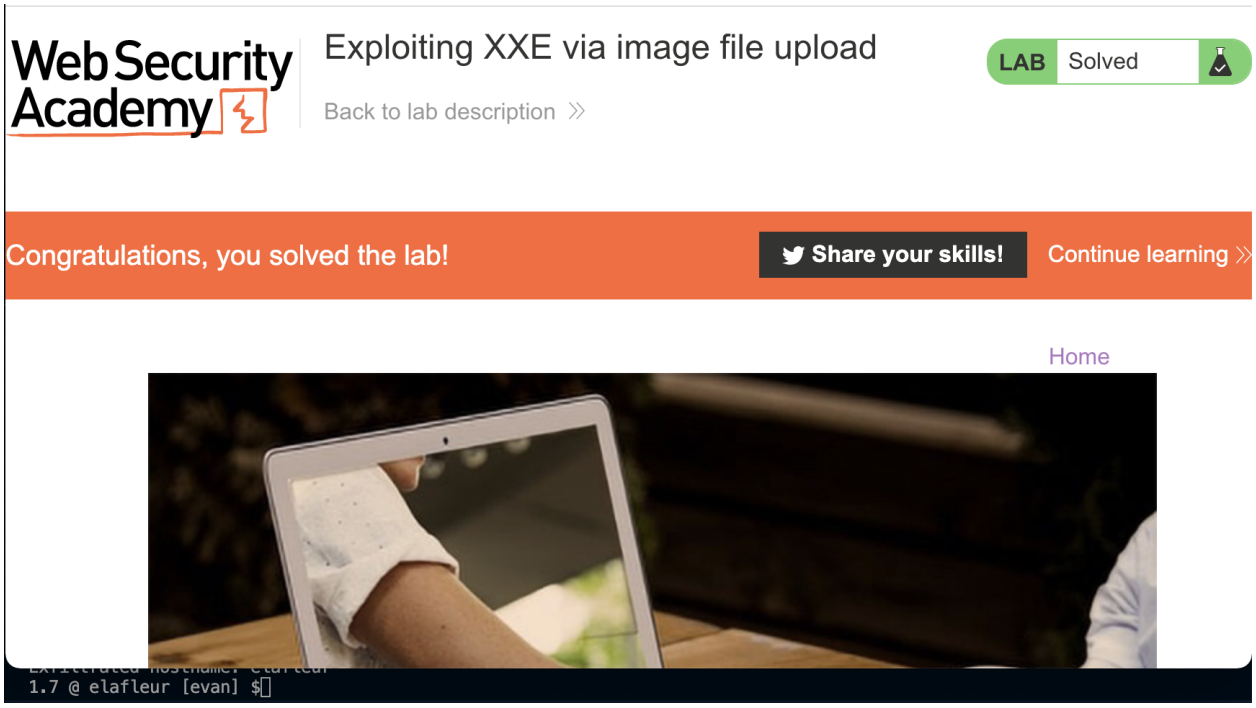


Xinclude-attack



Xxe-via-file-upload




Web Security Academy
Exploiting XXE via image file upload
Back to lab description
LAB
Solved
Congratulations, you solved the lab!
Share your skills!
Continue learning
Home
1.7 @ elafleur [evan] $